

**Narrowcasting-Enabled Groupware Music
Archive Exploration and Gestural Interfaces
for Orchestral Control in Cyberworlds**

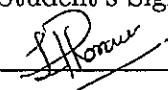


Rasika Ranaweera

A DISSERTATION
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE AND ENGINEERING

Graduate Department of Information Systems
University of Aizu
2016

I hereby declare that this dissertation is entirely the result of my own work except where otherwise indicated. I have used only the resources given in the list of references.

	Date	Student's Signature
Aizu-Wakamatsu	<u>2016-03-01</u>	<u></u>

© Copyright by Rasika Ranaweera.

All rights reserved.

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Michael Cohen

(Prof. Michael Cohen)

S. Nishimura

(Senior Assoc. Prof. Satoshi Nishimura)

Incheon Paik

(Senior Assoc. Prof. Incheon Paik)

Julián Villegas

(Assoc. Prof. Julián Villegas)

The dissertation titled

Narrowcasting-Enabled Groupware Music Archive

Exploration and Gestural Interfaces

for Orchestral Control in Cyberworlds

by

Rasika Ranaweera

is reviewed and approved by:

Main Referee

Professor

Michael Cohen

Michael Cohen

Date

1.3.16

Senior Associate Professor

Satoshi Nishimura

S. Nishimura

Date

Mar 1, 2016

Senior Associate Professor

Incheon Paik

Incheon Paik

Date

Mar. 1, 2016

Associate Professor

Julián Villegas

Julián Villegas

Date

2016/3/1

University of Aizu

January, 2016

This dissertation is dedicated to my loving family...

Acknowledgements

I am heartily thankful to my supervisor, Prof. Michael Cohen, for his continuous support of my research, for his patience, motivation, guidance, and immense knowledge. I could not have imagined having a better advisor and mentor for my Ph.D study and daily life.

To Prof. Minetada Osano, my former adviser for being kind and cheerful, Dr. Ajith Madurapperuma for his encouragement for higher education, Prof. Michael Frishkopf for his stimulating, illustrative, and encouraging correspondence, my gratitude.

To Dr. Masayuki Hisada, for allowing me to work at his company¹ for the last seven years till now and for his guidance with cheerful thoughts, my sincere appreciation.

To my mother, father, brother, and Mihoko, no words I can find which can express my love and gratitude to you.

Lastly, I offer my regards to all of those who supported me in any respect during the completion of my doctoral program.

¹<http://aizulab.com>

Abstract

As immersive virtual environments and online music networks become increasingly popular, it behooves researchers to explore their convergence: for example, groupware music browsers populated by figurative avatars. Collaborative virtual environments (CVEs), offer immersive experiential network interfaces to online worlds and media. “Folkways in Wonderland” (hereafter “FiW”) enables a place where avatar-represented users can go to browse musical databases. When audition is disturbed by cacophony of nearby tracks or avatar conversations, one’s soundscape can be refined since the system supports narrowcasting, a technique which allows information streams to be filtered. An active listener can fork self-identified avatars using a novel multipresence technique, locating representatives at locations of interest, each clone capturing respective soundscapes, controlled using narrowcasting functions $\{\text{self}, \text{non-self}\} \times \{\text{select (solo)}, \text{mute}, \text{deafen}, \text{attend}\}$. Likewise one can participate in a conference and at the same time join a global tour of music. FiW music browser is architected to use MX: IEEE 1599, a comprehensive, multilayered, music description standard. With the integration of the “Wonderland–CVE Bridge,” a user sitting on a rotary motion platform, for instance, with multiple, self-identified “multipresent” avatars to orient themselves

with azimuth or the swivel chair rotary motion platform while maintaining separate locations across an auditioned soundscapes (such as the global music library). The effectiveness of narrowcasting when auditioning music and conferencing was evaluated using FiW as a virtual laboratory. Experimental results suggest that narrowcasting and multipresence techniques are useful for collaborative music exploration and improve user experience. Participants feedback were positive regarding narrowcasting representations, variously based on colors, symbols, and icons.

Desktop virtual reality systems such as Alice, allows ordinary users can drag-and-drop objects (cities, buildings, furniture, instruments, etc.) from galleries and arrange them to create attractive cyberworlds. A simple virtual concert application using Alice in which musical instruments are arranged around a virtual conductor (an avatar of the user) located at their center. A user-conductor can use a smartphone as a simplified baton, pointing at a preferred instrument and tapping a button to start playing. The volume and panning of a selected instrument can be adjusted by simply tilting and steering the smartphone. When selected, an instrument is jiggled or its components dilated and contracted, and a spotlight illuminates it until the instrument is muted, providing conductor and audience with visual cues about the ensemble. Unlike other systems, ours does not require user or equipment to be placed at specific locations (contrasted with Kinect, Wii sensors, or camera-based tracking systems), there is no issue regard-

ing room lighting (such as digital camera-based tracking systems or Kinect), nor interference with colleagues or obstacles. The goal of using different equipment as a conductor's baton is to allow nonexpert users to lead a realtime concert within a cyberworld. The synchronization of gestures with music and animation has been one of the biggest challenges in many systems we have surveyed, although ours had only minimal delays. We had compared user experience with a contemporary commercial game, receiving acceptable ratings from the participants. By sensing its magnetometer, the twirling of a mobile phone can be used to sequence score-following music. Synchronizing such sequencing with sound spatialization, also modulated by the azimuth of the whirled phone, as through an annular speaker array, allows interactive, multimodal interaction.

Contents

Contents	ix
List of Figures	xv
List of Tables	xvii
I Overview	1
1 Introduction	2
1.1 Background	2
1.2 Major Contribution	3
II Proof of Concept	6
2 Gestural Interface for Virtual Concerts	7
2.1 Introduction	7
2.1.1 Related Work	7
2.1.2 Virtual Symphony Orchestra	10
2.2 Design and Development	12

2.2.1	Desktop VR: Alice	12
2.2.1.1	Animation in Alice	12
2.2.1.2	Control soundscape using Java Sound API	13
2.2.1.3	Sampled audio vs. MIDI	14
2.2.2	Smartphone Sensors and Capabilities	15
2.2.3	System Design	16
2.2.4	CVE Client–Server Architecture	17
2.2.5	Alice–CVE Bridge	18
2.2.6	Android–CVE and iOS–CVE Bridge	18
2.2.7	iBaton	20
2.3	Virtual Orchestral Performance	20
2.3.1	Preparation	20
2.3.2	Control VR Orchestra	21
2.3.2.1	Alice in Wonderland	22
2.3.2.2	Enhanced visuals with stereoscopy	22
2.4	Evaluation	23
2.4.1	Fantasia: Music Evolved	24
2.4.2	Subjects	25
2.4.3	Procedure	25
2.4.4	Experimental Results	26
2.4.4.1	User Experience	26
2.4.4.2	Event Synchronicity	27
2.5	Discussion	30
2.6	Conclusion and Future Work	31

3	Whirled Sequencing of Spatial Music	32
3.1	Introduction	32
3.2	Architecture	33
3.2.1	Data Distribution: Client–Server Architecture	35
3.2.2	Synthesis	35
3.2.3	Spatialization	36
3.3	Future Research	36
 4	 Exploring Music in Wonderland	 42
4.1	Immersive Virtual Environments	42
4.1.1	The Sims 3	42
4.1.2	On-Line Interactive Virtual Environment (OLIVE)	43
4.1.3	Second Life	44
4.1.4	Open Cobalt	44
4.1.5	Open Wonderland	45
4.1.6	Most Suitable Virtual Environment	45
4.2	Folkways in Wonderland	47
4.3	Related Research	49
4.3.1	Music Information Retrieval	49
4.3.2	Spatial Sound Diffusers	50
4.3.3	Social (Distributed) Music Audition	51
4.3.4	Ethnomusicology of, and through, cyberworlds	52
4.4	Exploring Folkways in Wonderland	53
4.4.1	Architecture and Implementation	57
4.4.1.1	Basics of Wonderland	57

4.4.1.2	Graphics Rendering using jME	59
4.4.1.3	Arranging Music Tracks in 3D Space	60
4.4.1.4	Music Collection Parsing	62
4.4.2	MX: IEEE 1599	62
5	Narrowcasting and Multipresence in Wonderland	65
5.1	Narrowcasting in Wonderland	65
5.1.1	Introduction and Related Work	65
5.1.2	Conferencing using jVoiceBridge	68
5.1.3	Audio Transmission for Voice and Music	68
5.1.3.1	Protocols: Control, SIP, and RTP	69
5.1.3.2	Calls, Treatments, and Softphone	69
5.1.4	Binaural Parallax: Localizing Sound	70
5.1.4.1	Mute (Nimbus Disable): $A \rightarrow \bar{B}$	74
5.1.4.2	Select or Solo (Nimbus Enable): $A \rightarrow \bar{B}^+$	75
5.1.4.3	Deafen (Focus Disable): $A \rightarrow -B-$	76
5.1.4.4	Attend (Focus Enable): $A \rightarrow +B+$	77
5.1.5	Textual Narrowcasting	77
5.1.6	Narrowcasting State Representations	79
5.1.6.1	Figurative Icons	82
5.2	Multipresence in Wonderland	83
5.2.1	Autoventriloquism	84
5.2.2	Autofocus	85
5.2.3	Coexistence: Not Only One→Many but Many→One	86
5.3	Evaluation and Assessment of FiW	86

5.3.1	Experimental Design	86
5.3.1.1	Procedure	87
5.3.1.2	Subjects and Preparation	87
5.3.1.3	Musical Scavenger Hunt	87
5.3.2	Experimental Results	88
5.4	Discussion	90
5.5	Enabling Multimodel Interfaces	91
5.5.1	Wonderland–CVE Bridge	91
5.5.2	Rotary Motion Platform	92
5.6	Conclusion & Future Research	93
III Summary		96
6 Conclusions		97
6.1	Gestural Interface for Music Control	97
6.2	Narrowcasting and Multipresence in Cyberworlds	98
IV Appendix		100
Appendix A		101
6.3	Activities and challenges	101
6.3.1	Preparatory formalization	102
6.3.2	How to play Fantasia	102
6.3.3	Create a song and export to iBaton	103
6.3.4	How to play iBaton	104

CONTENTS

6.4	Questionnaire	105
6.5	Critics and Suggestions:	106
	Appendix B	107
6.6	Activities and challenges	107
6.7	Preparatory formalization	109
6.8	Musical Scavenger Hunt Game	111
6.8.1	Preparation	111
6.8.2	Scoring Mechanism	111
6.8.3	Conclusion	112
6.9	Bugs, Issues, and Suggestions:	112
	Appendix C	113
	Curriculum Vitae	119
	References	122

List of Figures

2.1	Virtual concert	9
2.2	Profiled results using JVM Monitor.	14
2.3	System Schematic	17
2.4	Alice in Wonderland	21
2.5	Stereoscopic view of virtual concert	22
2.6	SUS scores based on user responses	27
2.7	User responses to the questionnaire	28
2.8	Performance plot with and without network latency.	30
3.1	<i>Poi</i> performance art	33
3.2	System Schematic	38
3.3	Panoramic images, synchronizable with twirling	39
3.4	Alice “steampunk orgel” scene for visual music	39
3.5	Alice “cyberpunk” <i>poi</i> fire twirling scene	40
3.6	Dynamic maps	40
3.7	Patch bay matrix mixer	41
4.1	Exploring Folkways in Wonderland	47
4.2	A typical FiW session	48

LIST OF FIGURES

4.3	Metadata display window:	55
4.4	FiW System Schematic	58
4.5	Different layouts of Folkways in Wonderland	61
4.6	MX: IEEE 1599 layers:	63
5.1	Simplified formalization of narrowcasting functions	67
5.2	Original Wonderland voice-chat panel	71
5.3	Extended FiW voice-chat panel	71
5.4	Distance attenuation	73
5.5	Audio treatment architecture in Wonderland	73
5.6	Extended voice-chat architecture	75
5.7	Voice transmission from source to sink	79
5.8	Nimbi of sources	81
5.9	Disambiguating multipresence	83
5.10	Autoventroliquism	85
5.11	Rotary motion platform	93
5.12	World Music in Wonderland	94
6.1	Create and export songs using GarageBand	103
6.2	Sequence diagram for <code>mute/solo</code> invocations	113
6.3	Sequence diagram for <code>deafen/attend</code> invocations	114

List of Tables

2.1	Different strategies for controlling virtual ensembles	11
2.2	Maximum number of polygons for each object	13
2.3	iBaton Operations:	19
2.4	iBaton Simulator	24
2.5	Scoring Scale	25
2.6	Questionnaire to evaluate user experience.	26
2.7	Performance data for baton gestures	29
4.1	Feature comparison of different virtual environments	46
5.1	Different methods for representation of narrowcasting operations .	78
5.2	Results of questionnaire	89
6.1	Scoring Scale	102
6.2	Preparatory formalization	102
6.3	How to play iBaton	104
6.4	iBaton Simulator Operations	104
6.5	SUS questionnaire to evaluate user experience.	105
6.6	SUS Scoring Scale	105

LIST OF TABLES

6.7	Questionnaire to evaluate user experience	106
6.8	Different methods for representation of narrowcasting operations .	108
6.9	Scoring Scale	108
6.10	Preparatory formalization	109
6.11	Questionnaire to evaluate user experience	110
6.12	Descriptions of classes for iBaton.	115
6.13	FiW Client Classes	116
6.14	FiW Common/Server Classes	117
6.15	FiW Utility Classes	118

Part I

Overview

Chapter 1

Introduction

1.1 Background

Our group is working on interactive multimedia including multimodal interfaces, virtual environments (VES), smartphones, and spatial sound. With the emergence of virtual environments, even ordinary users can drag-and-drop 3D objects (cities, buildings, furniture, instruments, etc.) from galleries and arrange them to create attractive cyberworlds. Alice is a such 3D rapid prototyping programming environment, we were inspired to control and display a virtual concert. A user who only has a taste for music and intention of live performance can simply use a smartphone as a conductor's baton, to control a virtual concert. Smartphones have a variety of sensors; such as proximity, ambient light, gyrosopic, etc. A gyrosopic sensor detects 3-axis angular acceleration around the X , Y and Z axes, enabling capture of roll, pitch, and yaw. Using these capabilities, a smartphone can be used as a pointing device, in this case a conductor's baton. Following the introduction, the second chapter of this dissertation describes our experiences

while controlling a virtual concert using a smartphone as a baton.

The third chapter describes sequencing of spatial music using whirling interfaces. “*Poi*,” a Māori performance art featuring whirled tethered weights, combines elements of dance and juggling. It has been embraced by contemporary festival culture, including extension to “glowstringing” or “fire twirling,” in which a glowstick or burning wick is whirled at the end of a tether. We further modernize this activity, opening it up to internet-amplified multimedia. By sensing its magnetometer, the twirling of a mobile phone can be used to sequence score-following music.

The fourth and fifth chapters describe a cyber world which we used to explore narrowcasting and multipresence features with multimodal interfaces. FiW enables music browsing in a multidimensional, multimedia CVE rather than through textual lists, enabling a sonic, social, and spatial experience. By implementing multimodal narrowcasting at both server (music audition) and client sides (voice- and text-chat), we show how narrowcasting features can extend existing systems. We further discuss interesting examples of a configuration enabled by multipresence and narrowcasting such as “autoventriloquism,” “autofocus,” and “coexistence.”

1.2 Major Contribution

We describe how smartphones can be used as a simplified batons in virtual concerts [Ranaweera et al. \(2012a\)](#). A user-conductor can point at an instrument to select it, tap the screen to start/pause playing, and slide a control to adjust volume. Even though using smartphones as pointing device is not a novel idea,

our experiments on controlling (selections, volume, panning, and play/pause) virtual concerts using smartphones suggest the power of emerging mobile devices as generalized remote controls for interactive multimedia and ubiquitous computing.

By modernizing *Poi* activity it opens up to internet-amplified multimedia. The ubiquity of the contemporary smartphone makes it an attractive platform for event location-based attractions [Ranaweera et al. \(2012b\)](#); [Cohen et al. \(2012\)](#). By sensing its magnetometer, the twirling of a mobile phone can be used to sequence score-following music. Synchronizing such sequencing with sound spatialization, also modulated by the azimuth of the whirled phone, as through an annular speaker array, allows interactive, multimodal interaction.

Folkways in Wonderland [Ranaweera et al. \(2013, 2011b\)](#) is an immersive interface for browsing selections from the Smithsonian Folkways music collection¹ inside “Open Wonderland.” This music cyberworld can also be served as a virtual laboratory for the ethnographic study of music. When audition of music is distracted from the nearby tracks, narrowcasting operations are used to refine ones soundscape [Fernando et al. \(2006\)](#). Further, this system is compliant with the MX: IEEE 1599 standard [Baggi and Haus \(2009\)](#), as the selection of track population metadata is curated in an XML file, enabling analogous applications to interact with Folkways music content. With the integration of the “Wonderland–CVE Bridge,” a user sitting on a rotary motion platform [Ranaweera et al. \(2009\)](#), for instance, with multiple, self-identified “multipresent” avatars to orient themselves with azimuth or the swivel chair rotary motion platform while maintaining separate locations across an auditioned soundscapes (such as the global music library). Using our cyberworld as a virtual laboratory, we evaluated

¹www.folkways.si.edu

the effectiveness of narrowcasting when auditioning music and conferencing. Experimental results suggest that narrowcasting and multipresence techniques are useful for collaborative music exploration and improve user experience. We also got positive feedback from the participants regarding narrowcasting representations, variously based on colors, symbols, and icons.

Part II

Proof of Concept

Chapter 2

Gestural Interface for Virtual Concerts

2.1 Introduction

We have created a mixed reality concert application using Alice, in which musical instruments are arranged around a virtual conductor (in this case the user) located at their center. A user-conductor can use a smartphone as a simplified baton, pointing at a preferred instrument and tapping to select or start playing. The volume and panning of a selected instrument can be adjusted by simply tilting and steering the smartphone.

2.1.1 Related Work

Contemporary related research spans various strategies and techniques for simulating a conductor's baton. Summarized by Table 2.1, tracing tactics include infrared (IR) Behringer (2005); Borchers et al. (2001); Schertenleib et al. (2004),

2. iBaton: Controlling Virtual Concerts using Smartphones

hand gestures [Segen et al. \(2000\)](#); [Je et al. \(2007\)](#), radio [Carlson et al. \(2003\)](#), and set-top console gaming wands [Bradshaw and Ng \(2008\)](#). Behringer’s music instrument interface system [Behringer \(2005\)](#) captures musician motion from optical signals captured by video camera. *Personal Orchestra* [Borchers et al. \(2001\)](#) lets users conduct an audio and video recording of an orchestra, featuring an IR baton and gesture recognition to interpret user input. *Virtual Orchestra* [Schertenleib et al. \(2004\)](#) is a multimodal system architecture in which performance is controlled with a magnetically tracked handheld device. [Carlson et al. \(2003\)](#) present low cost prototypes of mobile input devices by leveraging MANTIS Nymph wireless sensor nodes. [Segen et al. \(2000\)](#)’s visual interface is a realtime visual recognition system that enables a human conductor to control an electronic orchestra. [Je et al. \(2007\)](#)’s computer-based music play system recognize hand gestures captured by stereo camera. The *Conductor’s Jacket* [Nakra \(1999\)](#) a wearable device built into normal clothing, integrates electromyography (EMG) sensors, collecting and analyzing data from conductors. [Bradshaw and Ng \(2008\)](#) track a conductor’s hand movements using Nintendo Wii (3D accelerometer, IR tracking camera, and pressure sensors in Wiimote controller) and track balance shifts within a conductor’s stance (using Wii Balance Board) to analyse conducting gesture. Yamaha’s hand controller [Usa \(1996\)](#) is a realtime tempo controller, a tapper combined with electronic musical instruments. Harmonix’s virtual conducting game, *Fantasia: Music Evolved* (described in § 2.4.1), developed for Xbox with Kinect, uses gesture-based motions [Sanchez \(2014\)](#). The one-man orchestra platform [Tsui et al. \(2014\)](#) uses two Leap motion controllers (a Kinect-like controller which senses hands and fingers and follows their every move) for tracking and a varying number of smartphone devices replicating musical instruments.

2. iBaton: Controlling Virtual Concerts using Smartphones



Figure 2.1: Virtual concert: Instruments are placed in a semi-circular arrangement to make it easy for a user-conductor to select among them. A selected instrument is graphically emphasized using a spotlight-like effect. When played, an instrument is jiggled or its components dilated and contracted. When panned, a phantom replica of a selected instrument is appeared and swings along with audio.

The above systems use various types of complicated tracking devices to parse baton movements, making them complex [Bradshaw and Ng \(2008\)](#) or difficult to setup [Nakra \(1999\)](#). Camera-based methods have the disadvantage that they cannot capture any data if the source is outside the field of view or their view is obstructed in any way [Bradshaw and Ng \(2008\)](#). Most of those systems expect the conductor to perform precise gestures. The goal of using a smartphone as a conductor's baton is to allow nonexpert users to lead a realtime concert within a

2. iBaton: Controlling Virtual Concerts using Smartphones

cyberworld. Compared to other discussed systems (as surveyed in Table 2.1) our approach is rather simpler, requiring no additional tracking devices or cameras.

2.1.2 Virtual Symphony Orchestra

Generally in an orchestra instruments are arranged semicircularly around a conductor, who is responsible for leading performances of music to be played. Using understandable baton movements, the conductor signals to the orchestra regarding tempo or dynamics (loudness or volume), and also cues performers for their entrances. Gathering musicians and conducting a real orchestra is beyond the commitment of normal music lovers, but one can use virtual reality (VR), which allows interaction with a computer-simulated environment, to realize an artificial concert. Realtime expressive effects—such as shaping note onsets, adjusting tempo, articulations, dynamics, and note lengths in a musical score—are produced and modulated using software. However, ordinary computer interactions such as clicking on buttons or pressing keys to play instruments can be less appealing. It would be more natural and realistic if a user could point and control a virtual concert like a real conductor, sitting away from a monitor or projector (as shown in Figure 2.1). Such immersion can be enriched by 3D stereoscopic imaging, such as that displayed in a 3D theater. We deliberately maximize simplicity by eliminating complex and cumbersome devices such as head-mounted displays (HMDs), datagloves, tracking fiducials, and similar peripherals, avoiding a classical VR approach. A modern smartphone (or phablet) can be effectively used as a simplified baton instead of legacy pointing devices. By pointing at members of an ensemble, a user conductor can select an instrument, tap but-

2. iBaton: Controlling Virtual Concerts using Smartphones

Table 2.1: Feature comparison of different strategies for controlling virtual ensembles.

System	Strategy	Features	Limitations
Behringer (2005) 's music instrument interface system	Computer vision technology for obtaining the musical timing and the expression using IR and video camera	Score tracking module	Fast motion can cause significant problems for visual recognition
Schertenleib et al. (2004) 's <i>Virtual Orchestra</i>	Handheld device with a magnetic tracker, a gesture analyzer and 3D scenes and sound renderer	3D sound rendering and gesture analysis	Synchronization of music and animation
Borchers et al. (2001) 's <i>Personal Orchestra</i>	IR baton, a tracker, and gesture recognition algorithm for interprets user input gestures	Controls tempo, volume, and instrument sections	Lack of support for dynamics
Carlson et al. (2003) 's conductor's baton: a gestural interface	MANTIS Nymph wireless sensor nodes and Nymph base station	Inexpensive compared to alternative approaches	Communication latencies of the Nymph system
Segen et al. (2000) 's visual interface for conducting virtual orchestra	A pair of cameras to capture baton and conductor's hand and convey gestures to a sound synthesis system	Instantaneous response and reliable beat detection	Lack of precision of visual recognition at low light situations
Je et al. (2007) 's computer-based music play system	Stereo vision camera and vision-based hand gesture recognition	Requires no extra special devices such as sensor glove, motion capture system, IR camera, or electronic baton	Limited functionality in gesture recognition and the conductor always needs to face the vision sensor
Bradshaw and Ng (2008) 's hand movements tracking system	Wiimote with a reflective marker, two IR LED arrays (containing 96 LEDs each) mounted around two stationary Wiimotes for tracking	Visualization of beat points and allows a balance shifts to be tracked	Complex, tight setup, and limited field-of-view to capture visual information
Fantasia: Music Evolved, virtual conducting game	Gesture recognition through a Kinect motion sensor	Customizable music selections and allows playing with a partner	Kinect distance issues
Ranaweera et al. (2012a) 's iBaton for virtual orchestra	Smartphone to control a virtual orchestra via CVE cloud server	Simplicity, extensibility, responsiveness, and ease of setting up	No support for tempo and limited dynamics (except crescendo and decrescendo)

2. iBaton: Controlling Virtual Concerts using Smartphones

tons to start/pause playing, tilt the device to adjust instrumental volume louder (crescendo) or softer (decrescendo), and steer to adjust panning. Even though the left hand plays many roles in conducting music, such as cuing the players and controlling the dynamics [Segen et al. \(2000\)](#) our system does not recognize two-hand controls, although one can tap controls on the touchscreen to mimic bimanual gestures.

2.2 Design and Development

2.2.1 Desktop VR: Alice

Alice¹ is an innovative content creation environment featuring a rapid prototyping programming system which uses 3D graphics and a drag-and-drop interface for object arrangement and programming to develop interactive virtual spaces [Dann et al. \(2006\)](#); [Pausch et al. \(1995\)](#); [Seidman et al. \(2006\)](#). Alice v. 3 provides a rich application programming interface (API) to control and display objects, including some limited audio-related functions.

2.2.1.1 Animation in Alice

In Alice, motion of a virtual object in space can be influenced by virtual inertia, making motion more realistic. This ease-in/ease-out (also known as ‘slow in/slow out’) animation style [Guzdial and Rose \(2002\)](#); [Pierce \(2003\)](#) could however add delay. Built-in primitive ‘RIGHT_NOW’ tells the rendering engine to execute a command immediately. Such capability can be used to trigger animations that require

¹www.alice.org

2. iBaton: Controlling Virtual Concerts using Smartphones

instantaneous response. Alice scenes are rendered at 60 frames per second (confirmed using the OS X “Quartz Debug” profiling tool). We profiled iBaton using JVM Monitor, a Java profiler integrated with Eclipse (as seen in Figure 2.2) and programatically calculated the number of polygons (as shown in Table 2.2). and found our application scene contains about 20,000 polygons and 11 animations (at maximum pressure), which can be easily handled by a modern computer.

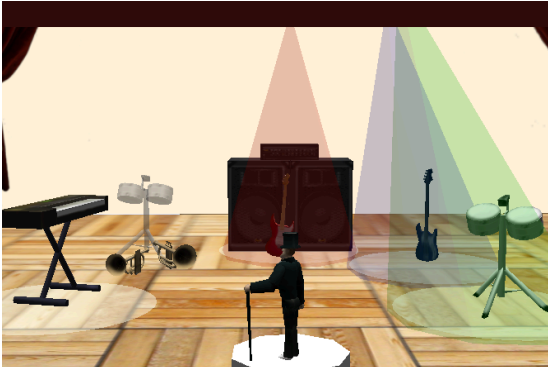
	Object Name	Count	Polygons
	Conductor	1	216
	Podium	1	96
	Speakers	1	108
	Curtains	1	12
	Keyboards	2	1320
	Trumpets	3	8280
	Guitars	2	3984
	Basses	2	3240
	Drums	2	1944
	Spotlights	5	270
	Light circles	5	900
Total	23	20370	

Table 2.2: Maximum number of polygons for each object (both real and phantom) in the virtual concert. Phantom objects (such as the dim timbales behind the trumpets) appear only when pan operation is triggered.

2.2.1.2 Control soundscape using Java Sound API

Alice provides a rich API to control and display objects, but native audio capabilities are limited to playing a sound file and adjusting volume before playing. Using a plugin for NetBeans,¹ Alice scenarios can be exported, edited, and executed as Java programs outside the native IDE (Integrated Development Environments).

¹www.netbeans.org

2. iBaton: Controlling Virtual Concerts using Smartphones

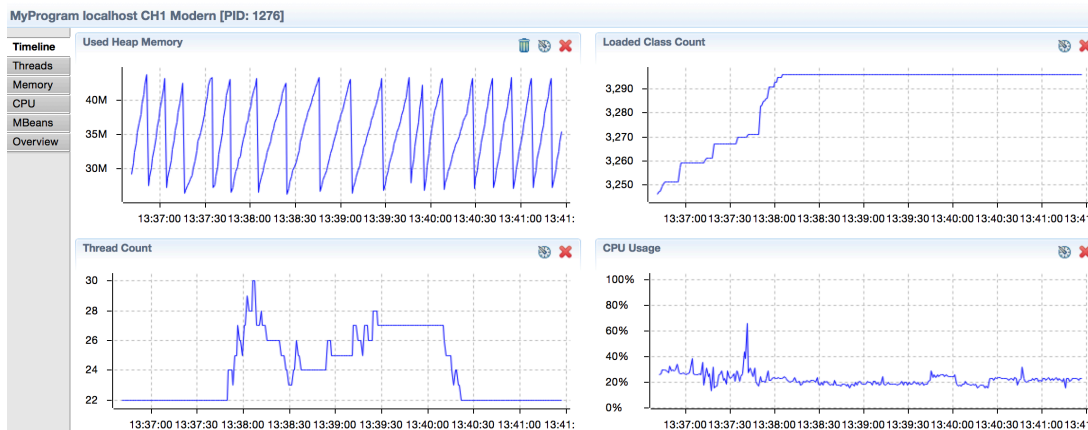


Figure 2.2: Profiled results using JVM Monitor. Note the number of threads in the “Thread Count” graph (lower left) increases when instruments are played and decreases when paused. When all the animations are rendered, a peak appears in the “CPU Usage” graph (lower right). Sawtooth shape of the “Used Heap Memory” graph (upper left) indicates decrement of memory when garbage collection occurs.

The Java Sound API, which provides more extensive control for effecting and controlling sampled audio and MIDI, was used to implement audio-related functions.

2.2.1.3 Sampled audio vs. MIDI

The `javax.media.sound.sampled` package provides handling of digital audio data (also known as ‘sampled audio’). A `Line` represents an element of the digital audio pipeline and a `Mixer` represents an audio device with one or more input `Lines`. `Lines` often have a set of controls—such as gain, pan, reverb, and sample rate—that affect an audio signal. By accessing these effects, dynamics and stereo mix of a musical ensemble can be adjusted during audition. Typical MIDI events in a file consist of actions with parameters such as note number, timing, velocity, and modulation. MIDI files are read by programs called sequencers that are used to control synthesizers. The `javax.sound.midi.MidiDevice.Sequencer`

2. iBaton: Controlling Virtual Concerts using Smartphones

interface contains methods [Flanagan \(2004\)](#) for basic MIDI sequencing [Rumsey \(1994\)](#), including tempo adjustment. The concert scenario is capable of adjusting panning and volume both for sampled audio and MIDI files.

2.2.2 Smartphone Sensors and Capabilities

Contemporary smartphones not only have high-resolution multitouch screens, high speed multicore processors, expansive memory, Wi-Fi, and mobile broadband access, but they also run third-party applications developed using advanced APIs. A wide variety of programming languages— including Objective-C/Swift (Apple iOS), Java (Google Android, BlackBerry), JavaScript/C++/C#/VB.NET (Windows Phone), Python (Symbian S60), and C/C++ (BlackBerry)— are available for developing applications. Smartphone companies encourage developers to create apps by supporting marketplaces (such as Apple App Store, Google Play, Windows Store, and Blackberry App World), providing IDEs (such as Xcode, Eclipse, Visual Studio, BlackBerry JDE, and NetBeans), simulators, frameworks, and code resources. Since the distribution of apps are already taken care by the vender companies iBaton can be easily shared and installed. Besides the previously mentioned features, smartphones also have a variety of sensors (including various means to detect proximity, ambient light, touch, and position). A gyroscopic sensor detects angular acceleration around the x , y , and z axes, enabling capture of *pitch*, *roll*, and *yaw* [Lee \(2012\)](#). The gyroscope may complement an accelerometer, a sensor that reports shake, vibration shock, or fall by detecting acceleration along any of three axes. Integrated data from accelerometric and gyroscopic MEMS (microelectromechanical systems) enable a device to recognize

2. iBaton: Controlling Virtual Concerts using Smartphones

approximately how far, fast, and in which direction it has moved. A magnetometer senses the Earth's magnetic field to determine orientation. The most basic function of a magnetometer is to act as an electronic compass, or to determine the direction that a user is facing Rowlands and James (2010). To minimize errors in sensor readings, the gyroscope, accelerometer, and magnetometer are used in combination through sensor fusion.

2.2.3 System Design

Using Alice, we have created a virtual concert scenario, in which instruments are arranged with a conductor located at their center. The scene resembles contemporary chamber orchestra,¹ consisting of woodwinds (saxophones), brass (trumpets), drums, keyboards, and strings (violins and bass). There are also phantom instruments hidden behind a large loudspeaker set. Using NetBeans, the scenario was edited to connect to our CVE (Collaborative Virtual Environment) session server Kanno and Cohen (2002) and invoke audio-related functions using the Java Sound API. As previously mentioned, positional information of a smartphone can be accessed using the platform's API and transmitted wirelessly. Depending on control device, a middleware program is sometimes needed to capture such events and forward them to a CVE session server. The extended Alice program, subscribed to the same channels, receives events and trigger functions to select instruments in the virtual concert (as shown in Figure 2.3). Depending on the event, virtual concert program can render animations and sound files associated with each instrument.

¹en.wikipedia.org/wiki/Orchestra

2. iBaton: Controlling Virtual Concerts using Smartphones

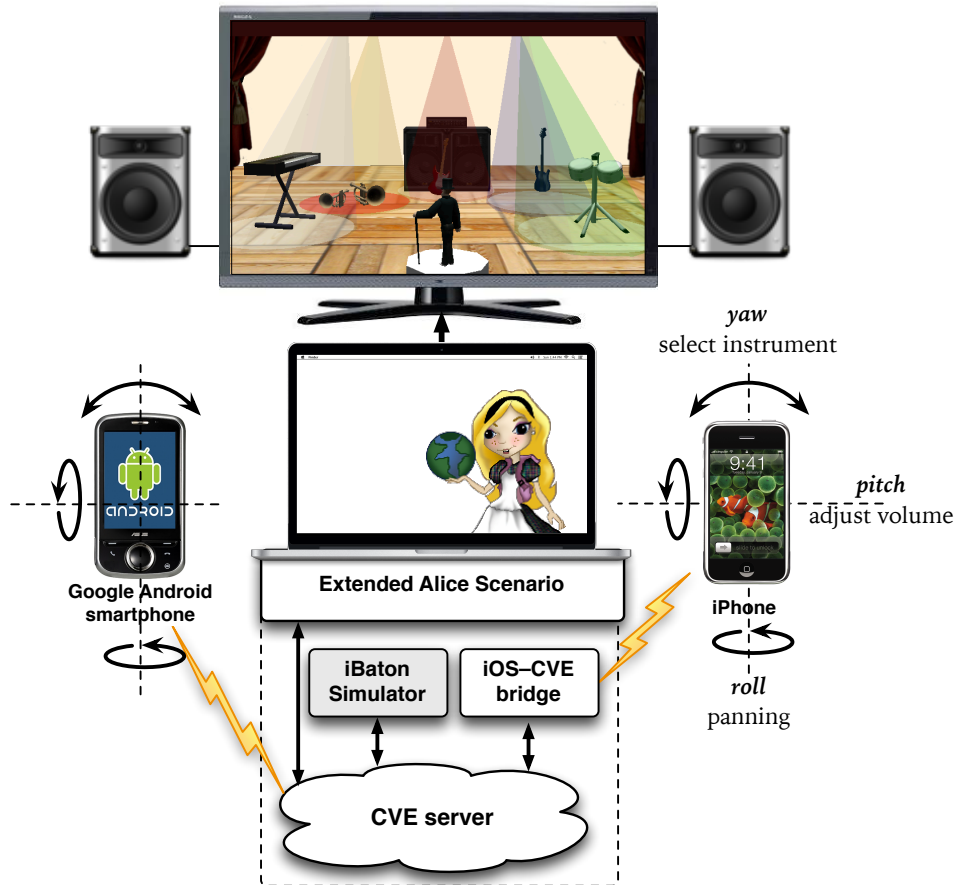


Figure 2.3: The virtual concert application (an extended Alice program) can connect to a CVE session server, the communication link between smartphones or tablets and Alice. Position information of a smartphone is accessed using the device’s API, and wirelessly sent to distributed ‘cloud-based’ services.

2.2.4 CVE Client–Server Architecture

The CVE is a Java-based client–server protocol developed by the Spatial Media Group at the University of Aizu. Clients connect to a session server via sharable channels and when they need to communicate with each other, they subscribe to shared channels. The CVE server receives events (rectangular coordinates and *pitch*, *roll*, & *yaw* rotation around these axis) from session clients and relays updates via multicast to other clients via relevant channels. An extensibility ‘extra

2. iBaton: Controlling Virtual Concerts using Smartphones

`parameter`' can be used to exchange information other than position. This architecture allows multimodal clients to collaborate with each other, including mobile applications, motion platforms, spatial sound, and stereographic displays.

2.2.5 Alice–CVE Bridge

The Alice–CVE Bridge is integrated monolithically with the Alice program to connect to the CVE server. At launch-time a connection is created and used to subscribe to a particular channel to which other clients are also subscribed. When position updates are received from the CVE server, orientation coordinate *yaw* can be used to determine which instrument is selected. Depending on the value of the `extra parameter`, an instrument can be started, paused, or have its volume adjusted.

2.2.6 Android–CVE and iOS–CVE Bridge

Both Android and iOS provide software development kits (SDKs) to access positional information. Programming Android devices to connect to a CVE server is straightforward since the respective native languages are both Java. As iOS native programming is in Objective-C (Swift was introduced in 2014), a device's spatial information can not be directly read by CVE server. Instead such information is streamed to a TCP socket and a middleware Java program reads the socket and relays to the CVE server as JSO, Java Serialized Objects. Even though this architecture is not as simple as that for the Android case, an end user need not be concerned with differences between platforms, except for minor differences between the user interfaces.

2. iBaton: Controlling Virtual Concerts using Smartphones

iBaton Interface	Function	Description
	Connection	IP address where the iOS bridge is running
	Calibrate	When the conductor faces the screen s/he can trim the heading using calibrate buttons. The adjusted yaw should be 90° when centered (facing forward) so the leftmost side is at 0° and rightmost is at 180°.
	Toggle Select	Turn this button on, then point at an instrument to select it. Only one of the three toggle buttons can be turned on at a time.
	Direction	Compass needle, indicating magnetic North
	Channel	CVE server channel to which client is subscribed
	Controls	Play, pause, mute and unmute buttons for sending commands as CVE extra parameters. “Send” button can be used to immediately send a “one shot” event regardless of the mode.
	Toggle Volume	Turn on, then tilt the phone to control gain (0 ≤ volume ≤ 1). A volume icon and value appear beside the toggle button when turned on.
	Toggle Pan	Turn on then steer the phone to control stereo pan left to right (−1 ≤ pan ≤ 1)
	Mode	When “Continuous” mode is selected, enabled rotational parameter values are continuously relayed to the session server.

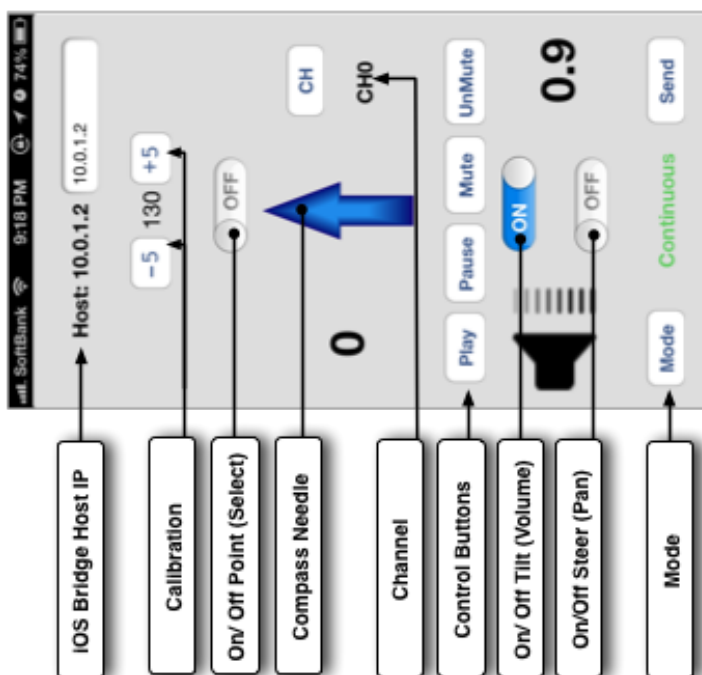


Table 2.3: iBaton Operations: User interface of iOS iBaton app, which sends device orientation to a CVE server through iOS-CVE bridge. It allows a smartphone to be used as a conductor’s baton to control virtual concerts.

2.2.7 iBaton

The iBaton is a smartphone app which we deploy as a virtual baton interface (as shown in Figure 2.2.6) for controlling (as described in Table 2.3) virtual concerts. In the iOS SDK (iBaton was originally developed for iOS 5 but it is also compatible with later versions), the device’s accelerometer and gyroscope data are encapsulated within the `CMMotionManager` class in the `CoreMotion` framework Lee (2012). Reading *roll*, *pitch* (elevation), and *yaw* (azimuth) from this fused sensor information, *roll* (steer) is mapped to pan, *pitch* (tilt) to volume, and *yaw* (point) to selection, allowing the iBaton to be used intuitively. In iBaton only one type of rotation is allowed at a time. Hence the users do not need to worry about smooth movements. Once the app is installed on an iOS compatible device, spatial information can be sent to a CVE server after the application launches through a middleware program.

2.3 Virtual Orchestral Performance

2.3.1 Preparation

A workstation running Alice renders the virtual concert scene, and stereo speakers attached to the workstation display the concert soundscape. A CVE server running on the same (or a different computer) accepts connections from both the virtual concert application and smartphones. Once connections are established and the smartphone is calibrated with monitor setup, a virtual concert can be conducted. In order to calibrate one should face the screen, point the phone baton at an instrument, then tap on “+/-” buttons intuitively to align direction

2. iBaton: Controlling Virtual Concerts using Smartphones

with selection. Because of this flexibility, the user is not required to stay centered in front of a monitor. As the arrangement is semi-circular, each instrument is assigned a $\frac{180^\circ}{n}$, where n is the number of musical instruments. Any angle within each sector range selects the respective instrument.

2.3.2 Control VR Orchestra

When an instrument is selected, its base is illuminated. A light circle on the floor changes to red when an instrument is muted. When played, an instrument is jiggled or its components dilated and contracted, and a spotlight appears until the instrument is paused, providing conductor and audience with visual cues about the ensemble. When a selected instrument is panned, a phantom replica appears in the graphical display and shifts along with its corresponding sound image. Instead of sending continuous orientation events to a CVE server, iBaton can be configured to respect an angular threshold. In this case throttled events are sent only when the difference between current and previous orientation values exceeds



Figure 2.4: Alice in Wonderland: A preliminary version of the Alice-authored virtual band scenario exported to Open Wonderland. (Wonderland client is running on a Windows XP computer.)

2. iBaton: Controlling Virtual Concerts using Smartphones

the threshold. Such a simple technique can be useful for optimizing performance of the system.

2.3.2.1 Alice in Wonderland

Alice scenarios can be integrated into Open Wonderland, a Java open-source framework for creating collaborative 3D virtual worlds, using a module named ‘CMU’ (for Carnegie Mellon University).¹ One can drag-&-drop Alice scenarios (.a3p files) into a running Wonderland client (as shown in Figure 2.4) and start & stop them. Wonderland provides client-server-based infrastructure, but Alice itself is only stand-alone desktop VR. Having integrated the systems, Alice scenarios can be published over the internet, letting users experience both the sophisticated groupware features of Wonderland and creative animations of Alice. Even though the CMU module is intended to be platform-independent, some unresolvable exceptions prevented us from experimenting with the behavior of the virtual concert with a virtual audience (avatars).

2.3.2.2 Enhanced visuals with stereoscopy

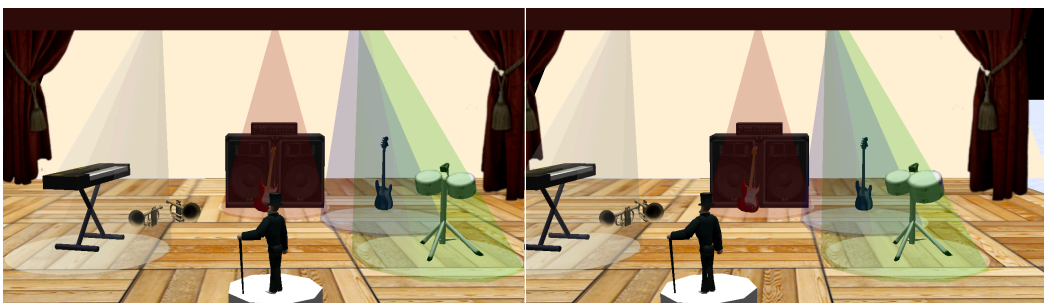


Figure 2.5: Stereoscopic view of virtual concert for cross-eyed free viewing. (Notice, for instance, the binocular parallax of image of conductor in foreground.)

¹openwonderland.org/index.php/table-layout-new/education-and-simulation/cmu-module

2. iBaton: Controlling Virtual Concerts using Smartphones

Stereoscopy refers to techniques for creating or enhancing the illusion of depth in an image by presenting two offset images laterally separately to the left and right eyes of viewers. Most stereographers separate stereo lenses by about 65 mm, the nominal intraocular or interpupillary separation, the distance between the eyes for natural looking stereo. We render stereo perspective by shifting one virtual camera by that baseline in our concert scenario, so immersion is enriched if users, for instance, don 3D eyewear or use a special screen that reflects separately polarized projected images (as shown in Figure 2.5).

2.4 Evaluation

Synchronization of gestures with music and animation has been one of the biggest challenges in many systems Behringer (2005); Carlson et al. (2003). Hence we measured time for each musical control action originated from a smartphone iBaton, as the respective message travelled through Wi-Fi network, was received by the CVE server, then by the virtual concert, triggering function execution delays during performances. We have considered evaluating our system using System Usability Scale (SUS), an effective method for assessing the usability of a product with a standard questionnaire Brooke (1996) to compare usability of our system against the Fantasia, an action game which players can control music using precise hand gestures. We hypothesize that users prefer controlling virtual concerts with smartphones than hand gestures. We further extended this experiment to evaluate user experience when the virtual orchestra is controlled using keyboard or mouse controls (as shown in Table 2.4).

2. iBaton: Controlling Virtual Concerts using Smartphones

iBaton Simulator Application	Function Description
	<p>Click on the arrow button replicates calibration step in iBaton app which align the selection with pointing direction.</p>
	<p>Instruments can be selected by sliding this control. Selected instrument is shown on the left side of the GUI.</p>
	<p>By pushing these buttons one can play, pause, mute, or unmute a selected instrument.</p>
	<p>4. Change volume of a selected instrument</p>
	<p>5. Change panning of a selected instrument</p>

Table 2.4: A desktop app(lication) which users can control the virtual orchestra using keyboard or mouse.

2.4.1 Fantasia: Music Evolved

Harmonix’s “Fantasia: Music Evolved” (hereafter abbreviated “Fantasia”) is a Kinect-powered Zhang (2012) rhythm action game inspired by Disney’s 1940 animated classic. Players are able to explore/select music using a 3D cursor known as “Muse” and control music played by waving their arms in a number of ways; sweeps, punches, sweep-and-hold, punch-and-trace, etc. After starting a song, chutes and paths flash on the screen to help “conductors” keep track of what’s to come. Players sweep either or both arms for sweep cues, which boost volume when hit or attenuate if missed. Switch cues, which occurs at regular intervals, let players select different instruments (“mixes”) by thrusting an arm forward at the right time and swiping towards selected instrument set.

2. iBaton: Controlling Virtual Concerts using Smartphones

1	2	3	4	5
Poor/Too Hard	Basic/Hard	Normal/Moderate	Good/Easy	Excellent/Very Easy

Table 2.5: Scoring Scale (5-point)

2.4.2 Subjects

All participants were Japanese students; computer science undergraduates, age 20–24. They are comfortable using computers as well as smartphone. They also have experience using GarageBand as a part of their computer music course.

2.4.3 Procedure

As entertainment computing, the applications encourage deployment in an enjoyable context, including social situations. As befitting such an opportunity, we set up an experiment to accommodate a small group of players, half a dozen or so, to provide a social context. Each current player was seated on a platform about a meter from the monitor, which was straddled by stereo speakers. Other colocated players sat informally around in a lounge setting. (Per Japanese custom, participants and observers removed their shoes as they entered the lounge area, making the setting more comfortable, clean, and intimate.) Because the Kinect sensor limits its range to around that of the foreground player, other people can be around an active player, including behind, without interfering with gameplay. After demonstrating how to play the game, participants were given basic instructions (in both English and Japanese) regarding game play, familiarize with gestures, select a song with the Muse, then play the game following the onscreen instructions.

Players composed their own songs (a week prior to the experiment), using

2. iBaton: Controlling Virtual Concerts using Smartphones

Q1.	How do you rate your overall Fantasia game playing experience?
Q2.	How easy was the control of the muse with your hand?
Q3.	How do you rate your iBaton experience?
Q4.	How easy was the iBaton calibration compared to Fantasia (Kinect sensor) calibration?
Q5.	How easy was smartphone control compared to keyboard/mouse control?
Q6.	How easy was the control orchestra with smartphone compared to control the muse with your hand?

Table 2.6: Questionnaire to evaluate user experience.

GarageBand, which were organized as separate tracks of about 20 s each. Because the iBaton interface modulates balance which controls intensity-based panning, the source tracks were centered. They were exported as MP3 and converted to WAV at launch time. In order to replicate the similar scenario like in Fantasia, participants were given random tasks verbally—“turn the guitar volume up,” “pan the keyboard to the far right,” “mute drums” etc.—after they started all the instruments. The iBaton graphical scene was rendered on the same flat panel television used for Fantasia, a 42" (~ 107 cm) Toshiba Regza 42Z3.

2.4.4 Experimental Results

2.4.4.1 User Experience

According to its Wikipedia entry,¹ Fantasia has already received strong acceptance, but subjects who played Fantasia (and also iBaton) for the first time rated Fantasia as “marginal” in acceptability rating, “D” or “F” in grade scale, and “Ok” in adjective ratings [Bangor et al. \(2009\)](#). All the subjects rated Fan-

¹wikipedia.org/wiki/Fantasia:_Music_Evolved

2. iBaton: Controlling Virtual Concerts using Smartphones

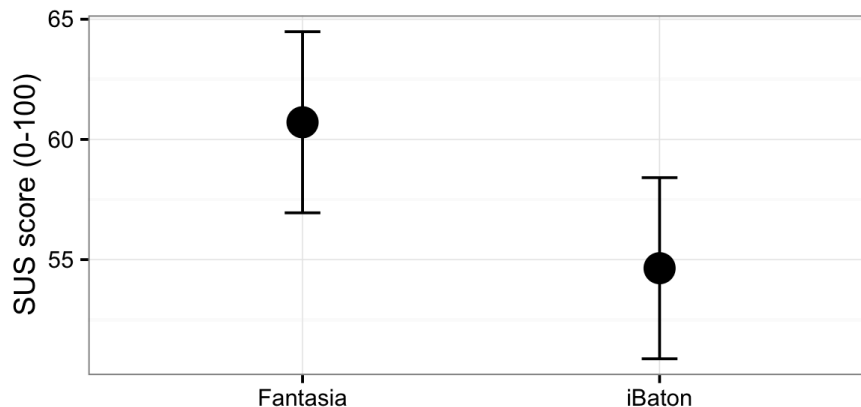


Figure 2.6: SUS scores based on user responses for Fantasia game and iBaton toy with seven subjects. Error bars are Fisher’s Least Significant Difference.

tasia as as good or better than the iBaton system. A within-subjects one-way ANOVA was conducted to compare user satisfaction between iBaton and Fantasia. There was no significant difference in user satisfaction for the two applications [$F(1, 12) = 3.89, p = .096, \eta_g^2 = .144$], as suggested by overlapped bars in Figure 2.6. According to our questionnaire (summarized in Table 2.6) responses (as shown in Figure 2.7), participants found it easy to control orchestra using keyboard/mouse than smartphone. But, they rated the hand-steered muse difficult compared to smartphone control.

2.4.4.2 Event Synchronicity

Since our prototype wireless connection typically goes through a Wi-Fi access point directly to a session server running on some host as display client, sensitivity to network congestion (workday vs. weekend, midday vs. nighttime, etc.) is presumably minimal, so we do not articulate such test conditions. Our experimentation was conducted using a WiFi-connected iPhone 5 and a late-2015 MacBook Pro

2. iBaton: Controlling Virtual Concerts using Smartphones

computer (also WiFi-connected, running the iOS-CVE bridge, session server, and the virtual concert), the “wired” (Ethernet) network was actually not used; only the wireless network was used. Our client-server architecture would allow distributed computation, spreading the various iBaton-related modules around the cloud, but such load-sharing is unnecessary, since the processes are light enough to share a single computer.

We conducted our experiments during “quiet” times (after normal working hours), so packet loss was not an issue. We collected timing data—`select`, `play`, `modulate volume`, and `pan`—plotted in Figure 2.8 and summarized in Table 2.4.4.2. As previously mentioned realistic effects in Alice add extra delay (which explains the time differences between `pan`, `select`, and `volume` vs. `play`). Even without such intentional delays, there is unavoidable network latency between smartphone client & CVE session server, and CVE server & concert ap-

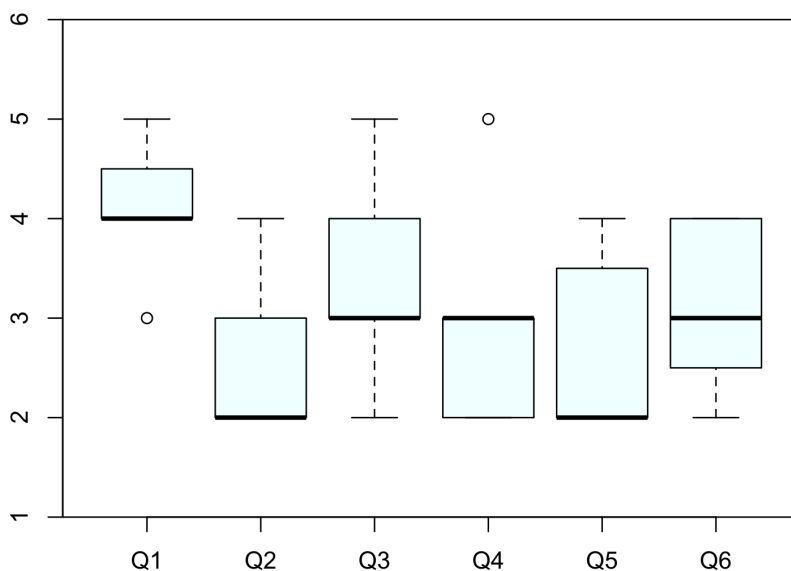


Figure 2.7: User responses to the questionnaire for user experience evaluation based on Table 2.5 Likert scale.

2. iBaton: Controlling Virtual Concerts using Smartphones

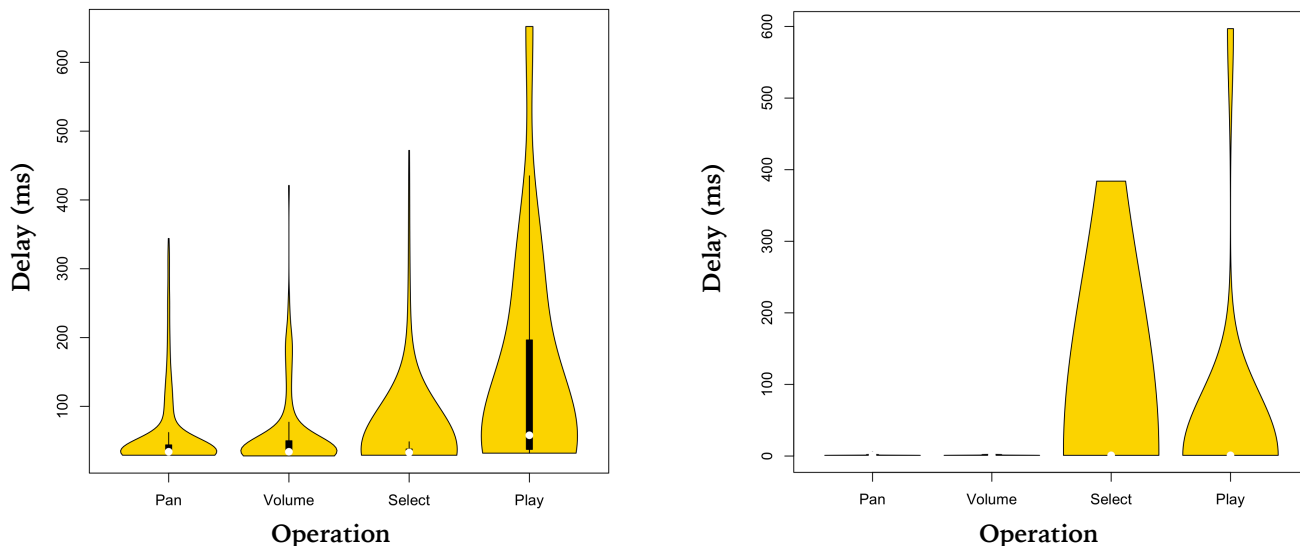
plication. Since all the required components ran locally, latencies between local components is negligible (we measured it as ~ 0.5 ms). The network latency between smartphone and computer was substantial yet unavoidable. In order to provide better control of the system, some of the realism effects of Alice had to remain unexercized.

	With Network				Without Network			
	Pan	Volume	Select	Play	Pan	Volume	Select	Play
Max.	344	421	472	652	6	5	384	597
75% Quartile	44	50	38	195	1	1	1	2
Median	34	34	33	58	1	1	1	1
Mean	58	57	48	137	1.16	1.23	1.64	36.35
25% Quartile	32	32	31	38	1	1	1	1
Min.	29	28	29	32	1	1	1	1

Table 2.7: Performance data with and without network latency (ms).

Since the collected data were not fitting to a normal distribution, violin plots were used to better represent the data. Each violin plot in Figure 2.8 expands from minimum to maximum and the width represents frequency at each point. All operations except change of volume consist of some animations (such as jiggle or component dilation and contraction) beside audio-related operations, which explains the relatively tight distribution. The `play` command invokes heavy animations (lighting or dimming spotlight and instrument-specific jiggling) and audio-related operations (playing includes change of volume too). This data indicates that even with Alice animations, our system has reasonable response time for selections, in contrast to realization of conductor commands in other systems.

2. iBaton: Controlling Virtual Concerts using Smartphones



(a) **Select**, **volume** modulation, and **pan** operations performance data spans 0–50ms, involving comparatively simpler animations. Medians (emphasised with white dot) of violin plots are slightly different for each operation.

(b) Even without network congestion, **play** operation reported comparatively large involving heavy animations and audio-related operations. The sources of variances in **selection** even without network delays are unknown, but is suspected to be due to Java virtual machine garbage collection (GC).

Figure 2.8: Performance plot with and without network latency.

2.5 Discussion

Conducting a real orchestra using precise baton movements requires long-term commitment to accumulate knowledge and skills. Most of the research projects discussed earlier (summarized in Table 2.1) require precise baton or hand movements to express intention of the conductor. Unlike other systems, ours does not require user or equipment to be placed at specific locations (contrasted with Kinect, Wii sensors, or camera-based tracking systems), there is no issue regarding room lighting (such as digital camera-based tracking systems or Kinect), nor interference with other players or obstacles.

2.6 Conclusion and Future Work

We have described how a smartphone can be used as a simplified electronic baton in mixed reality concerts. A user-conductor can point at an instrument to select it, tap the touchscreen to start/pause playing, and tilt/steer to adjust volume and panning. We conducted an experiment (briefly described in §IV), to validate network delays, operational delays, and user satisfaction. We compared user experience with a contemporary commercial game, receiving acceptable ratings from the participants which were comparable with those of the commercial game. Our current approach does not recognize complex movements, although smartphones can capture device movements in 3D space. We propose to retain simple gestures and replace complex gestures using image buttons or physical buttons such as volume controls. Such an approach does not require the user to be professional, and there are no time constraints for realtime response as the commands are instantaneous and precise.

Chapter 3

Whirled Sequencing of Spatial Music

3.1 Introduction

Originally a Māori (indigenous New Zealanders) performance art featuring whirled tethered weights, “*poi*” combines elements of dance and juggling (as seen in Figure 3.1(a)). It has been embraced by contemporary festival culture (especially rave-style electronic music events), including extension to “glow-stringing” or “fire twirling,” in which a glowstick (chemiluminescent plastic tube) or burning wick is whirled at the end of a tether. As seen in Figure 3.1(b), we further modernize this activity, opening it up to internet-amplified multimedia. The ubiquity of the contemporary smartphone makes it an attractive platform for even location-based attractions. By sensing its magnetometer, the twirling of a mobile phone can be used to sequence score-following music. (We had experimented with gyroscopic and accelerometric tracking with Arduino-like sensors, as well as op-

3. Whirled Sequencing of Spatial Music



(a) A *Poi* Performance (Image courtesy of www.pinterest.com)

(b) Multimedia-extended *Poi*

Figure 3.1: *Poi* performance art

tical tracking, but could not overcome various problems, so finally settled on the smartphone-embedded electronic compass.) Synchronizing this sequencing with sound spatialization, also modulated by the azimuth of the whirled phone, as through an annular (ring-shaped) speaker array, allows interactive, multimodal interaction.

3.2 Architecture

This project uses mobile smartphones and tablets (currently Google Android Samsung Galaxy S and Apple iOS iPhone & iPad), in particular their magnetometers (electronic compasses), to track azimuth, sending such heading to a collocated computer via Wi-Fi or over the 3G cellular network. Direct manipulation gives immediate multimodal feedback in both modes of operation— static

3. Whirled Sequencing of Spatial Music

pointing [Ranaweera et al. \(2012a, 2011a\)](#) and dynamic spinning or whirling.

Embedding such devices into a spinnable affordance allows a “spinning plate”-style interface, a novel interaction technique [Cohen et al. \(2013, 2012\)](#). While it is possible to “free spin” a tablet, such a skill is not very easily learned (or generalized to switch hands or spinning direction), and it is impossible to perform with any normal-sized mobile phone. However, anyone can whirl a tethered weight, and whirling is easier to control regarding speed. Whirling also accommodates flexible handedness (ambidexterity) and spinning direction (chirality).

A tether is threaded through a smartphone lanyard or bumper and attached to a kite string winder with a spindle, a swivel pivot handle affordance, around which the smartphone can freely revolve. The user adopts a “Statue of Liberty” pose (maintaining a safe distance from other people, walls, and other obstructions) and simply whirls the device in either direction with a lasso gesture.

Broad configurability allows flexible deployment. Transmission may be one-shot or continuous, including thresholded filtering for choked bandwidth, azimuthal (rotation) and/or circumferential (revolution), and wrapped (folding over at 360°) or unwrapped. A typical whirling rate of a 150 g device swung on a 1 m tether is almost 2 Hz, coinciding with a typical musical tempo of 120 bpm, so sequenced songs are naturally paced at one beat/revolution (or multiples thereof), but can enjoy jitter (tempo variation). A soft transmission gain can scale the control:display ratio, allowing fast twirling to be shared as more leisurely turning (or even “overdriven” to exaggerate such torque). Multimodal display (sounds, lights, vibration) can be invoked locally, on the whirled device, but our “mobile ambient” architecture can distribute such events to collocated “roomware” equipment.

3.2.1 Data Distribution: Client–Server Architecture

As seen in Figure 3.2, to enable easy integration with various multimodal displays, we use our own CVE to synchronize distributed clients. The CVE is a lightweight, subscription-based client–server protocol, multicasting events (such as azimuthal updates) on shared channels. Mobile interfaces were developed for both Google Android and Apple iOS— the former connecting directly with the CVE session server, the latter through some middleware and TCP sockets (Objective-C to Java Serialized Objects).

We use “VR₄U₂C” Bolhassan et al. (2004b) to rotate adjustable fields-of-view from 360° images (QTVR panoramas and turnoramas, such as those of local landmarks), as seen on the right of Figure 3.2 and in Figure 3.3, and middleware to integrate with virtual environments such as Open Wonderland or Alice Dann et al. (2008); Olsen (2011), as seen on the bottom right of Figure 3.2 and in Figs. 3.4 and 3.5, the CG models rigged to accept data from the whirled affordance. A “subscribe/publish” model describes our client–server protocol: a server pushes events on client-registered channels. This replicated unicast synchronizes distributed clients according to subscribed channels.

3.2.2 Synthesis

Currently Java-programmed synthesis is performed on a host’s CPU, and its analog audio signal appears on the (stereo) audio out port of the computer. The sequencing algorithm assumes the Nyquist criteria, that the whirling will be sampled at least twice per revolution. The sequence process listens for the event updates, infers the “ticks,” and synthesizes sequenced notes at the zero-crossings

3. Whirled Sequencing of Spatial Music

(0°, “12 o’clock,” ‘North,’ etc.). This basic algorithm only works for rhythmically simple tunes, such as nursery songs like “Twinkle Twinkle Little Star” (a.k.a. the “Alphabet Song” or, in Japan, “*Pikka Pikka Hikaru*”), which only have notes aligned with the crotchets.

3.2.3 Spatialization

Besides updating visual display clients, the CVE also configures a middleware Pure Data¹ (Pd) [Kreidler \(2013\)](#) program that controls a crossbar matrix mixer embedded in a patch bay router (we currently use the Edirol UA-101) to sweep a monaural audio channel across an 8-channel ring-shaped speaker array, as suggested by Figure 3.7. Pd uses a Vector Base Amplitude Panning (VBAP) [Pulkki \(1997\)](#) patch, configurable by image width (“spread”), to pan between successive pairs or triples of speakers. This 1:8 up-mix (which will eventually be a 2:8 up-mix, using both sides of a stereo pair, for both melody and harmony) diffuses audio and musical signals into a circumferential display. The line-level signal coming out of the 8-channel audio interface goes (via 1/4" phono plug connector cables) to eight powered speakers (Yamaha MSP3s) arranged in a 2 meter radius annulus spaced at equal (45°) intervals (notionally at 1:30, 3, 4:30, 6, 7:30, 9, 10:30, & 12 “o’clock” on a overhead polar mapping).

3.3 Future Research

Arbitrary rhythmically complex or syncopated tunes can not be played, so we are working on another technique that connects the whirled azimuth to standard MIDI

¹puredata.info

3. Whirled Sequencing of Spatial Music

beat clock (a.k.a. MIDI timing clock or simply MIDI clock, and not to be confused with MIDI time code, MTC, which is basically SMPTE time codes), 24 ppqn (**p**ulses **p**er **q**uarter **n**ote), or every 15° of whirling. These realtime clock pulses can be sensed asynchronously by a sequencer that accepts an external clock, and used to drive arbitrary rhythms, within the resolution of the note precision and the limits of the user tempo sampling. MIDI System Real-Time Messages,¹ comprising “Start,” the timing clocks, and “Stop,” will be used to synchronize the sequencer. (We are using the “MIDIClock” tool² to simulate such *jitter*, tempo adjustability.) The resultant analog signal can be routed in the audio interface, using its crossbar functionality. We are thinking to use “ABC notation”³ as a convenient textual coding, allowing easy transcoding to MIDI events and readable by visual music renderers, such as that shown in Figure 3.4, which could use runtime procedural modeling to map such melodic events into, for instance, tangent-triggering pips on a “orgel” music box’s rotating cylinder. Applications are available to convert ABC to MIDI, MIDI to ABC, and ABC to standard scores. Tens of thousands of folk tunes have been encoded efficiently in this way. We look (and listen!) forward to allowing multiple users to whirl their way together through a large song catalog.

¹www.midi.org/techspecs/midimessages.php

²midiclock.com

³abcnotation.com

3. Whirled Sequencing of Spatial Music

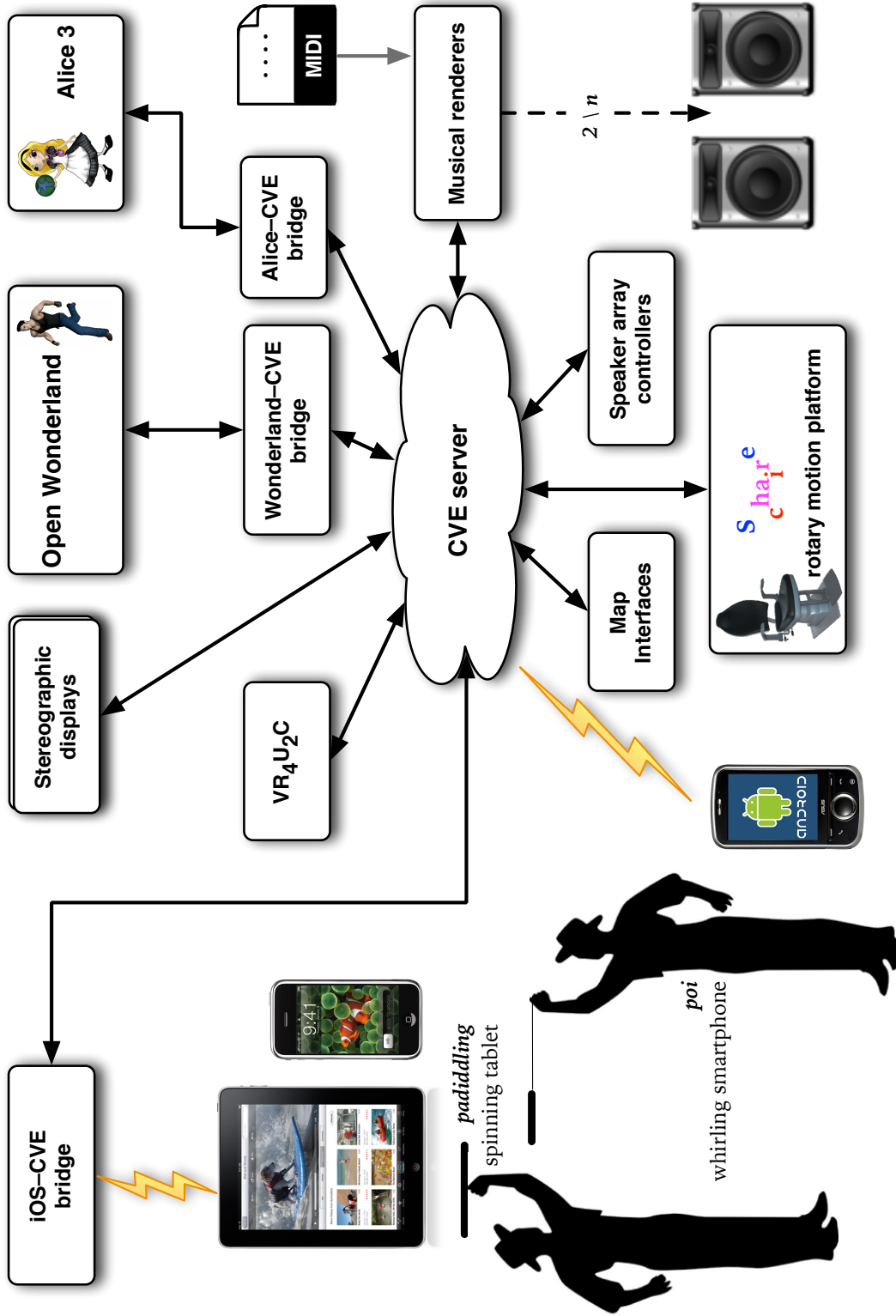


Figure 3.2: Our CVE provides a shared infrastructure, allowing heterogeneous, multimodal, conformant cloudy clients to display data from multiple spinning or whirling affordances (reconstruction of CVE architecture from [Ranaweera et al. \(2012b\)](#)).

3. Whirled Sequencing of Spatial Music



(a) Exterior



(b) Interior

Figure 3.3: Panoramic images, synchronizable with twirling (from [Bolhassan et al. \(2004a\)](#))



Figure 3.4: Alice (v. 3) “steampunk orgel” (music box) scene for visual music (from [Ranaweera et al. \(2012b\)](#))

3. Whirled Sequencing of Spatial Music

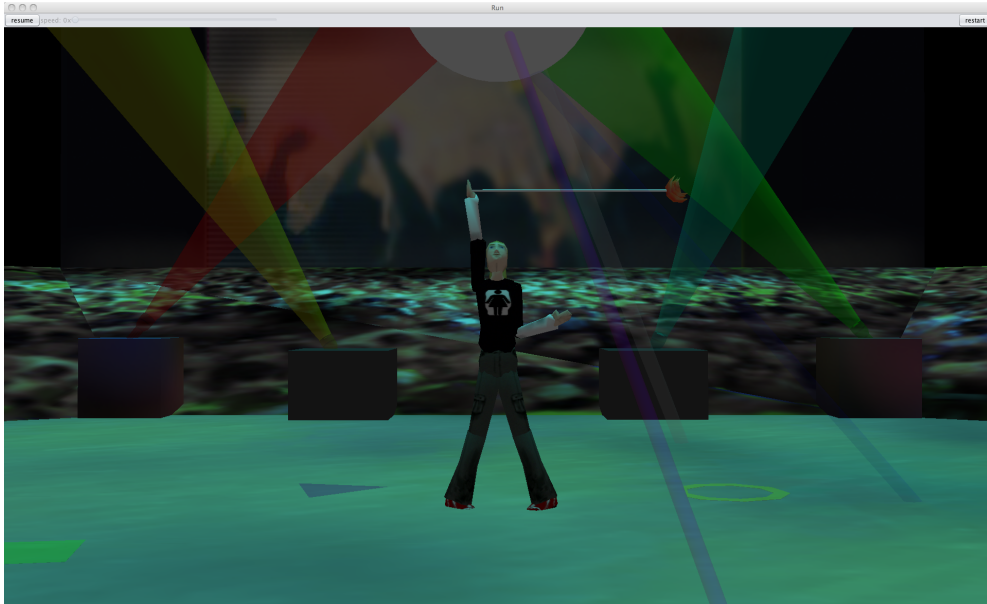


Figure 3.5: Alice “cyberpunk” *poi* fire twirling scene (with “eye candy”) (from [Ranaweera et al. \(2012b\)](#))

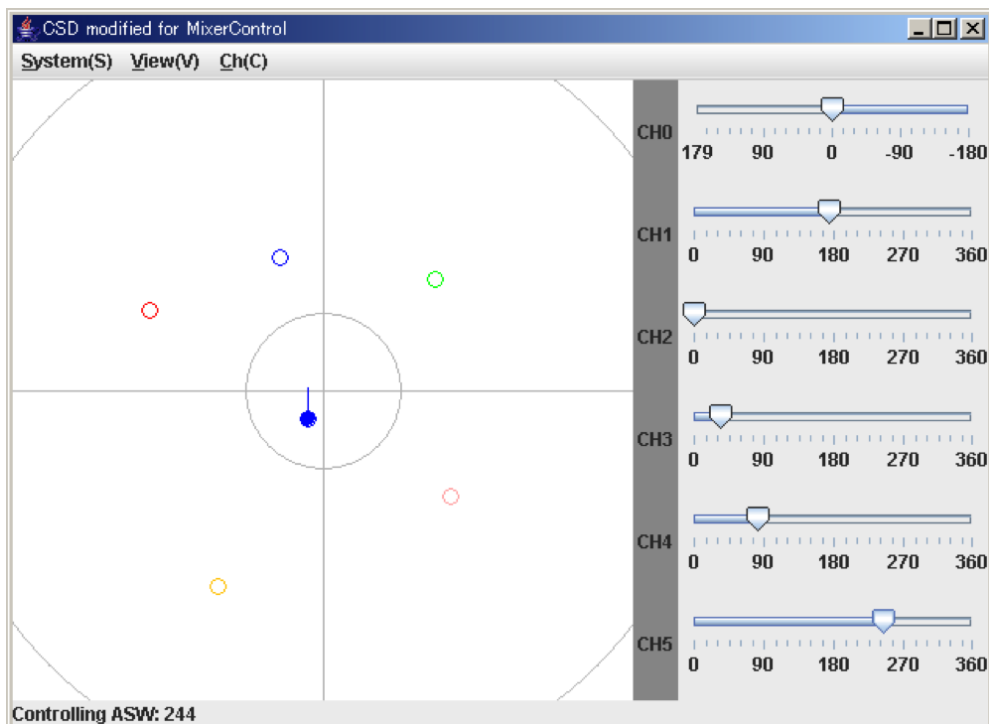


Figure 3.6: Dynamic maps provide orthographic perspectives for easy monitoring of planar motion (from [Yokomatsu \(2007\)](#)).

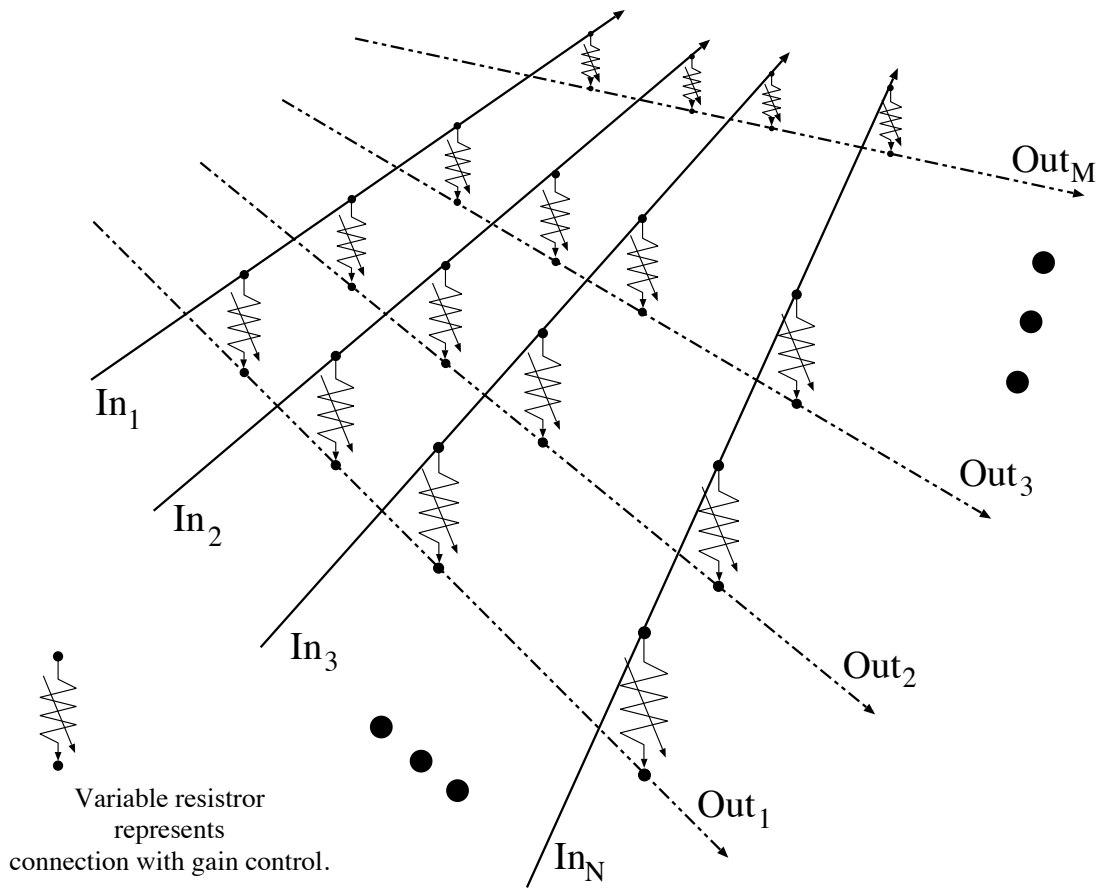


Figure 3.7: Patch bay matrix mixer embedded in the audio interface can distribute both locally rendered audio signals and externally sourced signal (from Yokomatsu (2007)).

Chapter 4

Exploring Music in Wonderland

4.1 Immersive Virtual Environments

Immersion refers to objective level of sensory fidelity provided by VR system (technology); presence (engagement) refers to user's subjective psychological response. An immersive virtual environment (IVE) is an artificial environment where users feel just as immersed as they usually feel in consensus reality. IVEs are used for collaboration and interaction of possibly many participants that may be spread over large distances. In general virtual worlds have characteristics such as: shared space, graphical user interface (GUI), immediacy, interactivity, and persistence [de Freitas \(2008\)](#).

4.1.1 The Sims 3

The Sims 3¹ is a strategic life simulation computer game developed by The Sims Studio and published by Electronic Arts. Player's representation in the game is

¹www.thesims3.com

called a “Sim” who has a personality, which define the character. Sims can have relationships with others, gardening, baking, having parties, working and earning virtual money (simoleons §), and buy items at a store using § or real money. The massively multi-player online (MMO) variation is called “The Sims Online” and smartphone/tablet version is called “The Sims™ Freeplay.” Even though sims are visually pleasing and they can do many things in a realistic fashion most of the content in the game are restricted.

4.1.2 On-Line Interactive Virtual Environment (OLIVE)

OLIVE¹ is a MMO virtual environment which provides avatars with a photo-realistic appearances of actual people, including automated gestures and voice-activated lip synthesis. Other than that it includes video display, document authoring, 3D audio, record and replay, content packs, and support for leading 3D authoring tools. The cyber worlds are produced and hosted by Forterra Systems. OLIVE applications typically focus on employee or staff training, and secure collaborative business decision-making [GChen et al. \(2008\)](#); [de Freitas \(2008\)](#). One of the differences between conventional MMO virtual environments and OLIVE is that real-world locations are replicated in an OLIVE cyber world, as opposed to fantasy-based worlds or hypothetical “islands.” OLIVE is commercial software and it requires a licence to connect, hence limiting social communications.

¹www.prweb.com/releases/virtual/olive/prweb530814.htm

4.1.3 Second Life

Second Life (SL)¹ is a multiuser virtual environment (MUVE) [Mayrath et al. \(2010\)](#) where users called “residents,” who can meet other residents, socialize, participate in individual and group activities, and use voice and text chat. A proprietary library called FMOD is used to power the SL audio engine. Second Life uses a client-server architecture, where the virtual world (known as the grid) runs on dedicated servers (per region) and clients run on users’ high-end personal computers. Since the Second Life client/viewer was made open-source, a number of accessibility solutions are available. Even though developers are not able to modify core server components, “Linden Scripting Language” can be used to add interactivity to objects to the client.

4.1.4 Open Cobalt

Open Cobalt² is a free, P2P, and open source software platform for constructing, accessing, and sharing virtual worlds both on local area networks LAN or across the internet, without any requirement for centralized servers. An integrated spacialized voice chat tool allows one to speak to other users in the same Cobalt space. 3D hyperlinking or “Space-linking” is the teleportation mechanism in Cobalt which allows one to traverse between regions or spaces. Cobalt allows merging virtual worlds but it has comparatively smaller user base.

¹secondlife.com

²www.opencobalt.org

4.1.5 Open Wonderland

Open Wonderland (OWL)¹ is Java open source toolkit for creating collaborative 3D virtual worlds (originally developed as “Project Wonderland” by Sun Microsystems, now supported by an independent foundation). Wonderland uses a client-server model [Crisostomo et al. \(2004\)](#) with various networking protocols for different data types [Kaplan and Yankelovich \(2011\)](#). The Wonderland server integrates several services that can be distributed across multiple machines to increase scalability [Gardner et al. \(2011\)](#). The Darkstar server [Waldo \(2008\)](#) provides a platform for Wonderland to track the frequently updated states of objects in the world. JVoiceBridge, a pure Java audio mixing application, communicates directly with the Darkstar server, providing server-side mixing of high-fidelity, immersive audio. Modules are the mechanism for packaging code, artwork, and other resources, for adding new capabilities to Wonderland. When deployed, a module is unpacked into a directory in the web server. Client code is available to clients via the web server and server code is installed in the Darkstar server [Kaplan and Yankelovich \(2011\)](#).

4.1.6 Most Suitable Virtual Environment

Even though the above virtual worlds provides many features [de Freitas \(2008\)](#); [Messinger et al. \(2009\)](#); [Thompson \(2011\)](#), we have compared and contrasted them (as in Table 4.1) and chosen Open Wonderland, because of its source code availability, high-fidelity immersive audio capabilities, distributed client-server architecture, and extensibility.

¹<http://openwonderland.org>

4. Exploring Music in Wonderland






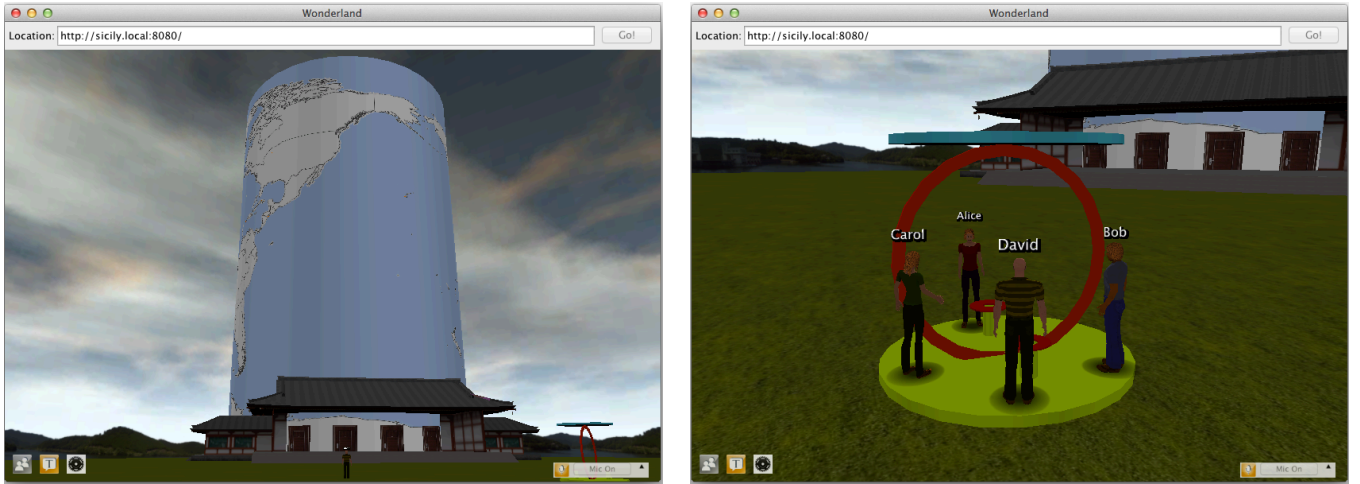
	The Sims	OLIVE	Second Life	Cobalt	Wonderland
Avatars					
Gestures	✓	✓	✓	✓	✓
Expressions	Tertiary	Quaternary	Tertiary	Primary	Secondary
Teleport	✓	✓	✓	✓	✓
Spatialization	✓	✓	✓	✓	✓
Shared Apps	✓	✓	✓	✓	✓
Content	Restricted	✓	✓	✓	✓
Distribution	Desktop, Online, Mobile	Hosted in SAIC	Free clients/viewers	OSS	OSS
Language	Edith	C++	C, C++, Linden Scripting Language	Squeak-Smalltalk	Java
Pros	Visually pleasing, sims have personalities and can do many things in a realistic fashion.	Ability to create lifelike settings and immersive training conditions.	Performance, provides learning management system (LMS) platform, large number of users.	Ability to merge virtual worlds, P2P, mature and stable technology.	High fidelity, platform independent, open source platform.
Cons	Difficult to manoeuvre or customize, restricted content and extensions.	Limited social communications, requires licence to connect.	Steep learning curve for users and developers, requires a powerful computer to connect.	Very small community, requires advanced programming skills to create any object.	Limited no of gestures and avatar animations, requires advanced programming skills for development.

Table 4.1: Feature comparison of different virtual environments based on [de Freitas \(2008\)](#); [Pereira \(2011\)](#)'s case studies and personal experience.

4. Exploring Music in Wonderland



(a) Exterior view of FiW main space.

(b) A discussion park allows quiet conversations among avatars.

Figure 4.1: Exploring Folkways in Wonderland

4.2 Folkways in Wonderland

Our musical cyberworld is called “Folkways in Wonderland”¹ because it is populated with track samples from Folkways Recordings, founded by Moses Asch and Marian Distler in 1948, directed by Asch until his death in 1986, and thereafter published and curated by Smithsonian Folkways, the non-profit record label of the Smithsonian Institution, located in Washington, D.C. During Asch’s remarkably productive tenure, the Folkways label produced 2,168 albums containing over 40,000 tracks, representing a stunningly diverse gamut of recordings from around the world, including musical, spoken, and environmental samples. Since acquiring the label in 1987, Smithsonian Folkways has expanded and digitized the Folkways collection, while enhancing and organizing its metadata, all of which is now available electronically. The Folkways collection thus offers sev-

¹www.youtube.com/watch?v=5Nmyc01qZml

4. Exploring Music in Wonderland

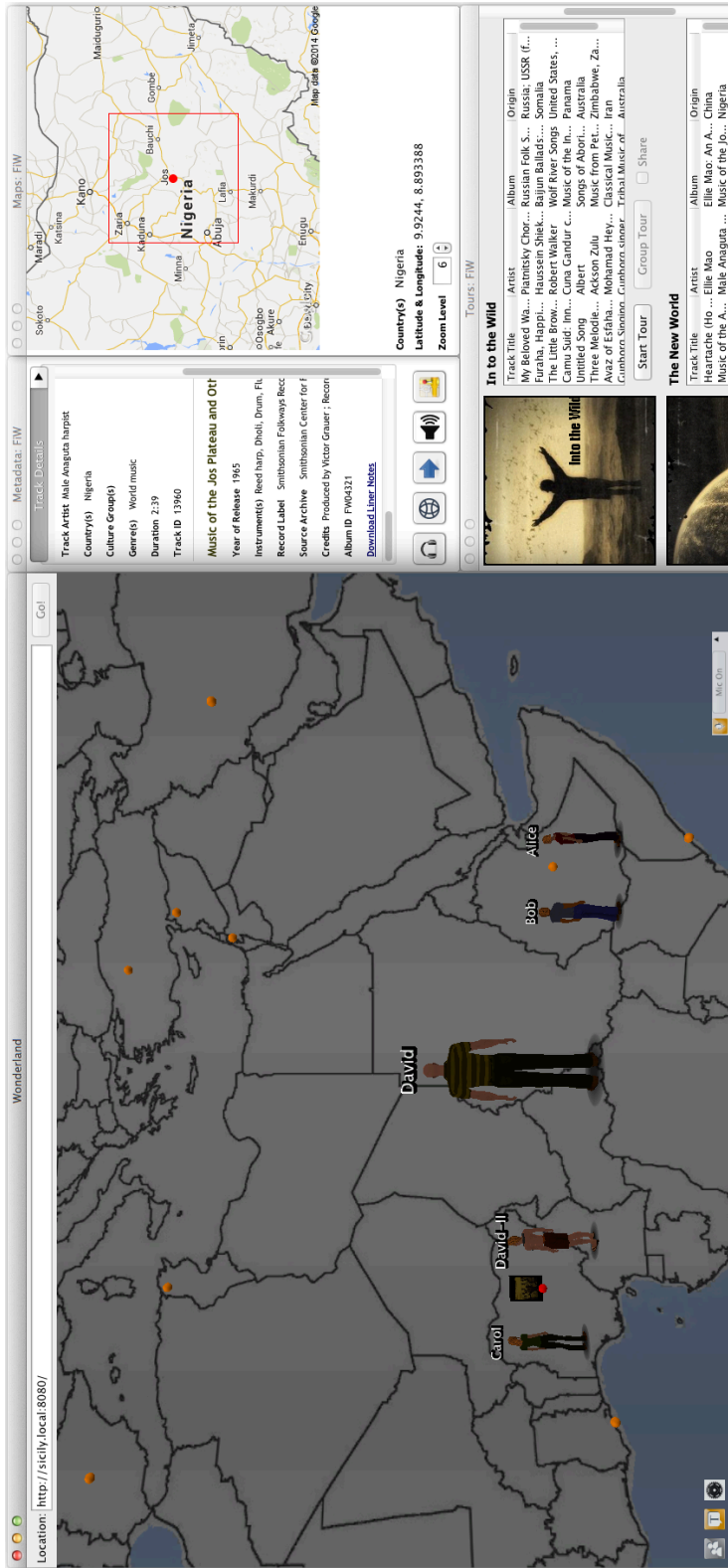


Figure 4.2: A typical Folkways in Wonderland session: In the upper-center window, a user browses metadata for a selected track (located in Nigeria) in the left window. Buttons allow the user to view liner notes, listen via virtual headphones (excluding competing sounds), find the track on the Smithsonian Folkways website, teleport to the origin of the track, search for other tracks, or view the track location on a zoomable Google map (upper right). The user may also embark on a tour, using a window like that shown in the lower right. A user with multiple instances of self can audition music in different places at once (as by David in center and David-II next to red dot). The metadata window shows details of the musical track auditioned by the focal avatar.

eral distinct advantages: it is large, diverse, global, well-documented, and digital. Furthermore, it is, in itself, one of the most important and enduring products of ethnomusicological research ever assembled.

From the full Folkways collection, we have selected, encoded, and geotagged a set of 1,166 music tracks, carefully chosen for aesthetic and cultural interest and geographical distribution, to represent a full spectrum of world music. Out of this set only several dozen populate the FiW cyberworld at any one time; the selection of track population is easily configurable, as track metadata are all stored in an XML file conforming the MX: IEEE 1599 standard [Baggi and Haus \(2013, 2009\)](#).

4.3 Related Research

FiW integrates various functionalities that are typically offered only separately by more specialized programs. Several classes of such focused applications are respectively consider in this section.

4.3.1 Music Information Retrieval

Finding a particular recording is generally supported by traditional search interfaces via metadata [Hughes and Kamat \(2005\)](#), but there is a growing need for improving search techniques via different information retrieval strategies. [Damm et al. \(2008\)](#) introduced a novel user interface for multimodal (audio-visual) music presentation as well as intuitive browsing and navigation. MusicSim [Chen and Butz \(2009\)](#) uses audio analysis techniques and user feedback for browsing and organizing large music collections. There are also different types of music search

engines. For instance, Musipedia¹ offers melody search functions. Music Ngram Viewer² encodes songs for look-up. Folktune Finder³ also has melody and contour search. Although most such applications and interfaces facilitate locating music and visualizing collections, it is also important to take into account what information is desired and how that information will be used after its retrieval Downie (2002). Kuhn et al. (2010)'s mobile music player incorporates several smart interfaces to access larger personal music collections and visualize content using similarity maps.

4.3.2 Spatial Sound Diffusers

Since FiW features immersive audition and spatial audio, it is also related to spatial sound diffusers, although most known ones are for virtual concerts with at least logically collocated musicians. That is, even though the tracks might have been separately recorded, the pieces of music are presented as if in a concert venue. Mention must also be made of online games, which increasingly feature spatialized voicechat. Some games (for example, those in the “Splinter Cell” and “Thief” series) even use the amplitude of positional player sounds to alert NPCs (AI-driven Non-Player Characters), further encouraging “stealth” operations. Funkhouser et al. (1999) describe an acoustic model to locate moving sources and receivers in a distributed VE. A virtual museum application demonstrated in Naef et al. (2002) discuss design principles and practical implementation issues for audio rendering.

¹www.musipedia.org

²www.peachnote.com/info.html

³www.folktunefinder.com

4.3.3 Social (Distributed) Music Audition

Many systems have been developed for music consumption, both stand-alone and distributed, of which perhaps Frank et al. (2008)'s "PlaySOM" interactive 2D music map application is representative. Such groupware systems are instances of collaboration technology for synchronous but distributed (not collocated) sessions. Boustead and Safaei (2004) compared various architectures for streamed audio delivery, including techniques for optimization based on similarity of distribution of avatars in a virtual space with that of human players in the real world.

The major commercial labels have not yet capitalized the way many people really consume, share, and experience digital music. Napster anticipated distributed (initially peer-to-peer) music sharing, but presented basically an asynchronous experience. Many people, especially younger listeners, enjoy music through networked music audition services. Such systems often offer social media features, generalized as "groupware" among human-computer interaction researchers and scientists. For instance, Last.fm promotes "scribbling," publishing one's music-listening habits to the internet, to monitor when and how often certain songs are played, but such journaling is an asynchronous practice. SongPop¹ is a social multiplayer online music identification game, in which players compete against others in realtime to identify song snippets. (In 2012 it was the highest-rated game on Facebook —Snider (2014). Shazam² and SoundHound³ feature realtime maps of music neighbors and what other users are listening to

¹www.songpop.fm

²www.shazam.com

³www.soundhound.com

as “Explore” and “My Music Map,” respectively.

Maybe in the future, online communities with avatars, currently mostly used for interactive 3D social interaction, will be used for browsing media. The main example of such a not-quite-mainstream environment is Second Life, which allows virtual concerts [Greenberg et al. \(2009\)](#), and runs from a distributed network of 40,000 servers (but might eventually be eclipsed by its founder’s subsequent venture, High Fidelity¹). Although network and processing latency prevents totally satisfying realtime experience for globally distributed online musicians, prerecorded tracks (such as those served by Music in Wonderland) can be streamed for “concert-like” experience.

[Boustead et al. \(2005\)](#) consider server-side optimization of compiled soundscapes, including accommodation of limited bandwidth and soundscape compilation distribution to clients for load-sharing. For a perfect network, running at the speed of light, packets would take about 67 ms to get halfway around the world (“worst best case”), which delay would be fine for conversations, but probably distractingly audible for distributed performance.

4.3.4 Ethnomusicology of, and through, cyberworlds

Ethnomusicology can be defined as a branch of human sciences that studies music in its social-cultural contexts, especially the ways in which people interact through shared musical experience and discourse about music, and how music thereby facilitates the emergence of social groups and communities [Nettl \(2005\)](#). Methodologically, ethnomusicology centers on qualitative research, mainly ethnographic

¹highfidelity.io

fieldwork relying upon participant-observation and informal interview techniques [Fine \(2001\)](#); [Barz and Cooley \(2008\)](#). Variables typically cannot be controlled.

Cyberworlds open new avenues for ethnomusicological research. A cyberworld is a social space, with important ramifications for real social interaction and culture-formation, and thus of tremendous concern to many scholars working in the social sciences and the humanities [Kong \(2001\)](#); [Taylor \(1997\)](#). As social cyberworlds incorporating music become increasingly prominent, the task of studying them falls to ethnomusicology. The ethnomusicologist seeks to comprehend social dimensions of musical cyberworlds, to enhance their musical functions, and to further understand music in social-cultural contexts more generally, since cyberworlds are closely related to the real world, and impact it strongly.

Now it is not only possible to build a cyberworld as the focus for ethnomusicological research, but necessary as well, since cyberworlds represent contemporary musical reality. Musical cyberworlds can enable a new paradigm for ethnomusicology. Instead of observing musical interactions in the world-as-encountered, one can study a virtual world whose parameters are, to a great extent, under the researcher's control. Such a cyberworld becomes a laboratory for ethnomusicological research, a means of better understanding other musical cyberworlds, and providing, for the first time, a controlled environment for the ethnomusicological study of virtual community.

4.4 Exploring Folkways in Wonderland

In order to enter our cyberworld a user must connect to our publicly available world hosted over the internet, using a web browser. By clicking on the Launch

4. Exploring Music in Wonderland

button, client software is downloaded automatically and the user is asked to provide credentials. After successful login, one can explore music in multiple ways, including placemarks and bookmarks (visually), entering a track’s sonic sphere (auditorily), and/or discussion (recommendations etc.) with another user (socially). The system is collaborative: multiple avatars can enter the space, listen to track samples, and contribute their own sounds (typically speech) to the mix via voice chat. Avatars hear all sound sources (musical tracks and sounds produced by other avatars) within the space, attenuated for distance, and combined according to a spatial sound engine that emulates real-world binaural hearing. Avatar-represented users are free to explore the cyberworld (as shown in Figure 4.5(a)), using keyboard and mouse-like controls to navigate through the surrounding virtual environment (including a building and a verdant park, as shown in Figure 4.5(b)), while interacting with one another and listening to music. Wonderland supports multiple perspectives, including endocentric (1st person), egocentric & chase camera (2nd person), and front camera (reflecting) points of view. Other interesting features of FiW can be described as follows:

Metadata: Track metadata is displayed in a separate pop-up window in a conventional way outside the VE (as shown in Figure 4.3). Since an avatar in Wonderland always stands in the middle of the client window, a HUD (Head Up Display) with periodically changing text and images would distract the user [Fadden et al. \(1998\)](#). Along with common information (track title, author, album, genre, origin, instruments used, etc.), track and album IDs are also displayed for cross-referencing purposes. An offline “web scraper” (or spider) grabs metadata from the Smithsonian site, ensuring that the displayed information is up to date.

Placemark: Each track is visually identified by a placemark, a clickable sphere

4. Exploring Music in Wonderland

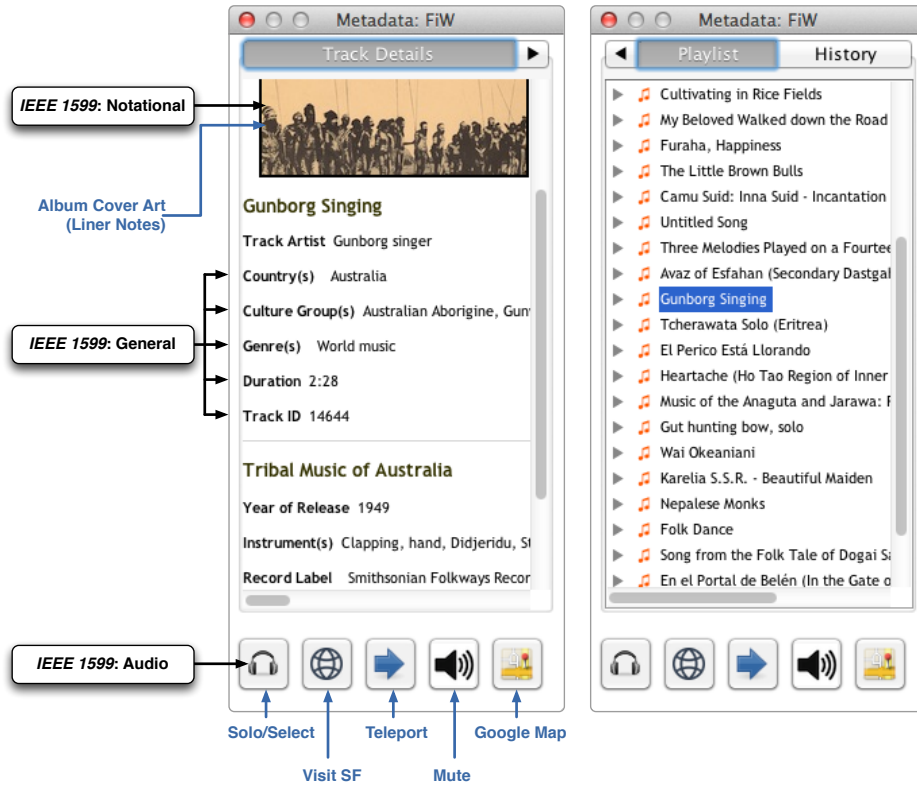


Figure 4.3: Metadata display window: The ‘Track Details’ tab shows song information, the ‘Playlist’ tab displays the entire collection as an outline, and the ‘History’ tab lists tracks visited by the user. Other operations, invoked by buttons at the bottom, allow exclusively auditioning a track, browsing selected track information at the Smithsonian Folkways site, teleporting to the origin of a track, muting a track, and ‘opening map window’ with a Google map to provide detailed, zoomable, topographic information. Clicking on album art brings up liner notes, which may include scores, musician interviews, critical commentaries, etc.

acting as a landmark. Once clicked, the sphere changes color (from orange to red), the track’s album cover appears above the selected node, the metadata window is updated, and a menu item (similar to a bookmark in web browser) is added to the placemarks list in the client browser. This red ball marks the track whose metadata is displayed in the popup window. When a different track is selected,

4. Exploring Music in Wonderland

the previous selection reverts back to its original mode (back to orange). One can also click on the ‘teleport’ button in the Metadata window to automatically move the avatar to a respective placemark.

Auditory focus: When tracks are near each other, overlapping projections create a dense mix, which is appropriate when exploring an entire collection by moving one’s avatar among distributed tracks. In order to listen to a particular track, an auditory focus function causes all other audio streams, including voice chat channels, to be blocked, providing auditory privacy. The `solo` or `select` function, iconified by a headphone symbol, is auto-released when a different track is played or corresponding button is pressed.

Ambient music: One can hear ambient music selected according to the collection when one’s avatar enters a conical gazebo in the center. Hidden virtual speakers (sources) have also been arranged to diffuse earth sounds such as water-waves and wind.

Tour: When joining a tour, an avatar simply relaxes and is led around the space automatically. Gravity is set to ‘off’ and camera perspective is set to ‘chase’ mode, which follows the avatar from above and behind. A tour consists of predetermined tracks. The metadata window is updated with information about each currently playing track. One can simply pause a tour to linger over a compelling track. Overlapping sonic zones during a tour would be distracting. One solution would be to adjust the sonic sphere radii according to density of tracks [Stewart et al. \(2008\)](#), and another would be donning the virtual “`solo`” headphones during the tour. Users should not have to turn shared audio on and off individually as there would result a kind of audio chaos in the VE—someone turning off what another had just turned on. Our novel solution is to allow the user to adjust

his/her own soundscape locally, “narrowcasting,” described in §5.1.1.

Textual search: A simple text search is provided to search across all track/album metadata associated with a given track and metadata associated with a container object (album). Users can also focus location search by specifying center (latitude and longitude) or country of origin.

4.4.1 Architecture and Implementation

4.4.1.1 Basics of Wonderland

Wonderland uses a client-server model [Crisostomo et al. \(2004\)](#) with various networking protocols for different data types [Kaplan and Yankelovich \(2011\)](#). TCP (Transmission Control Protocol) is used for communicating object properties and positions, while SIP (Session Initiation Protocol) & RTP (Real Time Protocol) are used for audio communication. The Wonderland suite integrates several services that can be distributed across multiple machines to increase scalability [Gardner et al. \(2011\)](#). The Darkstar game server [Waldo \(2008\)](#) provides a platform for Wonderland to track the frequently updated states of objects in the world. JVoiceBridge, a pure Java audio mixing application, communicates directly with the Darkstar server, providing server-side mixing of high-fidelity, immersive audio [Kaplan and Yankelovich \(2011\)](#). Modules are the mechanism for packaging code, artwork, and other resources to add new capabilities to Wonderland. When deployed, a module is unpacked into a directory in the web server. Client code is available to clients via the web server, and server code is installed in the Darkstar server [Kaplan and Yankelovich \(2011\)](#).

As shown in Figure 4.4, an Open Wonderland Client (OWLClient, near the

4. Exploring Music in Wonderland

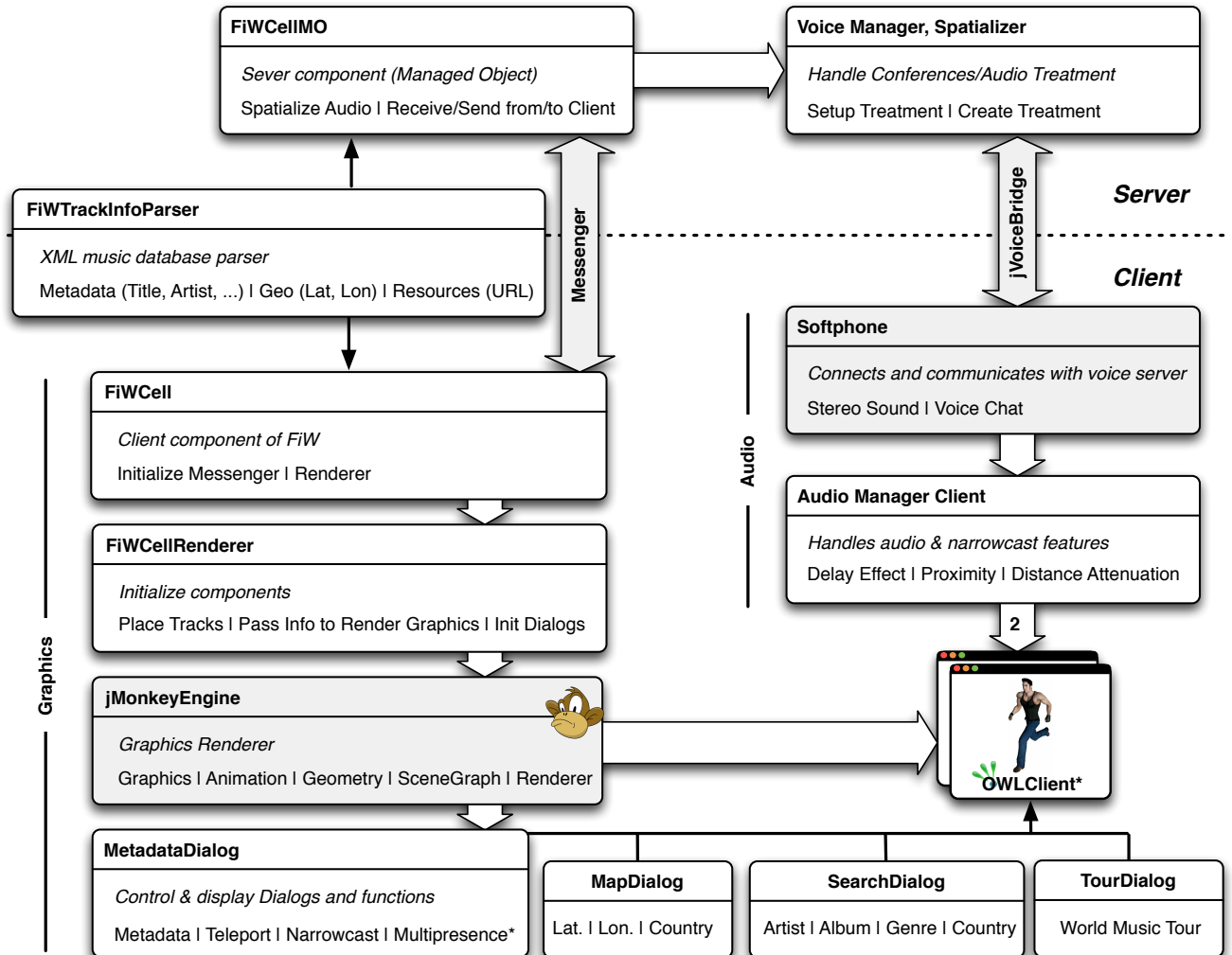


Figure 4.4: FiW system schematic (main components involved at runtime)

lower right of the diagram) connects to a server for messaging with other clients, while audio mixing is performed at `jVoiceBridge`, which is built into the Wonderland server (top). The `FiWCellIMO` server (top left) generates a list of track samples by parsing an XML database using a shared `FiWTrackInfoParser` when the server starts up. A `Spatializer` (top right) creates spatialized audio for the track list, and `VoiceManager` (also top right) is responsible for handling audio communica-

tion between clients. A `Softphone` (right upper center) at each client connects to the voice server. An `AudioManagerClient` (right lower center) controls stereo audio and provides narrowcasting operations for each client. `FiWCellRenderer` (left lower center), also using `FiWTrackInfoParser` (upper left), generates a list of track samples at the client side. The generated track list is used for rendering track markers with `jME`, displaying metadata of a selected track, and searching for particular keywords. The `FiWCellRenderer` is also responsible for rendering Java2D Swing dialogs, including those for Metadata, Map, Search, and Tour. (Thin vertical black arrows in the schematic represent inclusions, thick horizontal white arrows indicate API invocations, and thick vertical gray arrows denote communication. The modules drawn with white background were extended or newly developed in `FiW`.)

4.4.1.2 Graphics Rendering using `jME`

A cell system provides the structure of the virtual world. A cell is a volume in the 3D world described as either a bounding box or sphere, which collection is arranged into a tree data structure. The tree provides a spatial organization and requires that the bounds of any child cell be fully enclosed within the bounds of its parent. Cell Renderers facilitate the ability to render 3D objects using `jME` (jMonkey Engine), a high performance scenegraph-based graphics API.¹ Each `jME` Cell Renderer provides a scene graph which is added as a child of an “attach point” node in the Cell Renderer of its parent cell, so each child cell can inherit rendering state from the parent scenegraph [El Rhalibi et al. \(2009\)](#). (In contrast,

¹www.jmonkeyengine.com

“O3D,”¹ Google’s 3D engine, creates both a transform graph and a render graph instead of monolithic scenegraph.) The populated cylinder is a geometric 3D model rendered by jME; music tracks are children nodes of the cylinder. Each track is also represented as a 3D model with the respective album cover image set as the texture when the corresponding marker is clicked, positioned according to its geotag (latitude and longitude).

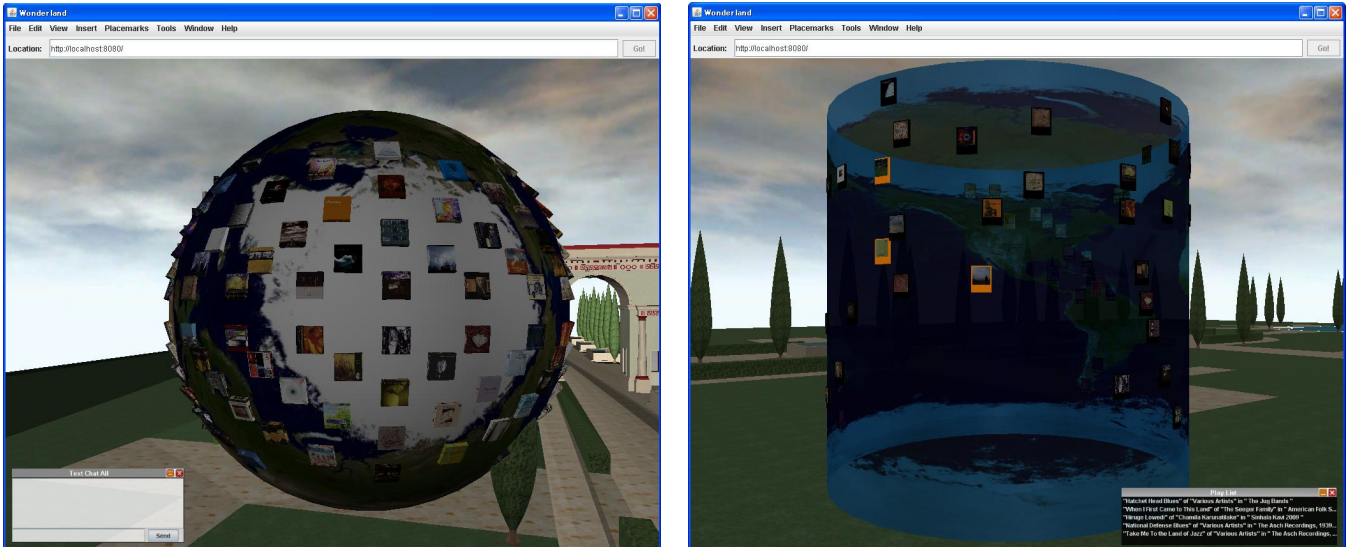
4.4.1.3 Arranging Music Tracks in 3D Space

FiW enables music browsing in a multidimensional, multimedia collaborative virtual environment (CVE) rather than through textual lists, enabling a sonic, social, and spatial experience. Both a sphere and a cylinder are structurally well suited for representing collections in 3D space because they can be tiled in a variety of ways, and symbolic representations of content can be projected onto nodes or points of intersection of curve segments from the tessellation [Stewart et al. \(2008\)](#). Since Wonderland does not permit recumbency, an avatar always stands vertically. A sphere limits angle of view more and more as objects approach the poles (as shown in [Figure 4.5](#)), but on the other hand a cylinder can provide a readily navigable structure onto which symbolic representations of collection content can be easily mapped.

The FiW music browser is, therefore, a cylinder upon which a rectangular world map is texture-mapped. In general, flat maps are derived by projecting a sphere onto a plane, with associated transformation of spherical to rectilinear coordinates. A flat map can preserve only certain features of the globe (e.g., direction, distance, area, or shape), while necessarily distorting others. For po-

¹code.google.com/p/o3d/wiki/GettingStarted

4. Exploring Music in Wonderland



(a) A sphere shaped arrangement of tracks which limits view near the poles. (b) Cylinder shaped arrangement for more navigational space

Figure 4.5: Different layouts of Folkways in Wonderland

sitioning tracks in a cylindrical world map, a suitable projection technique was chosen from the set of possible projections, including Mercator, Miller, Lambert, Behrmann, etc. We selected the Mercator projection [Fitzsimons \(2006\)](#), a rectangular map preserving rhumb lines but distorting distances and areas, especially near the poles. While modern cartographers have rightly critiqued this projection, the cylinder necessitates a rectangular map. The Mercator projection also harmonizes with Google Maps [Shaomei et al. \(2010\)](#), with which we hope to immersively integrate in a future release. Currently the cylinder is texture-mapped with a static image, but a more sophisticated deployment would feature scalable resolution by dynamically reloading maps. Finally, as few Folkways tracks are geotagged near the polar regions, Mercator distortions do not affect the spatial representation of music too adversely.

4.4.1.4 Music Collection Parsing

As previously mentioned, metadata of all tracks is stored in an XML file. XML (Extensible Markup Language) is an open standard by the W3C for interoperable unicode documents Lam et al. (2008). Processing of XML occurs in four steps: parsing, access, modification, and serialization. Various implementations of XML parsers use different models to extract data representations Lam et al. (2008). For example, DOM (Document Object Model) creates a tree object, VTD (Virtual Token Descriptor) creates integer arrays, and SAX (Simple API for XML) and StAX (Streaming API for XML) create sequences of events. Although parsing is the most expensive operation, by comparing performances of different parsers (based on Lam et al. (2008)'s XML processing performance characteristics) and choosing a suitable parser, namely VTD, we minimize performance overhead and best leverage the advantages of MX: IEEE 1599.

4.4.2 MX: IEEE 1599

Artistic, geographic, audio-related, and generic information describing the Folkways music collection is curated in XML format conforming to MX:IEEE 1599 Baggi and Haus (2009), a comprehensive, multilayered music description standard. MX, standing for musical application using XML, inherits all the features of XML—including inherent human-readability, extensibility, and durability Ludovico (2009)—and unifies features of MML¹ (Music Markup Language, a syntax for encoding different kinds of music-related events) and MusicXML² (which is designed for the exchange of scores) with some additional features, including the

¹www.musicmarkup.info

²www.musicxml.com

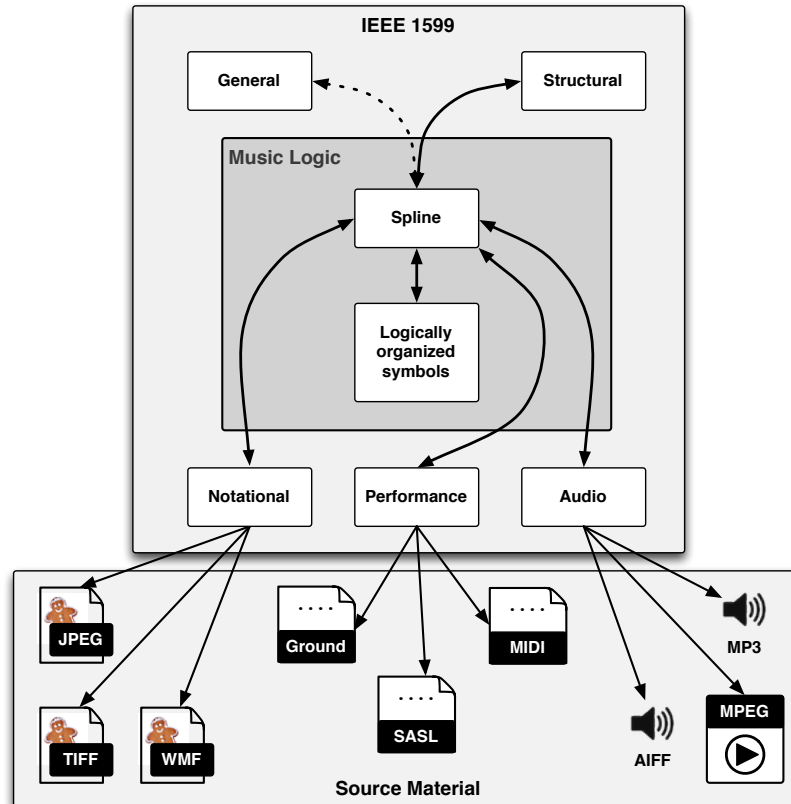


Figure 4.6: MX: IEEE 1599 layers (reconstruction of [Baggi and Haus \(2009\)](#)'s diagram of seven layers of a musical piece)

concept of layers. The six layers of the standard are as follows:

- *General* – music-related metadata, including title, author, date, genre, performance, and recording information (as shown in Figure 4.3 middle section)
- *Logic* – music description from a symbolic point of view, or score symbols
- *Structural* – identification of music objects and their mutual relationships
- *Notational* – graphical representations of the score (as shown in Figure 4.3 top section)
- *Performance* – parameters of notes played and sounds synthesized, specified by performance languages [Russo \(2008\)](#) such as MIDI

4. Exploring Music in Wonderland

- *Audio* – digital or digitized recordings of the piece (as shown in Figure 4.3 lower section)

Even though the Folkways curation has no information corresponding to the MX logical, structural, or performance layers, MX: IEEE 1599 allows empty layers [Ludovico \(2008\)](#), and there are no restrictions preventing browsing of other music collections when such information is available (as shown in this stub and reflected in the music browser as shown in Figure 4.3). Note that layers may contain URLs as well as directly accessed data, for extra flexibility and late binding. FiW keeps music-related data in an XML file conforming to the MX: IEEE 1599 multilayer structure as shown in this stub.

```
XML stub corresponding to the MX: IEEE 1599.
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ieee1599 SYSTEM "http://standards.ieee.org/downloads/1599/1599-2008/ieee1599.dtd">
<ieee1599>
  <track>
    <general>
      <albumTitle>Folk Music of Ghana</albumTitle>
      <trackTitle>Ataa oblanyo</trackTitle>
      <trackArtist>Various Artists</trackArtist>
      <instruments>Ashwa, Talking drum</instruments>
      <cultureGroups>Ewe</cultureGroups>
      <genre>World music</genre>
      <releasedIn>1964</releasedIn>
      <country>Ghana</country>
      <latitude>5.555717</latitude>
      <credits>Produced by Ivan Annan; Recorded ...</credits>
      <sourceArchive>Smithsonian Center for Folklife ...</sourceArchive>
      <trackDetailURL>http://music/trackdetail.aspx?itemid=xx</trackDetailURL>
    </general>
    <notational>
      <linerNotes>file:/music/folkways/notes/FW08859.pdf</linerNotes>
    </notational>
    <audio>
      <audioURL>file:/music/folkways/FW08859_03.30.mp3</audioURL>
    </audio>
    <logic/>
    <structural/>
    <performance/>
  </track>
  <track>...</track>
</ieee1599>
```

Chapter 5

Narrowcasting and Multipresence in Wonderland

5.1 Narrowcasting in Wonderland

5.1.1 Introduction and Related Work

“Narrowcasting” describes a technique which allows information streams to be filtered, for privacy, security, and user interface optimization in groupware solutions [Alam et al. \(2009\)](#); [Fernando et al. \(2006\)](#). Traditional conferencing systems over the PSTN (Public Switched Telephone Network) have become almost obsolete, as contemporary telecommunications systems support teleconferencing by providing audio, video, and data services by one or more means [Schooler et al. \(1991\)](#). A typical conferencing configuration consists of several avatars, representing distributed users, moving around a shared space with sources associated with each user’s voice, and sinks associated with users’ ears. Research has been

5. Narrowcasting and Multipresence in Wonderland

conducted to improve audio quality in conferences [Yankelovich et al. \(2006\)](#), address problems that impact effectiveness [Yankelovich et al. \(2006, 2004\)](#), and explore how audio improves communicative capability for interesting and useful shared media systems [Ackerman et al. \(1997\)](#). Analyzing existing conferencing systems, we pose the following questions: how does one effectively listen only to a particular member in a conference? How does one prevent his/her voice from being delivered to other members?

Even though narrowcasting operations have been implemented for workstations and mobile phones [Cohen and Györbiró \(2009\)](#); [Cohen and Kawaguchi \(2003\)](#); [Fernando et al. \(2003\)](#) or online chat systems such as Dolby Axon,¹ this project explores how narrowcasting operations can be effectively used when voice- or text-chatting and exploring music in enterprise-quality immersive virtual worlds. Most CVEs configure private conversations by just selecting a subset of connected members [Yankelovich et al. \(2005\)](#). Full-featured narrowcasting can be invaluable when listening to collections of music, especially when the music is spatialized. An obvious use case is to avoid unwanted cacophony when multiple tracks are too closely located, causing multiple nimbi [Greenhalgh and Benford \(1995\)](#) to overlap. Our voice bridge basically implements distributed modulation of source \rightarrow sink connectivities, as expressed by the predicate calculus expressions shown in [Figure 5.1](#).

¹axon.dolby.com

The general expression of inclusive selection is

$$\mathbf{active}(x) = \neg \mathbf{exclude}(x) \wedge (\exists y (\mathbf{include}(y) \wedge (\mathbf{self}(x) \Leftrightarrow \mathbf{self}(y)))) \Rightarrow \mathbf{include}(x).$$

So, for `mute` and `select (solo)`, the relation is

$$\mathbf{active}(\mathbf{source}_x) = \neg \mathbf{mute}(\mathbf{source}_x) \wedge (\exists y \mathbf{select}(\mathbf{source}_y) \wedge (\mathbf{self}(x) \Leftrightarrow \mathbf{self}(y))) \Rightarrow \mathbf{select}(\mathbf{source}_x).$$

`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

$$\mathbf{active}(\mathbf{sink}_x) = \neg \mathbf{deafen}(\mathbf{sink}_x) \wedge (\exists y \mathbf{attend}(\mathbf{sink}_y) \wedge (\mathbf{self}(x) \Leftrightarrow \mathbf{self}(y))) \Rightarrow \mathbf{attend}(\mathbf{sink}_x).$$

Figure 5.1: Simplified formalization of narrowcasting and selection functions in predicate calculus notation (Cohen and Fernando (2009)), where ‘ \neg ’ means “not,” ‘ \wedge ’ means conjunction (logical “and”), ‘ \exists ’ means “there exists,” ‘ \Rightarrow ’ means “implies,” and ‘ \Leftrightarrow ’ means “is equal to” (mutual implication). The duality between source and sink operations is tight, and the semantics are identical: an object is inclusively enabled by default unless, a) it is explicitly excluded (with $\underbrace{\mathbf{mute}}_{\text{source}} \parallel \underbrace{\mathbf{deafen}}_{\text{sink}}$, or, b) peers are explicitly included (with $\mathbf{select} \left[\underbrace{\mathbf{solo}}_{\text{sources}} \parallel \underbrace{\mathbf{attend}}_{\text{sinks}} \right]$ 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.

5.1.2 Conferencing using jVoiceBridge

Wonderland natively provides individually adjustable audio channels for each in-world, recorded sound source and live avatar. ‘jVoiceBridge,’¹ an open-source conferencing module, handles voice over IP (VoIP) audio communication for Wonderland. The voice bridge supports a conference by receiving monaural audio streams from all session members, directionalizing the streams into individualized soundscapes, and streaming a personalized stereo mix back to each member. Users can adjust their virtual speakers and microphones as well as other participants’ apparent intensities and sensitivities. A user mutes him/herself by effectively setting others’ virtual source-wise sensitivities to zero. By extending such built-in capabilities, narrowcasting features were implemented in Wonderland for voice-chat and music audition, considered as separate modalities.

5.1.3 Audio Transmission for Voice and Music

It is important to keep in mind the entire end-to-end pathway of sound in a teleconference, including careful use of often muddled jargon. A speaker’s voice, causing compression and rarefaction of air, is sensed by a microphone that measures pressure, transducing acoustic energy into electrical. This measurement, expressed as a voltage, is sampled (in time) and quantized (in amplitude) by an audio interface, converting the analog acoustic phenomenon into a digital signal, encoding it perhaps uniformly (as in PCM, pulse code modulation) or perhaps nonlinearly (as in μ - or a-law representations). This audio signal is filtered by the computer’s DSP hardware and software, including amplification (increased

¹java.net/projects/jvoicebridge

5. Narrowcasting and Multipresence in Wonderland

envelope size) and attenuation (decreased scale). The amplification or attenuation of a signal can be accomplished by adjusting its gain, a scalar coefficient which multiplies the raw signal and controls the dynamic range. (Balancing or panning a stereo signal involves coupled gain adjustments to a left–right signal pair.) Frequency-based adjustments— such as equalization, “sweetening,” aural enhancement, etc.— are also possible, typically by specifying frequency-band-specific amplifications or attenuations.

5.1.3.1 Protocols: Control, SIP, and RTP

The `jVoiceBridge` uses several protocols when sending and receiving audio over IP. Text-based proprietary commands (only ASCII) are used to set up and control calls. SIP is used to initiate audio connections between the voice bridge and other endpoints, such as softphones or a SIP IP phones. RTP is used to stream audio data mixed at the bridge to members of a conference.

5.1.3.2 Calls, Treatments, and Softphone

A call is the basic unit of audio. Each call has `MixDescriptors` which describe what audio streams the call should hear. A descriptor has information about the source of the audio and how loud and in what direction the call should hear the audio. A “treatment” is a special type of call which is simply an audio file that’s played to a call or a conference (as described in Figure 5.5(a)). Wonderland allows one to play music tracks (as a file uploaded to the server or URL) publicly using an `AudioTreatmentComponent`. These tracks can not be controlled locally or individually without affecting the soundscape of other participants in the session. `Softphone` is the client-side Java telephony application used to connect and

communicate with the voice bridge.

5.1.4 Binaural Parallax: Localizing Sound

Binaural sound enables an immersive listening experience, giving one a sense of presence and space [Bormann \(2005\)](#); [Gilkey and Anderson \(1997\)](#). Wonderland native audio spatialization supports stereo (two channels) and only horizontally flattened positioning, including delay effects (ITD, interaural time difference) and panning (IID, interaural intensity difference). Two interleaved channels are sent in each packet, one delayed by 0–0.63 ms (Equations 5.1 & 5.2), depending on the location of the source relative to the respective sink.

$$\text{ITD} = \frac{a}{c}(\theta + \sin \theta), -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}. \quad (5.1)$$

Approximating the radius of a head a as 8 cm and the speed of sound c as $300 \text{ m}\cdot\text{s}^{-1}$, maximum ITD can be estimated as

$$\text{ITD}_{\max} \approx \frac{0.08}{300} \left(\frac{\pi}{2} + 1 \right) \approx 0.6 \text{ ms}. \quad (5.2)$$

An RTP packet contains 20 ms (50 packets/s) of audio data. For μ -law encoding as used in jVoiceBridge, a packet comprises 160 8-bit samples ($8 \text{ kHz} \times 1 \text{ sample/s/Hz} \times 0.020 \text{ s}$) with a 12-byte RTP header, totally 172 bytes of data (with $20 \log 2^8 \approx 48 \text{ dB}$ dynamic range). For PCM stereo streams sampled at 44,100 Hz, there are $44100/50 \times 2$ 16-bit samples/packet, which is 3528 bytes of data payload (with $20 \log 2^{16} \approx 96 \text{ dB}$ dynamic range), plus the RTP header. Overhead for each audio packet, besides the 12-byte RTP header, includes 8 bytes for the UDP header

5. Narrowcasting and Multipresence in Wonderland

and 20 for the IP header Seo et al. (2010). After whatever local processing, the audio signal is packetized and sent over the internet to one or more distal nodes, corresponding to other participants in a session. These packets are resequenced by receiving nodes, allowing the reconstructed audio signals to stream through their own respective filters. The user interface or distributed processing might allow articulated control, featuring, for simple example, both source-wise and local computer audio amplification adjustment (colloquially called “dynamics,” “loudness,” or “volume” [not to be confused with 3-dimensional extent]). One could change, for instance, the loudness of each of the conferees individually, as well as amplification of the entire mix.

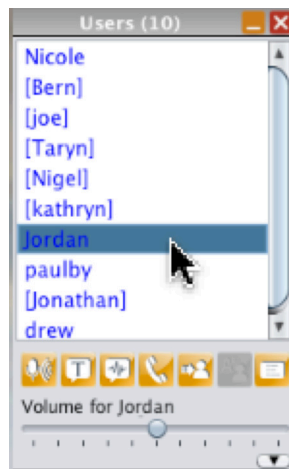


Figure 5.2: Original Wonderland voice-chat HUD (Head Up Display) panel with basic audio functions, including mute and individually adjustable volume for each user and self

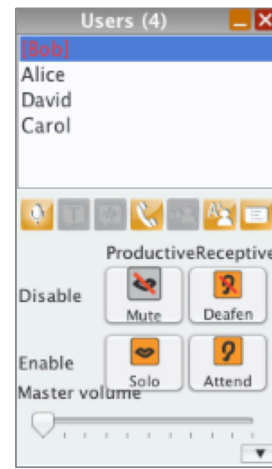


Figure 5.3: Extended FiW voice-chat panel indicating current status and matrix of narrowcasting operations {disable (exclude), enable (exclusively select, selectively include)} \times {production (sources), reception (sinks)}

Finally, at the audio interface of each attending computer, the digital signal is pumped through a DAC (Digital–Analog Converter), presented to an amplifier as

5. Narrowcasting and Multipresence in Wonderland

a continuous (in time and amplitude) voltage, which power signal in turn drives speakers, such as stereo loudspeakers or headphones. The ears and the listener’s brain have their own complicated physiological (biological reception) and psychoacoustic (psychophysical stimulus–sensation behavior) response to the sound, which is perceived, interpreted, and “heard.” Qualities such as amplitude of an auditioned signal are therefore a combination of source-side processing (including muting and softer gain adjustment such as “muzzle”) and sink-side processing (including deafening and softer gain adjustment such as “muffle”). The intensity of a signal corresponds to its instantaneous power, or energy/time, the time-averaged square of its running value. For a digital signal, intensity is the (windowed and averaged) sum of the squares of the samples. The “level” is proportional to the logarithm of the intensity or RMS (root mean square, the standard deviation of a centered [zero-mean] signal), associated with a channel’s subjective loudness or volume, since human perception of loudness is approximately logarithmic with power. Distance attenuation in the voice bridge is shown in Figure 5.4 and expressed by Equations 5.3 & 5.4.

$$d' = \frac{d - d_{fvs}}{d_e - d_{fvs}}, \quad (5.3)$$

$$\text{gain}(d) = \begin{cases} 1 & d \leq d_{fvs} \\ \text{falloff}^{d'} & d_{fvs} < d < d_e. \\ 0 & d_e \leq d \end{cases} \quad (5.4)$$

In contrast, the ordinary free-field inverse-square intensity attenuation for a nominal point source (representing a constant power source radiating acoustic power

5. Narrowcasting and Multipresence in Wonderland

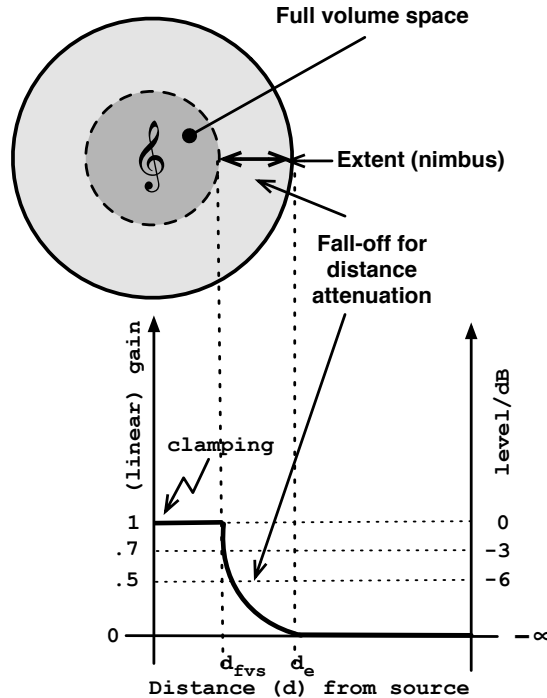
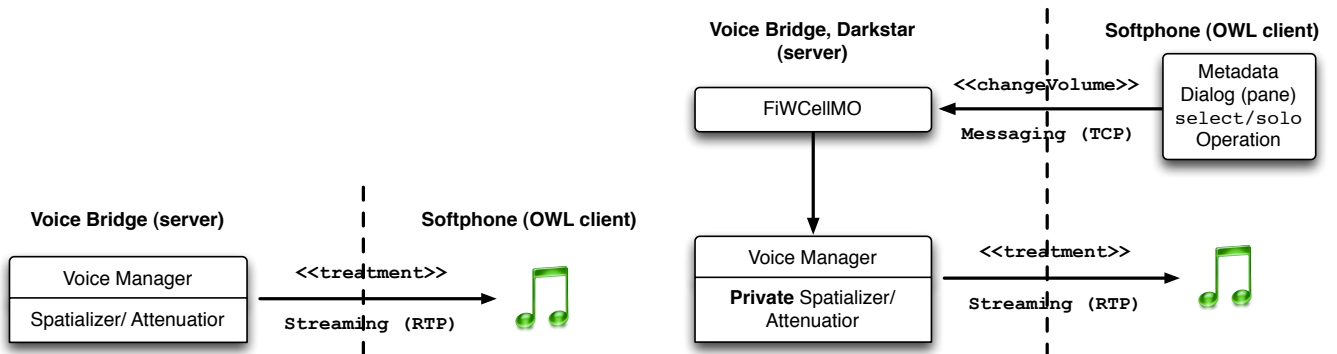


Figure 5.4: The ‘full volume space’ (d_{fvs}) of a source is represented by the dotted circle in the diagram, within which audio is heard at full volume. The ‘extent’ or nimbus (d_e) of the displayed source refers to the space in which sinks can receive the signal.



(a) Originally all clients can hear an audio treatment, but (b) Sending a message to FiWCellMO and setting a private spatilizer allows a client to mute treatment(s) without disturbing others’ soundscapes.

Figure 5.5: Audio treatment architecture in Wonderland

5. Narrowcasting and Multipresence in Wonderland

P through a notional sphere with area proportional to the square of its radius, which corresponds to source-to-sink distance d) is $I = P/4\pi d^2 \propto d^{-2}$. This is a power law relation, in contrast to the exponential function implemented by the voice bridge model and expressed above.

Each avatar is both a source (projecting a “nimbus,” as described in [Greenhalgh and Benford \(1995\)](#)) and a sink (sensing within a focus), but music tracks are only sources, not sinks. This means that avatars can be **deafened** or **attended**, but tracks cannot. However any source can be **muted** or **selected**. Furthermore, only avatars can issue such narrowcasting commands, as tracks lack an associated user with agency or volition. The narrowcasting commands are considered separately in the following paragraphs, $\{\text{source} \Rightarrow \text{nimbus}, \text{sink} \Rightarrow \text{focus}\} \times \{\text{disable}, \text{enable}\}$.

5.1.4.1 Mute (Nimbus Disable): $A \rightarrow \bar{B}$

Mute is a common media control function, available in most conferencing systems. Narrowcasting **mute** is a source-related command which blocks media coming from a source. Wonderland allows muting oneself as well as blocking audio from other participants in a conference by adjusting a sensitivity controller (as shown in [Figure 5.2](#)), as well as muting by an administrator like public branch exchange [PBX] mute [Alam et al. \(2009\)](#). But such functionalities are not available for individual music audition. Users can share audio by uploading audio files to a Wonderland server or accessing audio sources on the internet via URLs, dynamically adding new tracks to a mix. One might turn such shared samples on and off individually, but that would result a kind of audio chaos, someone turning off what another had just turned on. To overcome such complications, a new capability was devel-

5. Narrowcasting and Multipresence in Wonderland

oped for the FiW music browser to allow a user to turn individual tracks off and on without affecting others' soundscapes. Narrowcasting `mute` can be compactly expressed as in $A \rightarrow \bar{B}$ (with a minus sign stacked above a symbol for a source) or $A \rightarrow [B]$ (with surrounding square brackets), for A muting B .

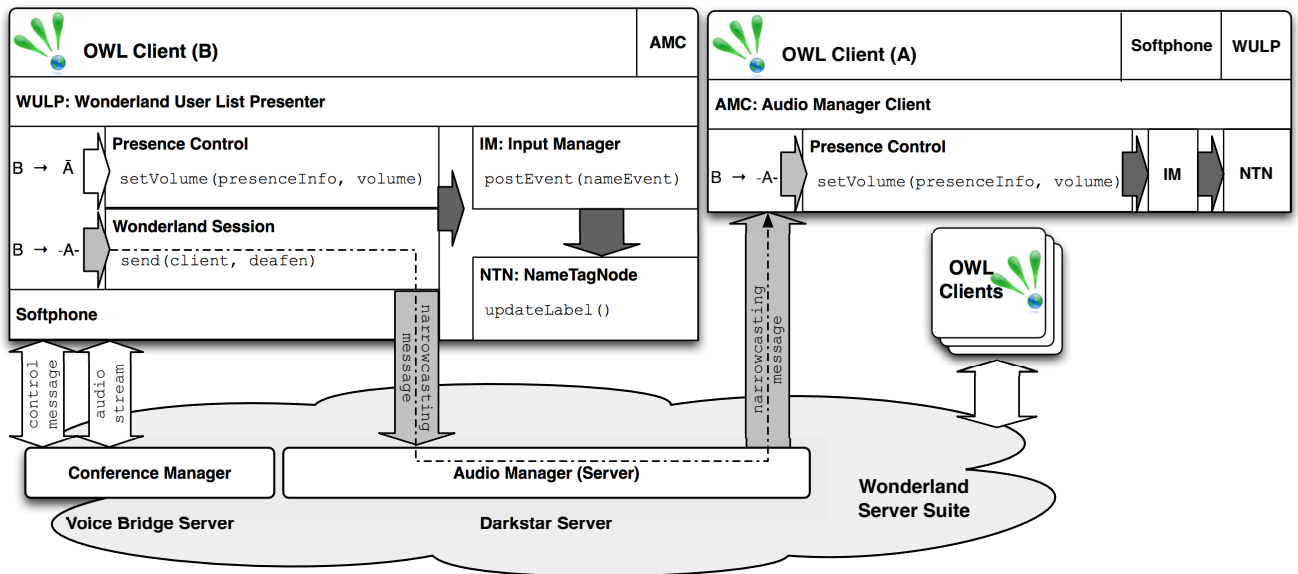


Figure 5.6: Extended voice-chat architecture: An Open Wonderland (OWL) client consists of a **Softphone** (middle left) which communicates with the **Conference Manager** in the Voice Bridge server, the **Wonderland User List Presenter** (WULP, top left), which allows one to adjust other participants' apparent sensitivity locally, and the **AudioManagerClient** (AMC, top right), which is the message receiver for control signals sent from the **AudioManager** in the Darkstar server. `Mute` and `select(solo)` are implemented by adjusting sensitivity in **PresenceControl** in the WULP. By sending a **Narrowcast** message via **AudioManager**, a receiver (AMC at a different client) can `mute` a sender, realizing `deafen` and `attend`.

5.1.4.2 Select or Solo (Nimbus Enable): $A \rightarrow \bar{B}$

While exploring a collaborative musical space, users might hear a mix of musical tracks when multiple tracks are nearby. By `muting` tracks individually, one can

5. Narrowcasting and Multipresence in Wonderland

focus on a particular track. Alternatively, narrowcasting `select` (a.k.a. `solo`), a source-related command which limits projected sound (the nimbus) to particular sources, can be invoked to avoid unwanted cacophony. By extending the `mute` function, the `select` function resets all projection amplifications as described by [Greenhalgh and Benford \(1995\)](#) to zero, excepting a selected track. The `select` or `solo` operation can be abbreviated as in $A \rightarrow \overset{+}{B}$ (with a plus sign above a symbol for a source) or $A \rightarrow]B[$ (with square brackets facing outwards), for A soloing B .

5.1.4.3 Deafen (Focus Disable): $A \rightarrow -B-$

`Deafen` is a sink-related command which blocks media going to a sink. Such a function could be useful to a virtual ethnomusicologist [Cooley et al. \(2008\)](#) for private discussions with colleagues, excluding, for instance, casual tourists. `Deafen` can be compactly represented as in $A \rightarrow -B-$ (with minus signs straddling a symbol for a sink), for A deafening B . A `deafen` operation can be realized in different ways in a client-server architecture:

- A source can block outgoing media to particular sinks.
- A crossbar matrix mixer at the soundscape or media server can block streams to a particular sink.
- A sink can block incoming media from particular sources.

We have selected the last method, exploiting the duality relationship between `mute` and `deafen` ($A \rightarrow -B- \equiv B \rightarrow \bar{A}$). This simplicity can be useful when implementing narrowcasting functions in any conferencing system, because all the functions described here can be elaborated just from basic `mute`. For example,

5. Narrowcasting and Multipresence in Wonderland

if user A wants to **deafen** B , a **Narrowcast (Deafen)** message is sent from A to B and B sets A 's apparent intensity to zero (as traced by the stippled line in Figure 5.6's component diagram).

5.1.4.4 Attend (Focus Enable): $A \rightarrow +B+$

Attend is a sink command which limits received media streams to only those explicitly apprehended by particular sinks. When user A wants to **attend** B , so that only B can hear A , all except B are **deafened** to A . **Attend** can be expressed as in $A \rightarrow +B+$ (with plus signs straddling a symbol for a sink), for A attends B .

5.1.5 Textual Narrowcasting

As previously mentioned, besides voice-based communication, Wonderland provides public and private text-based chatspaces. Voice and text-chats have respective merits: [Geerts \(2006\)](#); [Yankelovich et al. \(2005\)](#) text-chat has advantages regarding simplicity and bandwidth considerations, whereas voice-chat is richer and more natural. In order to provide a more articulated experience for textual conferencing in FiW, narrowcasting features were added to the native chat system. When a source is **muted**, text typed by the user associated with that source's avatar is blocked. When a **selection** exists, all text is blocked except that from the selected source(s). Text is not delivered to **deafened** sinks and is delivered only to particular sinks when any are **attended**.

5. Narrowcasting and Multipresence in Wonderland












Representation	Exclude (Disable)		Include (Enable)	
	Mute	Deafen	Select	Attend
Figurative icons or symbols are used in metadata window (as shown in Figure 4.2) and voice-chat panel (as shown in Figure 5.3).				
Distinctive border colors are used when a track (as described in § 4.4) is muted or selected.				
Combinations of distinctive colors and glyphs decorate monikers in the voice-chat panel (as shown in Figure 5.3).	[Alice] Alice (Mute)	-Bob- Bob (Deafen)]Carol[Carol (Select)	+David+ David (Attend)
Figurative avatars with decorated names floating over head indicate applied narrowcasting operations.				
 <p>FiW Control Panel: Users can toggle preferred narrowcasting decorations (or even disable such display completely). Asserting “Arrange” resizes and places popped-up windows automatically (as in Figure 4.2). Text search and tour windows can also be opened from here.</p>				

Table 5.1: Different methods for representation of narrowcasting operations

5.1.6 Narrowcasting State Representations

When the audition interface is extended using narrowcasting features, complementary visual cues [Lee and Kim \(2008\)](#) should support it without many distractions or complications. The symbology for narrowcasting operations is complicated by the conflation of associations of source and sink icons. As illustrated by [Figure 5.7](#), the chain of simplex transmission for voice-chat starts with a source, a mouth, emitting sound which is sensed by a sink, a microphone, conveying the signal through the network for reproduction by another source, a loudspeaker, which signal is captured by another sink, an ear. Historically, a slashed speaker icon is used for mute and a headphone icon (connoting private listening) is used for `select (solo)`, but such associations don't generalize to soundscape models with sinks. In contemporary conferencing equipment, "mute" usually means local reticence, like a video "sneeze button": distal parties' voices are still audible, but the muting side has "put its hand over the mouthpiece." [Table 5.1](#) shows how narrowcasting operations are represented across different situations in our cyberworld.

Narrowcasting is multimodal, applicable to music, voice, and text streams. Our system supports two different kinds of sound sources: musical selections and

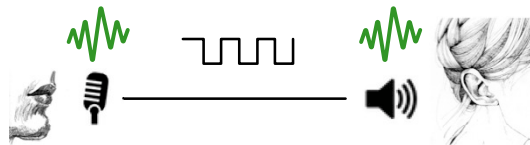


Figure 5.7: Voice transmission from human source to human sink over network: source (mouth) $\xrightarrow{t_o}$ sink (microphone) $\xrightarrow{t_o}$ source (speaker) $\xrightarrow{t_o}$ sink (ear).

5. Narrowcasting and Multipresence in Wonderland

avatar conversation (“voice-chat”). Narrowcasting for music enables aesthetic focus; narrowcasting for talk, like that for text-chat, enables cognitive focus. The former is required for dense presentation of musical sound, the latter for virtual worlds in which many avatars are expected to be able to interact. Music tracks and voice-chat are controllable by narrowcasting functions. In the FiW user interface, narrowcasting widgets (controls and displays) are distributed across two panels—solo and mute buttons in the metadata pop-up window, and {source, sink} \times {include, exclude} (as seen in Figure 5.3) in the user head-up display, a floating panel inside the main view window. Narrowcasting state is also displayed with the respective source or sink, as shown in Figure 5.3.

Narrowcasting controls are grayed-out when narrowcasting is disabled. Such inhibition is currently established locally, by each user (rather than a super-user administrator or server custodian). This narrowcasting enable/disable switch (as seen in the last row of Table 5.1) was deployed in order to test the usefulness of the narrowcasting privacy suite, so that we could instruct experimental test subjects to disable it before performing some task, and then re-enable it before performing similar tasks. That is, the narrowcasting absence or presence is an optional experimental condition.

Nimbus, or auditory extent of sources, is visualizable as a translucent boundary (as seen in Figure 5.8), enabled via an extensibility module capability of Wonderland. We generalized its heretofore rigid parameters, and it is now adjustable on both the server and the client sides. (However, the focus, or sensitivity of the sinks, is just a degenerate notional point inside the respective avatars’ heads.)

The audio server has a virtual crossbar matrix for a session’s sources and sinks. Distributed narrowcasting state, controlled by the respective clients, is

5. Narrowcasting and Multipresence in Wonderland



Figure 5.8: Translucent spheres represent nimbi of sources. Overlapped sections visualize overlapped sonic spheres where cacophony is present.

compiled into a Boolean audibility attribute for each source–sink combination. Each client streams its voice channel to the server, which otherwise manages all the Folkways musical sources (“treatments” in Wonderland jargon). The respective source–sink spatial arrangements, including direction and range, extended by multipresence and disambiguated by autofocus (described below in § 5.2.2), is used to parameterize sound spatialization, as the server renders 2-channel (binaural) contributions for each musical source for each client, which composite stereo pair is streamed back to the terminals, the respective clients. At the client side, local volume is effectively multiplied by the distributed audibility flags (bits), and the respective source stream mixed or not into the locally displayed composite soundscape.

It would have been more elegant to inhibit transmission of inaudible streams at

5. Narrowcasting and Multipresence in Wonderland

the server side, rather than depending upon each respective client to ignore them, since doing so would conserve network bandwidth and also prevent eavesdropping. However, the audibility attribute is separate from volume control, so there is no way a client could override such discretion (without meta-techniques such as editing and recompiling client software), and engineering considerations (details of Wonderland architecture) trumped logical economy.

5.1.6.1 Figurative Icons

Icons and symbols as part of GUIs play a significant role in user experience [Marcus \(2003\)](#). The extended FiW voice-chat and music browser functions use carefully designed icons for narrowcasting operations. Mute is a complex operation to represent, as a speaker icon is used in operating systems and media players, but a microphone icon is used in contemporary messenger systems such as Skype,¹ LINE,² and Google Talk.³ Three types of mute icons are used in our cyberworld:

- A slashed microphone iconifies Wonderland voice-chat mute.
- Narrowcasting `mute`, which was previously figuratively represented as hand clapped over a mouth [Cohen \(2000\)](#), is now iconified by a slashed mouth.
- As in many media players, muting a track is represented as a slashed speaker icon.

Previous research [Cohen \(2000\)](#); [Fernando et al. \(2006\)](#) figuratively used a megaphone to represent `select (solo)` and ear trumpets to represent `attend`. In our cyberworld, a headphone icon is used to represent `solo` for music auditioning,

¹www.skype.com

²line.naver.jp/en

³www.google.com/talk

5. Narrowcasting and Multipresence in Wonderland

as donning a headphone naturally implies auditioning a particular sound source privately. Narrowcasting *deafen*, formerly represented as hands clasped over ears, is now iconified by a slashed ear.

5.2 Multipresence in Wonderland

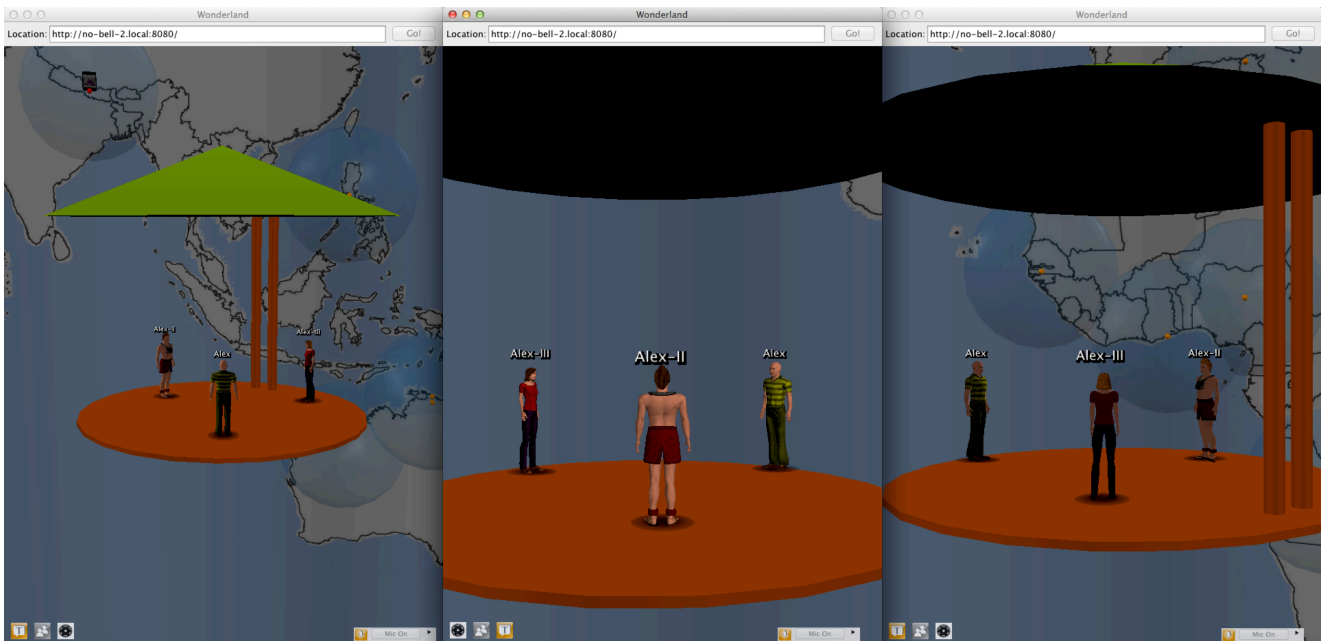


Figure 5.9: Disambiguating multipresence: To acquire control of a self-identified avatar, a user should select one of the Wonderland windows. Confusion about “Who am I now?” can be intuitively resolved, since an avatar in Wonderland always stands in the middle of its client window, even across different perspectives. Users can manifest different identities (Alex-II: the Mohawk-coiffed avatar in the center window) or even change apparent gender (Alex-III: in the right window).

Multipresence allows each user in a virtual environment to have presence in several places or spaces at once by designating multiple representatives as ‘self,’ effectively increasing one’s attendance in groupware activities [Cohen and Fernando \(2009\)](#); [Taylor \(1999\)](#). Multiple sources are useful for broadening one’s

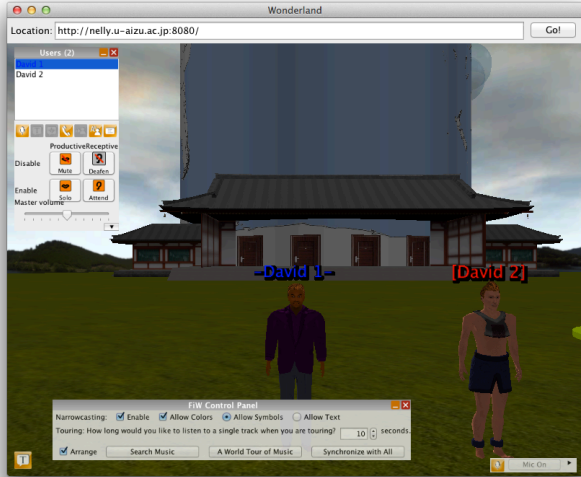
5. Narrowcasting and Multipresence in Wonderland

exposure within or across virtual spaces. Multiple sinks are useful for monitoring distributed sources. Wonderland ordinarily allows running a single instance of a client per computer. It caches assets (local settings, 3D objects, and avatar models) into a local directory when a client connects to a server for the first time. This cache is used to populate a client scene in subsequent sessions. By allowing a client to have multiple caches, we are able to execute multiple clients acting as clones of a local user (as shown in Figures 4.2 and 5.9). When an avatar is ‘forked,’ the Wonderland server can recognize if multiple avatars represent the same user (host), but there is no way others can recognize such difference. While exploring world music, one might want to simultaneously pay close attention to tracks associated with different countries or genres, maintaining consistency with other participants. An active listener can spawn “doppelgänger” presence and locate representatives at each location of interest, the clones capturing each respective avatar’s soundscapes, individually controlled using narrowcasting functions such as `self-deafen`. Similarly one can participate in a conference and at the same time join a world tour of music.

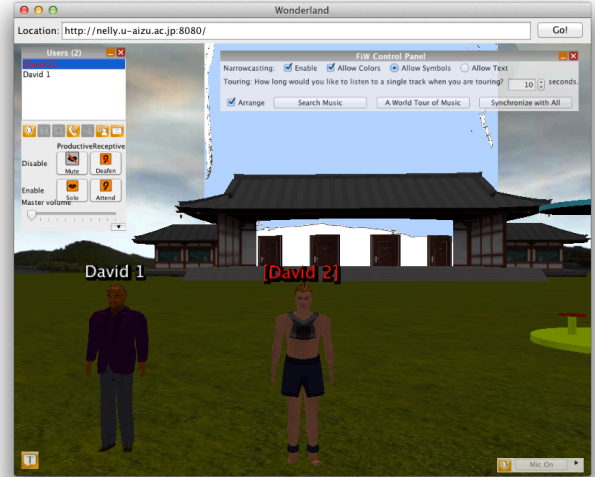
5.2.1 Autoventriloquism

An interesting example of a configuration enabled by multipresence and narrowcasting is “autoventriloquism.” One’s own voice can be heard from another place by forking one’s avatar, deafening one (the source) and muting the other (the sink) to avoid the usual “sidetone” feedback (as shown in Figure 5.10)

5. Narrowcasting and Multipresence in Wonderland



(a) David 1 is self-deafened.



(b) David 2 is self-muted.

Figure 5.10: Autoventroliquism: Generally one hears one’s own voice in both ears, but with this multipresence configuration, one can hear one’s voice coming from the left.

5.2.2 Autofocus

Because of multipresence, multiple selves might be within hearing range of a particular track, separately spatialized for each. Using an “autofocus” technique [Fernando et al. \(2006\)](#) to avoid the paradoxes of such seemingly conflicted soundscapes, the nearest self-identified sink (one of the clones) is automatically, implicitly **attended**, and the others are implicitly **deafened** to that source.

As multipresence can be thought to effectively enable virtual cloning, which is metaphorically a “fork” operation that splits a process in an operating system, so can autofocus be considered analogous to “join” that coalesces processes: multipresence articulates distributed self-identified avatar existence (violating the usual singleton cardinality of existence), and autofocus resolves the ambiguity of non-unique perspective.

5.2.3 Coexistence: Not Only One→Many but Many→One

Not only may a single user have multiple avatars, but also multiple users can share a location as, for instance, when joining a tour. A docent can initiate a tour, sharing a track with others in a session. Users may join the tour, allowing their avatars to be automatically guided through a sequence of track placemarks. An avatar automatically guided in a tour could passively relax while one in parallel conference might actively react. Even though this technique does not support pure implementation of coexistence in which multiple users share a single avatar, it provides satisfying user experience when joining a tour of world music.

5.3 Evaluation and Assessment of FiW

5.3.1 Experimental Design

Using Folkways in Wonderland as a virtual laboratory, we pose the following questions: how do social actors, represented by avatars, interact in such an immersive cyberworld, when presented with a specific collaborative task? How does one effectively locate and listen only to a particular song or a member? A laboratory environment enables us to control variables and thus answer—at least within this restricted environment—questions about such dependencies with rigor that cannot be achieved in the real world.

5. Narrowcasting and Multipresence in Wonderland

5.3.1.1 Procedure

The experiment, which conformed to institutional ethics guidelines for subjective experiments (and required no special dispensation),¹ was conducted in two sessions (a week apart) in three stages: preparatory formalization, individual and group tasks (briefly described in § 6.6), and a music scavenger hunt game. Prior to entering the space, each subject was profiled, recording age, gender, and ethnic background, all considered as variables within the social cluster. During the second and third stages participants were organized into teams of two or three members and asked to perform group activities including narrowcasting.

5.3.1.2 Subjects and Preparation

Participants were computer science undergraduate students, age 20–24: six females and 17 males; 12 Chinese, 10 Japanese, and one American. After 30 minutes of orientation, a feature-spanning exercise was conducted to train the participants. Instantiation (“spawning”) jitters creation locations so everyone doesn’t “beam down” together. Avatars can be initialized with randomly varied appearance, but students were encouraged to personally customize their puppets, which costume and features persist across subsequent sessions (as start-up preferences), while exploring FiW.

5.3.1.3 Musical Scavenger Hunt

Organized into teams, participants were encouraged to work together by interacting in the space. Understandably and predictably, participants tended to team-up

¹<https://www.coloradocollege.edu/other/irb/principles-of-ethical-research.dot>

5. Narrowcasting and Multipresence in Wonderland

according to nationality and native language. A “target track” (hint only, such as “find a track with no instruments”) was announced by the referee, and the goal of the game was to locate this track on the map. Once each target track was located, the referee announced to all participants that it was found, and the winners were asked to explain their strategy before everyone was prompted with the next target track.

5.3.2 Experimental Results

Even though audio narrowcasting features and multipresence have been previously described, they had never been evaluated before. Further, narrowcasting has been applied here not only to musical audition and voice-chat, but also to text-chat. Full narrowcasting is more powerful than conferencing systems with room models for privacy, since narrowcasting explicitly includes sink operations (`deafen` and `attend`) as well as source functions (`mute` and `select [solo]`). In combination with multipresence, the granularity of narrowcasting control is finer than that allowed by systems that use a conference room as the metaphor of isolation. For instance, Dolby Axon requires making a private chatroom to adjust bidirectional privacy. Therefore the experimental protocol was specifically designed to have participants use and evaluate multimodal narrowcasting features, i.e., repeat some tasks enabling and disabling narrowcasting while evaluating the FiW music browser. Salient characteristics of narrowcasting include: when such features are used in a cyberworld, how useful they are, and how easily they can be used. During the training session participants were taught how to use the system, including its narrowcasting commands. Participants were then asked to

5. Narrowcasting and Multipresence in Wonderland

Questionnaire Item	Rating					Quality
	Responses	2	3	4 or 5		
Overall experience	22	1	5	16		73%
Helpfulness of narrowcasting features	20	0	5	15		75%
Helpfulness of multipresence	20	1	4	15		75%
Intuitiveness of visual cues (icons and colors)	21	2	4	15		71%
Ease of finding track compared to navigating around Smithsonian Folkways web site	20	1	7	12		60%
Helpfulness of placemarks and history mechanism	21	0	3	18		86%
	Responses	No	Yes	Agreement		
Usefulness of colors to distinguish narrowcasting operations	23	2	21			91%
Usefulness in sharing a secret	21	4	17			81%
Intuitiveness of using mouth & ear as base for icons	22	1	21			95%

Table 5.2: Results of questionnaire based on 6-point Likert-Type scale **Chomeya (2010)** (0: worst, 1: poor, 2: basic, 3: normal, 4: good, 5: excellent). Two users rated a few of the items as “basic,” but no “worst” or “poor” judgements were recorded. Agreement is expressed as a fraction of total ratings (where negative is [0-2] and positive is [3-5]). Quality is expressed “good” or “excellent” as a fraction of responses.

5. Narrowcasting and Multipresence in Wonderland

compare and contrast songs from different parts of the world, recall the techniques or features they used, and answer a questionnaire regarding their experience. A summary of the results of the post-exercise questionnaire is shown in Table 5.2.

Thirteen out of 18 participants answered the technique used to locate tracks as “Search” facility and two stated “Placemark” feature. Thirteen out of 19 participants rated “Tours” as their favourite feature of FiW. Six liked the diverse music itself, four liked the “Search” function, and others cited other features. All participants who evaluated the usefulness of “select” and “mute” for multiple track audition found it useful. Out of 19 participants, nine cited “chats” as helpful for private communication where narrowcasting is built-in and seven specifically mentioned “narrowcasting.” When participants were asked whether narrowcasting decorating glyphs (++, --,], []) were useful to understand, 17 out of 21 answered positively. The favourite display combination was the combination of colors and symbols (e.g.: “[Carol]”). Four preferred the combination of colors and text (e.g.: “Carol (Select)”). All except one agreed that the figurative icons based on mouth and ear representations in the HUD are intuitive. Due to lack of time, experimental data could not be collected for the musical scavenger hunt.

5.4 Discussion

Social networks such as Facebook, Twitter, and LINE have gained global popularity during recent years. Users exchange text messages and multimedia (graphics, video and audio) or conduct conferences among private or public groups. Even though there are limited privacy settings—such as blocking users (related

5. Narrowcasting and Multipresence in Wonderland

to “mute”), following a certain topic (analogous to “select”), and sharing information only among a set of friends—narrowcasting could provide more articulated control over shared media. For example, if one wants to share an exclusive secret, narrowcasting provides a way to form a coterie without having to make a new group or private chat room with selected confidants. The problem with making a new group or chat room is that generally recipients could accept such group request (in FaceBook or LINE, but not Dolby Axon) and they might send a subsequent group invitations to others, in which case the original user’s goal would be compromised. Using “deafen,” one can simply exclude some members and share a secret with the rest of a group without making a new clique. The same approach can be used against “cyber bullying” [Campbell \(2005\)](#): one may just exclude (mute and/or deafen) a boor without leaving a group of colleagues. The narrowcasting solo function can be effective in situations such as virtual classrooms [Gardner et al. \(2011\)](#); [Ibanez et al. \(2010\)](#); [Peña-Ríos et al. \(2012\)](#), allowing a student to select a teacher or mentor to closely monitor and thereby avoid distractions. Since our experiment concluded that most users have positive feelings about such user interface conventions regarding narrowcasting, we hope to evangelize such functions with social media applications developers.

5.5 Enabling Multimodel Interfaces

5.5.1 Wonderland–CVE Bridge

We extended Wonderland client [Ranaweera et al. \(2009\)](#) to connect our own CVE server. This component, dubbed the “Wonderland–CVE Bridge,” is middleware

5. Narrowcasting and Multipresence in Wonderland

that obtains position of a Wonderland avatar and relays that information to a CVE server using required position parameters x , y , & z location coordinates and $roll$, $pitch$, & yaw orientation coordinates. The position details of an avatar can be used collaboratively in virtual environments, as they determine the localization of sounds, viewport of each avatar, and standpoint on a map. The bridge is also capable of receiving position parameters and updating the location and direction of each Wonderland avatar. Via this component, mobile applications platformed on NTT DoCoMo *iappli* can also display and control Wonderland-hosted avatars.¹ By extracting emotions from voice, then relaying them through CVE server, user emotions can eventually be displayed in virtual world by way of animations [Amarakeerthi et al. \(2010\)](#); [Weerasinghe et al. \(2013, 2010\)](#).

5.5.2 Rotary Motion Platform

The $\text{S}_{\text{chair}}^e$, a swivel chair with a servomotor, can be deployed as a rotary motion platform, a multimodal output device in which haptic display modality is yaw [Duminduwardena and Cohen \(2004\)](#). In groupware situations, such orientation can also be used to twist iconic representations of a seated user (Figure 5.11), avatars in a virtual world [Ramsamy et al. \(2006\)](#). Exploring FiW, a user controls his/her avatars while listening to music and appreciating orientation displayed by the rotary motion platform. Decoupling orientation from location allows rotating avatars without translating them. Such distinction allows, for instance, multiple, self-identified “multipresent” avatars to orient themselves with azimuth or the swivel chair rotary motion platform while maintaining separate locations across

¹www.youtube.com/watch?v=zPzx2OkSsl4

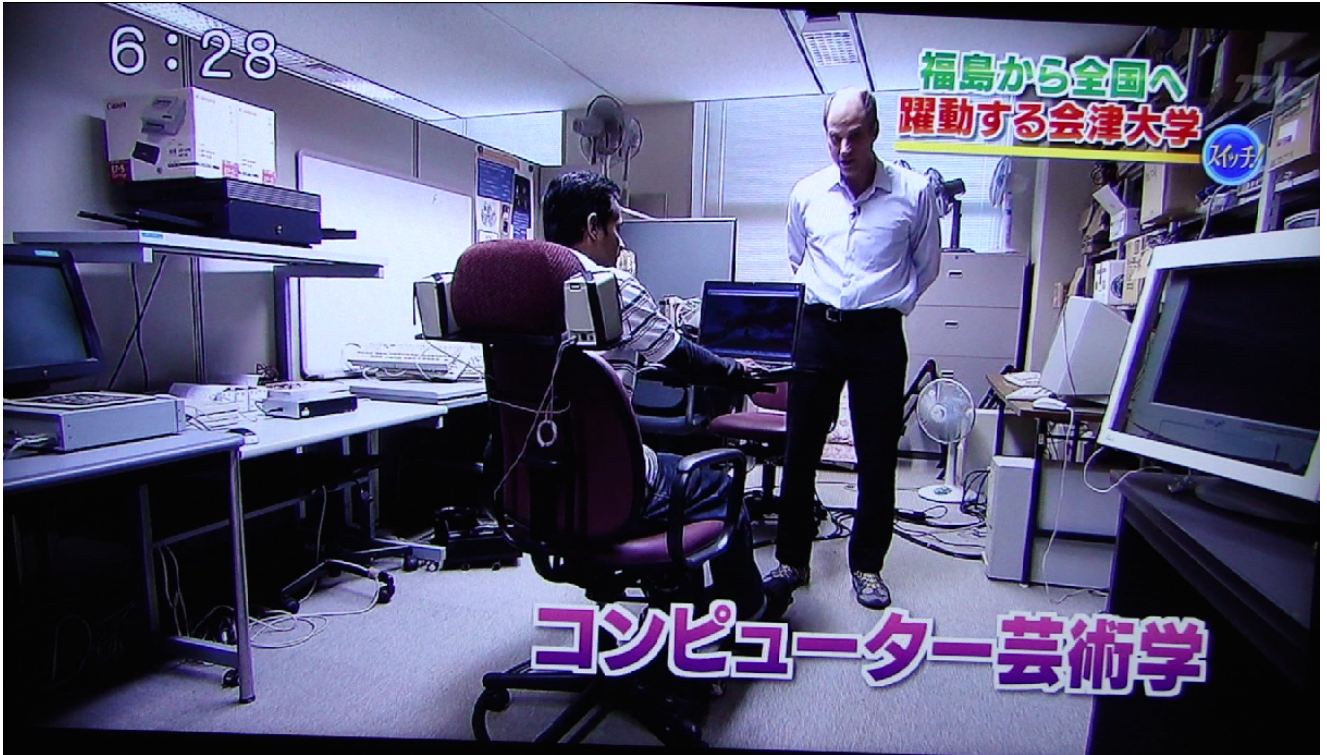


Figure 5.11: Proprioceptive mixed reality experience with $\mathcal{S}_{\text{hare}}$ rotary motion platform. “Nearphones” straddling the headrest display spatial sound with minimal crosstalk.

an auditioned soundscapes (such as the global music library).

5.6 Conclusion & Future Research

We have presented a novel application for listening to world music inside a virtual space. Rather than finding tracks using traditional interfaces, an avatar- or avatars-represented user can explore music immersively while adjusting their soundscape with narrowcasting. Users can invoke `mute` or `select` functions to listen only to particular songs when cacophony might distract. The same functions

5. Narrowcasting and Multipresence in Wonderland



Figure 5.12: World Music in Wonderland (WMIW), a distributed application incorporating audio clips, ancillary artwork, metadata, and virtual landscapes and cityscapes, enabling collaborative information seeking through virtual reality style music browsing. This extensible system defines a virtual audio-visual space comprising a series of maps and a built environment (streets, buildings, rooms), and populated by arbitrary collections of sonic tracks (speech, music, sound), together with metadata. (WMIW is funded by the The Canadian Society for Traditional Music.)

5. Narrowcasting and Multipresence in Wonderland

can be used to exclude distracting members when focusing on particular speaker or writer in voice- or text-chats. To prevent one's voice or words from being delivered to other members, users can use **deafen** or **attend** functions. By cloning, one can be at multiple spaces at the same time. This is the first enterprise-quality, distributed narrowcasting and multipresence system, which flatters and is showcased by the unique Smithsonian Folkways world music collection. Conforming to MX: IEEE 1599, we embrace standard encoding format, encouraging possible future extensions. For instance, the authors are preparing a space that emphasizes shared musical heritage between Cape Breton (Nova Scotia) and Ukraine (Figure 5.12).¹

¹diversitycapebreton.ca/ir/visit-virtual-cape-breton

Part III

Summary

Chapter 6

Conclusions

6.1 Gestural Interface for Music Control

We described how to use a smartphone as a simplified electronic baton in mixed reality concerts. A user-conductor can point at an instrument to select it, tap the touchscreen to start/pause playing, and tilt/steer to adjust volume and panning. The synchronization of gestures with music and animation has been one of the biggest challenges in many systems we have surveyed, although ours had only minimal delays. Unlike contemporary systems, ours does not require user or equipment to be placed at specific locations there is no issue regarding room lighting nor interference with other players or obstacles. We had compared user experience with a contemporary commercial game, receiving acceptable ratings from the participants. (Readers are encouraged to view a short video of the system in action.)¹

We modernized the “*Poi*,” a Māori performance art, opening it up to internet-amplified multimedia. By sensing its magnetometer, the twirling of a mobile phone can be used to sequence score-following music. Synchronizing such se-

¹www.youtube.com/watch?v=J7iX9QKtxOc

quencing with sound spatialization, also modulated by the azimuth of the whirled phone, as through an annular (ring-shaped) speaker array, allows interactive, multimodal interaction.

6.2 Narrowcasting and Multipresence in Cyber-worlds

We have presented a novel application for listening to world music inside a virtual space. Rather than finding tracks using traditional interfaces, an avatar- or avatars-represented user can explore music immersively while adjusting their soundscape with narrowcasting. Users can invoke `mute` or `select` functions to listen only to particular songs when cacophony might distract. The same functions can be used to exclude distracting members when focusing on particular speaker or writer in voice- or text-chats. To prevent one's voice or words from being delivered to other members, users can use `deafen` or `attend` functions. By cloning, one can be at multiple spaces at the same time. This is the first enterprise-quality, distributed narrowcasting and multipresence system, which flatters and is showcased by the unique Smithsonian Folkways world music collection. Conforming to MX: IEEE 1599, we embrace standard encoding format, encouraging possible future extensions. For instance, the authors are preparing a space that emphasizes shared musical heritage between Cape Breton (Nova Scotia) and Ukraine (Figure 5.12).

By implementing multimodal narrowcasting at both server (music audition) and client sides (voice- and text-chat), we show how narrowcasting features can

extend existing systems. FiW is also the first instance of narrowcasting (`select [solo]`, `mute`, `deafen`, `attend`) being applied to text-chat. Various types of narrowcasting state representations— including colors, symbols, and icons— were alternatively presented and evaluated (in § 5.3.1). We also exploited the duality between `mute` and `deafen` (as traced by the stippled line in Figure 5.6’s component diagram). We further discussed interesting examples of a configuration enabled by multipresence and narrowcasting such as “autoventroliquism,” “autofocus,” and “coexistence.”

Part IV

Appendix

Appendix A

Characterizing Subjective Experience: iBaton

Rasika Ranaweera and Michael Cohen

Name: _____

Student ID: _____

6.3 Activities and challenges

- Register, position your self,¹ practice, and play “Harmonix’s *Disney Fantasia: Music Evolved*” game
- Calibrate iBaton and control virtual concert using gestures
- Use iBaton Simulator to control virtual concert using keyboard and mouse
- Please answer the questionnaire (this document)

Please create a new account for the XBox One using a Microsoft account credentials. Sign up for a free account (<https://signup.live.com/signup>) if you do not have one. (You are also free to use an existing account.)

¹<https://support.xbox.com/en-US/xbox-one/accessories/setup-play-space>

1	2	3	4	5
Poor/Too Hard	Basic/Hard	Normal/Moderate	Good/Easy	Excellent/Very Easy

Table 6.1: Scoring Scale (5-point)

6.3.1 Preparatory formalization

Task	Fantasia	iBaton	Status
Calibration	Stand back about 1.4m from Kinect, then raise your hands until the game console recognizes your posture.	Face the screen, point the iPhone at an instrument, then tap “+/-” buttons to align direction with selection.	
Training	Practice then play a selected game (any realm).	Try playing different orchestral music.	
Composing and controlling	Try a “Composition Spell.”	Try controlling your own music.	

Table 6.2: Preparatory formalization

6.3.2 How to play Fantasia

- Follow the instructions on the screen and locate a realm (Hint: You should meet “Yensid” on your way)
- Familiarize with the game play by practicing until you are confident
- Play the selected game and remember your score when completed
- Unlock a “Composition Spell” then move your arms up and down to compose a song.

Tips

- Extend your arms then clap to go to upper level.
- Press \equiv button in the controller to interrupt the game play.

6.3.3 Create a song and export to iBaton

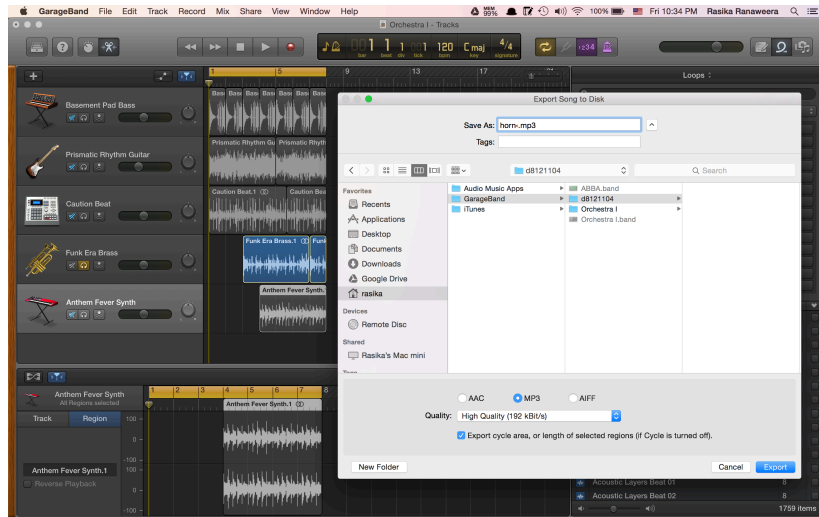


Figure 6.1: Create own songs using GarageBand, export to iBaton, and control your own orchestra!

★ Task: Create about 10 sec. long song with five instruments using GarageBand.

- bass (upright or guitar)
- guitar (acoustic or electric)
- horn (Saxophone, Trumpet, Trombone, etc)
- bebop (Drums)
- keyboard (Piano, Organ)

Upon completion, solo each track, export as MP3 (Share / Export Song to Disk...) with “Export cycle area, or length of selected regions” option checked. When exporting, please make sure to create a new directory with your student ID, then use common instrument name (e.g.: *m516xxx/bass.mp3*). Finally zip the directory and share (via email).

6.3.4 How to play iBaton

iBaton	Task
	<ol style="list-style-type: none"> 1. Connect to a CVE server by entering IP address
	<ol style="list-style-type: none"> 2. Set the mode to “Continuous”
	<ol style="list-style-type: none"> 3. Enable selection then point at instruments
	<ol style="list-style-type: none"> 4. Tap on “+/-” buttons to align direction with selection
	<ol style="list-style-type: none"> 5. Start or stop instruments by pushing buttons
	<ol style="list-style-type: none"> 6. Toggle volume and pan switches then steer or tilt the smartphone to adjust volume and panning individual instruments

Table 6.3: Actions 1, 2, and 4 need to be done once only at the beginning.

iBaton Simulator	Task
	<ol style="list-style-type: none"> 1. Click on the arrow and enter a value (this replicates calibration step)
	<ol style="list-style-type: none"> 2. Move the slider for selection (selected instrument is shown)
	<ol style="list-style-type: none"> 3. Start or stop instruments by pushing buttons
	<ol style="list-style-type: none"> 4. Change volume of a selected instrument
	<ol style="list-style-type: none"> 5. Change panning of a selected instrument

Table 6.4: After calibration step move the sliders using mouse or keyboard.

6.4 Questionnaire

★ Task: Control the virtual orchestras following the instructions in Table 6.3 (both classical and modern) about 2 minutes each.

★ Task: Then use the software simulator following the instructions in Table 6.4 and repeat step above.

★ Task: Please complete the questionnaire in Table 6.11 using Table 6.1 scale.

★ Task: Rate *Disney Fantasia* and *iBaton* (both referred to as ♣) by answering following questions based on the following scale (Table 6.6).

	Question	Fantasia	iBaton
1.	I think that I would like to play ♣ frequently		
2.	I found ♣ unnecessarily complex		
3.	I thought ♣ was easy to use		
4.	I think that I would need the support of a technical person to be able to use ♣		
5.	I found the various functions in ♣ were well integrated		
6.	I thought there was too much inconsistency in ♣		
7.	I would imagine that most people would learn to use ♣ very quickly		
8.	I found ♣ very cumbersome to use		
9.	I felt very confident using ♣		
10.	I needed to learn a lot of things before I could get going with ♣		

Table 6.5: SUS questionnaire to evaluate user experience.

1	2	3	4	5
Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree

Table 6.6: SUS Scoring Scale (5-point)

1.	How do you rate your experience in seventh generation of video game consoles? (Microsoft Xbox 360, Sony Playstation 3, Nintendo Wii)	Likert
2.	Do you have experience playing Fantasia: Music Evolved? 1. Never heard of it 2. Never played it before 3. Played a few times 4. Good at it 5. Expert	
3.	What was your score in “Fantasia: Music Evolved?” Selected Realm:	Likert
4.	How do you rate your overall game playing experience?	Likert
5.	Rate the difficulty for entering (extend arms) or exiting (expand then clap).	Likert
6.	How easy was the control of the muse with your hand?	Likert
7.	How would you rate yourself using smartphones?	Likert
8.	How do you rate your iBaton experience?	Likert
9.	How easy was the iBaton calibration compared to Disney Fantasia (Kinect sensor) calibration?	Likert
10.	How easy was smartphone control compared to keyboard/mouse control?	Likert
11.	How easy was the control orchestra with smartphone compared to control the muse with your hand?	Likert
12.	How entertaining was controlling your own music with iBaton compared to composing music with Disney Fantasia?	Likert

Table 6.7: Questionnaire to evaluate user experience

6.5 Critics and Suggestions:

After completing all the tasks (check them off when tried), please answer the following questions, then write a few sentences about the user experience.

<p>My experience:</p>

Appendix B

**Characterizing Subjective Experience
& Experimental Protocol:
Folkways in Wonderland
with Narrowcasting and Multipresence
Rasika Ranaweera, Michael Cohen, and Michael Frishkopf**

Name: _____

Student ID: _____

6.6 Activities and challenges

- Read “Using Folkways in Wonderland (“FiW”) on OS X”

Connect to <http://163.143.133.141:8080> (aka. <http://nelly.u-aizu.ac.jp:8080>) via Safari web browser and click the Launch button, enter your student ID & full name in romaji in the Wonderland client window, login, and wait a bit until the client is finished initializing components.



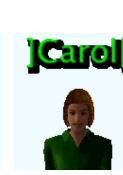





Representation	Exclude		Include	
	Mute	Deafen	Select	Attend
Decorated names floating over avatar heads indicate applied narrowcasting operations. (Combination of distinctive colors and text with glyphs)				
Figurative icons or symbols are used in metadata window and voice chat panel.				
Colors used in voice chat panel.	[Alice]	-Bob-]Carol[+David+

Table 6.8: Different methods for representation of narrowcasting operations

Narrowcasting widgets (controls and displays) are distributed across three panels: check box to enable/disable in control panel head-up display, solo and mute buttons in the metadata pop-up window, and $\{\text{enable, disable}\} \times \{\text{production, reception}\}$ in the head-up display: solo, mute, attend, deafen. The narrowcasting state is also displayed with the respective source or sink using colors and glyphs (written characters, decorating user names floating over avatars)

To activate **multipresence**, open another client window by launching a separate browser window or tab and enter another moniker (name) for yourself.

0	1	2	3	4	5
Worst	Poor	Basic	Normal	Good	Excellent

Table 6.9: Scoring Scale (6-point)

6. Appendix B

1.	How do you rate yourself in experience using OWL or Second Life?	Likert
2.	How do you rate your overall experience?	Likert
3.	Could the narrowcasting features help filter multiple sources and sinks in the cyberworld? (If yes, how do you rate it then?)	Likert
4.	Did the multipresence feature help you to compare and contrast tracks easily? (If yes, how do you rate it then?)	Likert
5.	Did the visual cues (colors and icons) allow intuitive understanding of narrowcasting state?	Likert
6.	Was it easy to locate/find a track and listen to tracks in FiW compared to the Smithsonian web site? (If yes, how do you rate it then?)	Likert
7.	How would you rate the use of placemarks and history mechanism?	Likert

Table 6.11: Questionnaire to evaluate user experience

8.	What type of tools and techniques did you use when locating a particular track?	Open-Ended
9.	Mention three useful features of FiW.	Open-Ended
10.	How do you compare navigating to previously visited track (placemarks or history items in metadata window) in FiW against a web browser's back button or history and bookmarks mechanisms?	Open-Ended
11.	Were the solo and mute functions helpful when listening to multiple tracks at once, compared to multiple tabs or windows in the web browser?	Dichotomous

★ Task: Share a confidential or private communication with a proper subset of your teammates. (You should identify at least two friends in the session then share a secret with only one of them)

12.	What was your strategy? Could narrowcasting features help you complete this task?	Open-Ended
-----	---	------------

★ Task: Disable narrowcasting (uncheck the ‘Enable Narrowcasting’ in the control panel) and repeat above secret-sharing task.

13.	What was your strategy for the private communication?	Open-Ended
14.	Are the colors useful to distinguish narrowcasting operations?	Dichotomous
15.	Are the decorating glyphs (++, -,+, -,], [) useful or difficult to understand?	Dichotomous
16.	Toggle using colors and glyphs for expressing narrowcasting state. What is your favorite formalizing combination? You, You, -You-, -You-, You (Deafen) (Please select one)	Multiple Choice
17.	Are the figurative icons (based on mouth and ear) in the HUD intuitive?	Dichotomous

6.8 Musical Scavenger Hunt Game

6.8.1 Preparation

- Make sure you have a partner and s/he is not sitting next to you.
- The goal of the game is to locate a “target track” track on the map. (please follow the referee’s instructions carefully)

6.8.2 Scoring Mechanism

- The person who locates the track gets a point, which is automatically divided equally among his/her group.
- A wrong answer results in a negative point, distributed in the same way.
- Once a target track has been located, the referee announces to all participants that it’s been found, and provides the next target track.

- At the end of a fixed period of time, the players with the most points wins.
(Winners will be announced later)

6.8.3 Conclusion

Send your console logs to the referee as evidence of the session. This data is also be used to analyze your activities; such as the time spent to find a specific track. Copy the content of the “Java console - Open Wonderland 0.5” and save the content as text format as “m516xxxx.txt.”

To	d8121104@u-aizu.ac.jp
cc	mcohen@u-aizu.ac.jp
Subject	FiW Session: m516xxxx
Attachment	m516xxxx.txt

6.9 Bugs, Issues, and Suggestions:

Appendix C

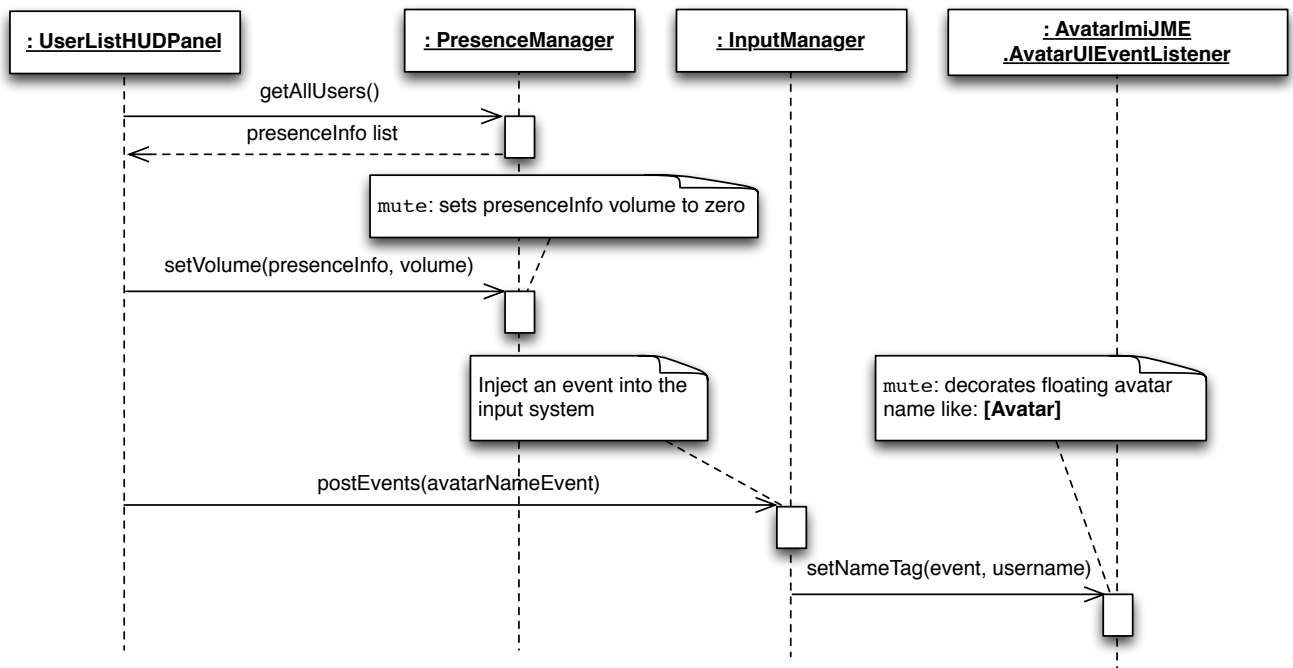


Figure 6.2: Interaction between objects for narrowcast `mute/solo` invocations. Sequence diagrams are based on UML 2 standard ([UML basics: The sequence diagram](#))

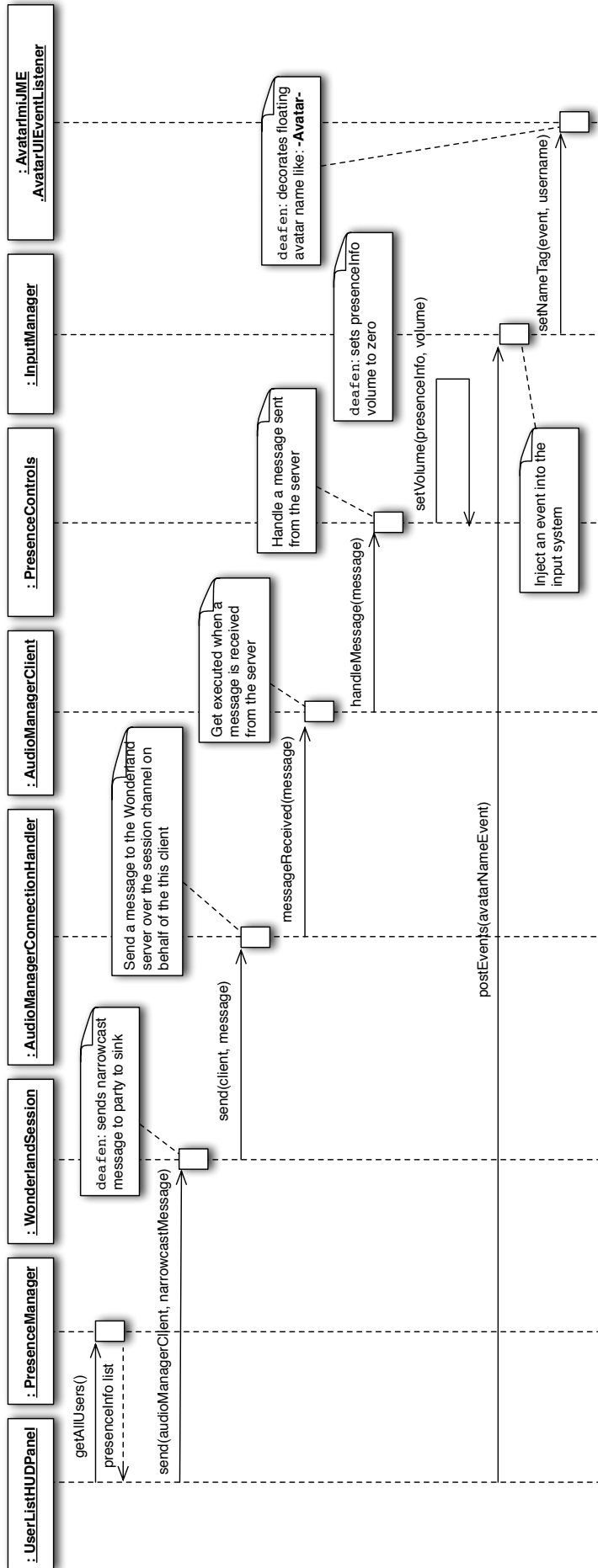


Figure 6.3: Interaction between objects for narrowcast deafen/attend invocations.

6. Appendix B

MyProgram	Inherited from <code>org.alice.apis.moveandturn.Program</code> which is the starting point of the concert program.
MyScene	Inherited from <code>org.alice.apis.moveandturn.Scene</code> which renders the 3D scene (stage, instruments, and the conductor) and connects to the CVE server.
<i>cve.ClientIF</i>	Gets position from the CVE server, clients are expected to implement the methods in this interface.
<i>util.Instrumental</i>	Defines all the audio capabilities an instrument should have.
util.Mode	An enumeration to distinguish STEREO (for stereoscopy) and MONO modes which are supported by this program. When the program is running in stereo mode the camera of the scene moves to right <i>65mm</i> , about the distance between human eyes.
util.Instrument	An enumeration to distinguish instruments in the concert. This includes Keyboard , Trumpet , Bass , Timbale , and Guitar .
MyGuitar	Represents an instrument (in this case a 3D guitar) which has audio capabilities and animations (jiggled, dilated, or contracted depending on the instrument: Keyboard , Trumpet , Bass , and Timbale).
MyJockProm	Represents the 3D conductor in the concert reacts when events received from the server.
util.ConcertLayout	Represents the concert stage where instruments are placed in a semi-circular arrangement.
util.Piece	Creates a piece of the stage. Start and end mark boundaries, and when an angle is given can tell whether the angle is belong to this piece.
util.SoundClip	Loads audio stream of an instrument using a <code>util.AudioClipHandler</code> .
util.AudioController	The implementation of audio capabilities including gaining and panning is done here using <code>line</code> , an element of the digital audio “pipeline,” such as a mixer, an input or output port, or a data path into or out of a mixer.

Table 6.12: Descriptions of classes for iBaton.

FiW Client Package
<u><i>org.jdesktop.wonderland.client.cell.registry.spi.CellFactorySPI</i></u> Responsible for generating the necessary information to generate a new cell.
<u>org.jdesktop.wonderland.client.cell.Cell</u> The client side representation of a cell. A Cell represents some 3D volume on the client and encapsulates some specific functionality. All of the different kinds of dynamic components in Wonderland are often their own custom Cell types.
org.jdesktop.wonderland.modules.fiw.client.FiWCell Representation of the client side FiW cell.
org.jdesktop.wonderland.modules.fiw.client.FiWCellFactory Enables creating new FiW cells in-world via the Cell Palette inside Wonderland.
<u>org.jdesktop.wonderland.client.jme.cellrenderer.BasicRenderer</u> Abstract Renderer class that implements <code>CellRendererJME</code> , an interface for all JME based Cell Renderers.
org.jdesktop.wonderland.modules.fiw.client.FiWRenderer Renderer for Folkways in Wonderland module which creates the texture-mapped world cylinder, track placemarks, a conical gazebo, and entry doors.
<u>org.jdesktop.wonderland.client.cell.ComponentMessageReceiver</u> A Component that provides a cell specific communication channel with the server.
<u>org.jdesktop.wonderland.client.input.EventClassListener</u> A simplified event listener which provides the input system with an array of event classes.
org.jdesktop.wonderland.modules.fiw.client.EntryDoor Allows creating new instances of EntryDoors, which resembles a door to the music browser. This is a 3D model as well as an event listener.
org.jdesktop.wonderland.modules.fiw.client.PlaceMark Represents a clickable dot in FiW browser. When clicked, color of this 3D object is changed and metadata window is also updated.
org.jdesktop.wonderland.modules.fiw.client.AlbumFrame AlbumFrame is the container for album art image. When narrowcasting operations are executed frame color is changed accordingly.
org.jdesktop.wonderland.modules.fiw.client.Particles A flare animation when one clicks on a door.
org.jdesktop.wonderland.modules.fiw.client.ClientMessenger A utility class that fetches the channel component of the cell and sends a message to the Wonderland server.
<u>org.jdesktop.wonderland.modules.fiw.client.ui</u> ui package contains classes such as; <code>Browser</code> , <code>ControlPanelHUD</code> , <code>MapDialog</code> , <code>TrackDialog</code> , <code>TourSelectionDialog</code> which are windows and panels for FiW.

Table 6.13: Descriptions of Wonderland native classes (underlined), FiW classes (embolded), packages (plain), and interfaces (emphasized) separated with required packages for a module in Wonderland.

FiW Common Package	
org.jdesktop.wonderland.modules.fiw.common.FiWTrack	Representation of a musical track which has metadata (author, album, genre, location,) as attributes.
<i>org.jdesktop.wonderland.modules.fiw.common.IFinder</i>	An interface which has methods to find the albums, artists and other resources.
org.jdesktop.wonderland.modules.fiw.common.Finder	Implementation of <code>org.jdesktop.wonderland.modules.fiw.common.IFinder</code> .
org.jdesktop.wonderland.modules.fiw.common.AlbumInfoParser	XML parser implementation for musical database that can be used in both client and server components.
org.jdesktop.wonderland.modules.fiw.common.FiWUtils	A utility class that can be used in both client and server.
org.jdesktop.wonderland.modules.fiw.common.FiWGenericCellMessage	Sub classes of <code>CellMessage</code> are allowed in Wonderland client-server communication channels. <code>FiWGenericCellMessage</code> is immediately sent to the server to start ambient music when an avatar enters the gazebo.
org.jdesktop.wonderland.modules.fiw.common.FiWAudioVolumeMessage	When one executes narrowcasting <code>mute</code> or <code>solo</code> for music, <code>FiWAudioVolumeMessage</code> is sent to the server to adjust track volumes for this client.
org.jdesktop.wonderland.modules.fiw.common.FiWPlayListRequest	Admin can change the position of the FiW model using Wonderland built-in tools like “Object Editor.” After changing position <code>FiWPlayListRequest</code> needs to be sent to the server to restart playing tracks at new positions.
org.jdesktop.wonderland.modules.fiw.common.FiWCurrentTrackMessage	When one is touring <code>FiWCurrentTrackMessage</code> is sent to server after changing avatar position. Then the server can synchronize different avatars to support group tour.
FiW Server Package	
org.jdesktop.wonderland.server.cell.CellMO	Superclass for all server side representation of a cell.
<i>org.jdesktop.wonderland.server.spatial.ViewUpdateListener</i>	This listener is used by cells to get notification when a view which is displaying a cell is updated.
org.jdesktop.wonderland.modules.fiw.server.FiWCellMO	Server side representation of the Folkways in Wonderland module. This allows spatializing audio tracks and also allows individually adjusting track volume.
FiWCellMO.AlbumInfoCellMessageReceiver	Implements the channel component message receiver interface and hides Darkstar implementation details from the developer.

Table 6.14: Descriptions of classes in FiW common and server packages.

FiW Util Package	
org.jdesktop.wonderland.modules.fiw.util.AlbumTraverser	This class provides Directory traverser, which collects files (album art images and track mp3s downloaded by web scraper) under a given directory and its sub directories and generate the list of tracks and albums.
org.jdesktop.wonderland.modules.fiw.util.AlbumResourcesLocator	Using <code>org.jdesktop.wonderland.modules.fiw.common.AlbumInfoParser</code>) this class loads an FiW database XML to memory, traverse through directories, collects files under a given directory and its sub directories, and generates a data structure of hash table as <code>Directory:String, Collection<TrackID:Integer></code> . Directory represents an album id and album art has the same name as directory with image extension. Track ids are tracks with audio extensions. But as the files kept in standard, just follow the standard without iterating directories.
<i>org.jdesktop.wonderland.modules.fiw.util.IAlbumWriter</i>	This class can be used to generate script specific xml file according to the FiW specification.
org.jdesktop.wonderland.modules.fiw.util.AlbumWriter	<code>AlbumWriter</code> can be used to generate script specific XML file according to the FiW specification. Different type of implementations such as DOM, SAX, and VTD.
org.jdesktop.wonderland.modules.fiw.util.CSVReader	Smithsonian Folkways track metadata is curated in Excel format. To convert Excel to XML, first save file as .csv format and use this class to generate XML.
org.jdesktop.wonderland.modules.fiw.util.FreeSoundCSVReader	Another music collection (Freesound: http://freesound.org) is also curated in Excel format. This class can generate XML conforming to FiW.
org.jdesktop.wonderland.modules.fiw.util.FileCloner	<code>FileCloner</code> can be used to duplicate a given file multiple times. Multiples of a single file was used when FiW server was stress tested with different types (<i>mp3</i> and <i>wave</i>) and lengths (30 seconds clips vs. 12 minutes) to confirm audio capabilities in Wonderland (in § 4.1.6).
org.jdesktop.wonderland.modules.fiw.util.ResourceDownloadManager	Represents the offline web scraper (spider) which downloads resources (including album art images and track mp3s) and metadata from the Smithsonian site specified by the track URL.
org.jdesktop.wonderland.modules.fiw.util.experiment.ScavengerHuntAnalyzer	<code>ScavengerHuntAnalyzer</code> class is used to scan the user logs (submitted by the experiment participants) and calculate the time taken for one to find a particular musical track.

Table 6.15: Descriptions of utility classes for FiW module.

Curriculum Vitae

Education

- Ph.D. Candidate in Computer Science and Engineering, University of Aizu, (Dissertation title: “Narrowcasting-Enabled Groupware Music Archive Exploration and Gestural Interfaces for Orchestral Control in Cyberworlds”), Japan (Expected completion date: March 2016)
- M.Sc. Computer Science and Engineering, University of Aizu, Japan, 2010
- B.Sc. Information Technology, University of Moratuwa, Sri Lanka, 2005

Selected Publications

- Rasika Ranaweera, Michael Cohen, and Michael Frishkopf. Narrowcasting and Multipresence for Music Auditioning and Conferencing in Social Cyberworlds. *Presence: Teleoperators and Virtual Environments*, 2015 24:3.
- Rasika Ranaweera, Michael Cohen, and Michael Frishkopf. Narrowcasting Enabled Immersive Music Browser for Folkways World Music Collection. In *CASA: 26th International Conference on Computer Animation and Social Agents*. May 2013, Istanbul, Turkey. <http://www.cs.bilkent.edu.tr/casa2013>.

- Rasika Ranaweera, Michael Cohen, Kensuke Nishimura, Yuya Sasamoto, Yukihiro Nishikawa, Tetunobu Ohashi, Ryo Kanno, Anzu Nakada, and Julián Villegas. Whirled Sequencing of Spatial Music. AES Japan Section, Sendai, Oct. 2012.
- Rasika Ranaweera, Michael Frishkopf, and Michael Cohen. Folkways in Wonderland: A cyberworld laboratory for ethnomusicology. In Int. Conf. on Cyberworlds. Oct. 2011, Canada. <http://cw2011.cpsc.ucalgary.ca>.

Research Experience

Research Assistant (2010–2013)

Supervisor Prof. Michael Cohen

Topic Narrowcasting- and Multipresence-Enabled Music Browser

Description Proof of concept FiW, a novel application for listening to music inside a virtual space. System supports MX: IEEE 1599, narrowcasting and multipresence. A broad survey of related research and a user evaluation is compiled and published.

Affiliation University of Aizu

Research Associate (2009–2010)

Supervisor Dr. Masayuki Hisada

Topic Research and development concerning the web application security with combination of static and dynamic analysis

Description Supported by the Program for Promotion of Private-Sector Key Technology Research of the National Institute of Information and Communication Technology (NICT) of Japan

Affiliation NST Inc. Japan

Teaching & Working Experience

- Teaching Assistant, Human Interfaces & Virtual Reality, University of Aizu. (2010–2015)
- Engineer, Aizulab Inc., Japan. (June 2013–present)
- Software Engineer, Ridgecrest Asia (Pvt) Ltd., Sri Lanka. (March 2008–Mar 2010)

Scholarships and Awards

- 2012: Selected as a research assistant to have preferential treatments by the University of Aizu
- 2010, 2011, & 2012: Japan Student Services Organization (JASSO) scholarship
- 2009: Rotary Yoneyama scholarship
- 2008: Support Association for International Students of the University of Aizu (SAISUA) scholarship

References sorted by name

- Ackerman, M. S., Starr, B., Hindus, D., and Mainwaring, S. D. (1997). Hanging on the 'wire: a field study of an audio-only media space. *ACM Trans. on Computer-Human Interaction*, 4(1):39–66. [66](#)
- Alam, S., Cohen, M., Villegas, J., and Ahmed, A. (2009). Narrowcasting for articulated privacy and attention in SIP audio conferencing. *J. of Mobile Multimedia*, 5(1):12–28. [65](#), [74](#)
- Amarakeerthi, S., Ranaweera, R., and Cohen, M. (2010). Speech-based emotion characterization using postures and gestures in CVEs. In *CW: Proc. Int. Conf. on Cyberworlds*, pages 72–76, Singapore. [92](#)
- Baggi, D. and Haus, G. (2009). IEEE 1599: Music encoding and interaction. *IEEE Computer*, 42(3):84–87. [4](#), [49](#), [62](#), [63](#)
- Baggi, D. L. and Haus, G. M. (2013). *Music Navigation with Symbols and Layers: Toward Content Browsing with IEEE 1599 XML Encoding*. John Wiley & Sons, Inc. [49](#)
- Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual

- SUS scores mean: Adding an adjective rating scale. *J. of Usability Studies*, 4(3):114–123. [26](#)
- Barz, G. F. and Cooley, T. J., editors (2008). *Shadows in the Field: New Perspectives for Fieldwork in Ethnomusicology*. Oxford University Press, 2nd edition. [53](#)
- Behringer, R. (2005). The concept of a visual interface for conducting humans and synthesizers. In *5th Open Wkshp. on MUSICNETWORK*, Vienna. [7](#), [8](#), [11](#), [23](#)
- Bolhassan, A., Cohen, M., and Martens, W. (2004a). VR4U2C: A multiuser multiperspective panorama and turnorama browser using QuickTime VR and java featuring multimonitor and stereographic display. *Virtual Reality Society of Japan*, 9(1):69–78. [39](#)
- Bolhassan, A., Cohen, M., and Martens, W. L. (2004b). VR₄U₂C: A Multiuser Multiperspective Panorama and Turnorama Browser Using QuickTime VR and Java Featuring Multimonitor and Stereographic Display. *Trans. Virtual Reality Society of Japan*, 9(1):69–78. ISSN 1344-011X. [35](#)
- Borchers, J. O., Samminger, W., and Mühlhäuser, M. (2001). Conducting a realistic electronic orchestra. In *UIST: Proc. ACM Symp. on User Interface Software and Technology*, pages 161–162, New York. ACM. [7](#), [8](#), [11](#)
- Bormann, K. (2005). Presence and the utility of audio spatialization. *Presence: Teleoperators and Virtual Environments*, 14(3):278–297. [70](#)
- Boustead, P. and Safaei, F. (2004). Comparison of delivery architectures for

- immersive audio in crowded networked games. In *NOSSDAV: Proc. 14th Int. Wkshp. on Network and Operating Systems Support for Digital Audio and Video*, pages 22–27, New York. ACM. [51](#)
- Boustead, P., Safaei, F., and Dowlatshahi, M. (2005). DICE: Internet delivery of immersive voice communication for crowded virtual spaces. In *Proc. IEEE Conf. 2005 on Virtual Reality, VR '05*, pages 35–41, Washington. IEEE Computer Society. [52](#)
- Bradshaw, D. and Ng, K. (2008). Tracking conductors hand movements using multiple Wiimotes. In *AXMEDIS: Int. Conf. on Automated Solutions for Cross Media Content and Multi-channel Distribution*, pages 93–99. [8](#), [9](#), [11](#)
- Brooke, J. (1996). SUS—a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7. [23](#)
- Campbell, M. A. (2005). Cyber bullying: An old problem in a new guise? *Australian J. of Guidance and Counselling*, 15(1):68–76. [91](#)
- Carlson, J., Han, R., Lao, O., Narayan, C., and Sanghani, S. (2003). Rapid prototyping of mobile input devices using wireless sensor nodes. In *WMCSA: Proc. Wkshp. on Mobile Computing Systems and Applications*, pages 21–29. [8](#), [11](#), [23](#)
- Chen, Y.-X. and Butz, A. (2009). Musicsim: Integrating audio analysis and user feedback in an interactive music browsing UI. In *Proc. Int. Conf. on Intelligent User Interfaces, IUI '09*, pages 429–434, New York. ACM. [49](#)

- Chomeya, R. (2010). Quality of psychology test between Likert scale 5 and 6 points. *J. of Social Sciences*, 6(3):399–403. [89](#)
- Cohen, M. (2000). Exclude and Include for Audio Sources and Sinks: Analogs of mute & solo are deafen & attend. *Presence: Teleoperators & Virtual Environments*, 9(1):84–96. [82](#)
- Cohen, M. and Fernando, O. N. N. (2009). Awareware: Narrowcasting Attributes for Selective Attention, Privacy, and Multipresence. In Markopoulos, P., de Ruyter, B., and Mackay, W., editors, *Awareness Systems: Advances in Theory, Methodology and Design*, Human-Computer Interaction Series, chapter 11, pages 259–289. Springer. [67](#), [83](#)
- Cohen, M. and Györbiró, N. (2009). Mobile Narrowcasting Spatial Sound. In *Proc. IWPASH: Int. Wkshp. on the Principles and Applications of Spatial Hearing*, Zao. [66](#)
- Cohen, M. and Kawaguchi, M. (2003). Narrowcasting operations for mobile phone CVE chatspace avatars. In *Proc. ICAD: Int. Conf. on Auditory Display*, pages 6–9, Boston. [66](#)
- Cohen, M., Ranaweera, R., Ito, H., Endo, S., Holesch, S., and Villegas, J. (2012). Whirled worlds: Pointing and spinning smartphones and tablets to control multimodal augmented reality displays. In *HotMobile: Int. Wkshp. on Mobile Computing Systems and Applications*, San Diego. [4](#), [34](#)
- Cohen, M., Ranaweera, R., Nishimura, K., Sasamoto, Y., Endo, S., Oyama, T., Ohashi, T., Nishikawa, Y., Kanno, R., Nakada, A., Villegas, J., Chen,

- Y. P., Holesch, S., Yamadera, J., Ito, H., Saito, Y., and Sasaki, A. (2013). “Tworls”: Twirled worlds for multimodal ‘padiddle’ spinning & tethered ‘poi’ whirling. In *ACM SIGGRAPH 2013 Posters*, SIGGRAPH ’13, pages 67:1–67:1, New York. ACM. [34](#)
- Cooley, T. J., Meizel, K., and Syed, N. (2008). *Shadows in the Field: New Perspectives for Fieldwork in Ethnomusicology*, chapter Virtual fieldwork: three case studies, pages 90–107. Oxford University Press, 2nd edition. [76](#)
- Crisostomo, S., Safaei, F., and Platt, D. (2004). On the provision of immersive audio communication to massively multi-player online games. In *ISCC: Proc. Ninth Int. Symp. on Computers and Communications*, volume 2, pages 1000–1005, Washington. IEEE Computer Society. [45](#), [57](#)
- Damm, D., Fremerey, C., Kurth, F., Müller, M., and Clausen, M. (2008). Multimodal presentation and browsing of music. In *ICMI: Proc. 10th Int. Conf. on Multimodal Interfaces*, pages 205–208, New York. ACM. [49](#)
- Dann, W., Cooper, S., and Pausch, R. (2006). *Learning to Program with Alice*. Prentice Hall. [12](#)
- Dann, W. P., Cooper, S., and Pausch, R. (2008). *Learning to Program with Alice*. Prentice Hall, second edition. ISBN-10-13-208516-X, ISBN-13-978-0-13-208516-8. [35](#)
- de Freitas, S. (2008). Serious virtual worlds: A scoping study. Technical report, Joint Information Systems Committee, Bristol. [42](#), [43](#), [45](#), [46](#)

- Downie, J. S. (2002). Toward a theory of music information retrieval queries: System design implications. In *3rd Int. Conf. on Music Information Retrieval*, pages 299–300, Paris. [50](#)
- Duminduwardena, U. C. and Cohen, M. (2004). Control System for the $\mathcal{S}_{\text{ha}_i\text{r}^e}$ Internet Chair. In *CIT: Proc. Fourth Int. Conf. on Computer Information Technology*, pages 215–220, Wuhan. [92](#)
- El Rhalibi, A., Merabti, M., Carter, C., Dennett, C., Cooper, S., Ariff Sabri, M., and Fergus, P. (2009). 3D Java web-based games development and deployment. In *ICMCS: Int. Conf. on Multimedia Computing and Systems*, pages 553–559. [59](#)
- Fadden, S., Ververs, P. M., and Wickens, C. D. (1998). Costs and benefits of head-up display use: A meta-analytic approach. In *Proc. Human Factors and Ergonomics Society Annual Meeting*, volume 42, pages 16–20. [54](#)
- Fernando, O. N. N., Adachi, K., Duminduwardena, U., Kawaguchi, M., and Cohen, M. (2006). Audio Narrowcasting and Privacy for Multipresent Avatars on Workstations and Mobile Phones. *IEICE Trans. on Information and Systems*, E89-D(1):73–87. [4](#), [65](#), [82](#), [85](#)
- Fernando, O. N. N., Adachi, K., Duminduwardena, U. C., Kawaguchi, M., and Cohen, M. (2003). Audio Narrowcasting for Multipresent Avatars on Workstations and Mobile Phones. In *ICAT: Proc. Int. Conf. on Artificial Reality and Telexistence*, pages 106–113, Tokyo. [66](#)
- Fine, G. A. (2001). *International Encyclopedia of the Social and Behavioral Sci-*

ences, chapter Participant Observation, pages 11073–11078. Elsevier Science.

[53](#)

Fitzsimons, D. (2006). Rhumb lines and map wars: A social history of the mercator projection by mark monmonier. *The Professional Geographer*, 58(4):497–499. [61](#)

Flanagan, D. (2004). *Java examples in a nutshell*. O'Reilly. [15](#)

Frank, J., Lidy, T., Peiszer, E., Genswaidner, R., and Rauber, A. (2008). Ambient music experience in real and virtual worlds using audio similarity. In *Proc. Int. Wkshp. on Semantic Ambient Media Experiences*, SAME '08, pages 9–16, New York. ACM. [51](#)

Funkhouser, T., Min, P., and Carlbom, I. (1999). Real-time acoustic modeling for distributed virtual environments. In *SIGGRAPH: Proc. 26th Annual Conf. on Computer Graphics and Interactive Techniques*, pages 365–374, New York. ACM Press/Addison-Wesley Publishing Co. [50](#)

Gardner, M., Gnem-Gutierrez, A., Scott, J., Horan, B., and Callaghan, V. (2011). Immersive education spaces using Open Wonderland from pedagogy through to practice. In *Multi-User Virtual Environments for the Classroom: Practical Approaches to Teaching in Virtual Worlds*, pages 190–205. Hershey, PA: Information Science Reference. [45](#), [57](#), [91](#)

GChen, Y.-F., Rebolledo-Mendez, G., Liarokapis, F., de Freitas, S., and Parker, E. (2008). The use of virtual world platforms for supporting an emergency response training exercise. In *Proc. 13th Int. Conf. on Computer Games: AI*,

- Animation, Mobile, Interactive Multimedia, Educational & Serious Games*, pages 47–55, Wolverhampton. [43](#)
- Geerts, D. (2006). Comparing voice chat and text chat in a communication tool for interactive television. In *NordiCHI: Proc. Nordic Conf. on Human-Computer Interaction: Changing Roles*, pages 461–464, New York. ACM. [77](#)
- Gilkey, R. H. and Anderson, T. R., editors (1997). *Binaural and Spatial Hearing in Real and Virtual Environments*. Lawrence Erlbaum Associates, Mahway. ISBN 0-8058-1654-2. [70](#)
- Greenberg, J., Nepkie, J., and Pence, H. E. (2009). The SUNY oneonta second life music project. *J. of Educational Technology Systems*, 37(3):251–258. [52](#)
- Greenhalgh, C. and Benford, S. (1995). MASSIVE: a distributed virtual reality system incorporating spatial trading. In *Proc. 15th Int. Conf. on Distributed Computing Systems*, pages 27–34, Vancouver. [66](#), [74](#), [76](#)
- Guzdial, M. and Rose, K. (2002). *Squeak: open personal computing and multimedia*. An Alan R. Apt book. Prentice Hall. [12](#)
- Hughes, B. and Kamat, A. (2005). A metadata search engine for digital language archives. *D-Lib Magazine*, 11(2). [49](#)
- Ibanez, M. B., García, J. J., Galán, S., Maroto, D., Morillo, D., and Kloos, C. D. (2010). Multi-user 3D virtual environment for spanish learning: A Wonderland experience. In *ICALT: Proc. Int. Conf. on Advanced Learning Technologies*, pages 455–457, Washington. IEEE Computer Society. [91](#)

- Je, H., Kim, J., and Kim, D. (2007). Hand gesture recognition to understand musical conducting action. In *RO-MAN: 16th Int. Symp. on Robot and Human Interactive Communication*, pages 163–168. [8](#), [11](#)
- Kanno, T. and Cohen, M. (2002). An architecture for collaborative virtual environments. *3D Forum: J. of Three Dimensional Images*, 16(1):166–174. ISSN 1342-2189. [16](#)
- Kaplan, J. and Yankelovich, N. (2011). Open Wonderland: An extensible virtual world architecture. *IEEE Internet Computing*, 15(5):38–45. [45](#), [57](#)
- Kong, L. (2001). Religion and technology: Refiguring place, space, identity and community. *Area*, 33(4):404–413. [53](#)
- Kreidler, J. (2013). *Programming Electronic Music in Pure Data*. Wolke Verlagsges, 2nd edition. [36](#)
- Kuhn, M., Wattenhofer, R., and Welten, S. (2010). Social audio features for advanced music retrieval interfaces. In *Proc. Int. Conf. on Multimedia*, pages 411–420, New York. ACM. [50](#)
- Lam, T. C., Ding, J. J., and Liu, J.-C. (2008). XML document parsing: Operational and performance characteristics. *IEEE Computer*, 41:30–37. [62](#)
- Lee, S. and Kim, G. J. (2008). Effects of visual cues and sustained attention on spatial presence in virtual environments based on spatial and object distinction. *Interacting with Computers*, 20(4-5):491–502. [79](#)
- Lee, W.-M. (2012). *Beginning IOS 5 Application Development*. John Wiley & Sons. [15](#), [20](#)

- Ludovico, L. A. (2008). Key concepts of the IEEE 1599 standard. In *Proc. Conf. on the Use of Symbols to Represent Music and Multimedia Objects*, pages 15–26, Lugano. IEEE CS. [64](#)
- Ludovico, L. A. (2009). IEEE 1599: a Multi-layer Approach to Music Description. *J. of Multimedia*, 4(1):9–14. [62](#)
- Marcus, A. (2003). Icons, symbols, and signs: Visible languages to facilitate communication. *Interactions*, 10(3):37–43. [82](#)
- Mayrath, M., Traphagan, T., Jarmon, L., and Resta, P. (2010). Teaching with virtual worlds: Factors to consider for instructional use of Second Life. *J. of Educational Computing Research*, 43(4):403–444. [44](#)
- Messinger, P. R., Stroulia, E., Lyons, K., Bone, M., Niu, R. H., Smirnov, K., and Perelgut, S. (2009). Virtual worlds - past, present, and future: New directions in social computing. *Decision Support Systems*, 47(3):204–228. [45](#)
- Naef, M., Staadt, O., and Gross, M. (2002). Spatialized audio rendering for immersive virtual environments. In *VRST: Proc. ACM Symp. on Virtual Reality Software and Technology*, pages 65–72, New York. ACM. [50](#)
- Nakra, T. M. (1999). *Inside the Conductor’s Jacket: Analysis, interpretation and musical synthesis of expressive gesture*. PhD thesis, Massachusetts Institute of Technology. [8](#), [9](#)
- Nettl, B. (2005). *The Study of Ethnomusicology: Thirty-one issues and concepts*. University of Illinois Press. [52](#)

- Olsen, V. S. (2011). *Alice 3 Cookbook*. Packt Publishing Ltd., Birmingham. ISBN-10 1-849514-92-5. 35
- Pausch, R., Burnette, T., Capehart, A., Conway, M., Cosgrove, D., DeLine, R., Durbin, J., Cossweiler, R., Koga, S., and White, J. (1995). Alice: Rapid prototyping for virtual reality. *IEEE Computer Graphics and Applications*, 15:8–11. 12
- Peña-Ríos, A., Callaghan, V., Gardner, M., and Alhaddad, M. J. (2012). Towards the next generation of learning environments: An InterReality learning portal and model. In *Intelligent Environments*, pages 267–274. 91
- Pereira, J. (2011). The AVALON project and Second Life – the analysis and selection of a virtual world for language learning and teaching. https://www.academia.edu/6346889/The_AVALON_Project_and_Second_Life_The_analysis_and_selection_of_a_virtual_world_for_language_learning_and_teaching, [Online; accessed 15-November-2015]. 46
- Pierce, J. S. (2003). Introducing Alice to a Squeak Wonderland. In *Proc. on Creating, Connecting and Collaborating through Computing*, pages 40–43, Kyoto. IEEE Computer Society. 12
- Pulkki, V. (1997). Virtual source positioning using vector base amplitude panning. *J. of the Audio Engineering Society*, 45(6):456–466. 36
- Ramsamy, P., Haffegge, A., Jamieson, R., and Alexandrov, V. (2006). Using haptics to improve immersion in virtual environments. In *Computational Science – ICCS 2006*, pages 603–609. 92

- Ranaweera, R., Cohen, M., and Endo, S. (2011a). Smartphone as conductor's baton for virtual concerts. In *ACMP: Proc. Int. Conf. on Asia Computer Music Project*, Tokyo. [34](#)
- Ranaweera, R., Cohen, M., and Endo, S. (2012a). iBaton: Conducting virtual concerts using smartphones. In Klyuev, V. and Vazhenin, A., editors, *HCCE: Proc. Int. Conf. on Human-Centered Computer Environments*, Aizu-Wakamatsu. [3](#), [11](#), [34](#)
- Ranaweera, R., Cohen, M., and Frishkopf, M. (2013). Narrowcasting enabled immersive music browser for Folkways world music collection. In *CASA: Proc. Int. Conf. on Computer Animation and Social Agents*, Istanbul. [4](#)
- Ranaweera, R., Cohen, M., Nishimura, K., Sasamoto, Y., Nishikawa, Y., Ohashi, T., Kanno, R., Nakada, A., and Villegas, J. (2012b). Whirled sequencing of spatial music. In *Proc. Audio Eng. Soc. Japan Sect.*, Sendai. [4](#), [38](#), [39](#), [40](#)
- Ranaweera, R., Frishkopf, M., and Cohen, M. (2011b). Folkways in Wonderland: A cyberworld laboratory for ethnomusicology. In *Cw: Proc. Int. Conf. on Cyberworlds*, pages 106–112, Banff. [4](#)
- Ranaweera, R., Nagel, N., and Cohen, M. (2009). Wonderland–CVE bridge. In *Proc. HC-2009: 12th Int. Conf. on Humans and Computers*, pages 91–98, Hamamatsu. [4](#), [91](#)
- Rowlands, D. D. and James, D. A. (2010). Enhancing smart device applications using motion and location. In *CreateWorld: Working on the Edge: Creativity, Technology, and Innovation*, pages 36–46, Brisbane. [16](#)

- Rumsey, F. (1994). *MIDI Systems and Control*. Butterworth-Heinemann, MA, 2nd edition. 15
- Russo, E. (2008). The IEEE 1599 standard for music synthesis systems. In *Proc. Conf. on the Use of Symbols to Represent Music and Multimedia Objects*, pages 49–53, Lugano. IEEE CS. 63
- Sanchez, M. (2014). Fantasia: Music Evolved Review- IGN. <http://www.ign.com/articles/2014/10/21/fantasia-music-evolved-review>, [Online; accessed 15-November-2015]. 8
- Schertenleib, S., Gutierrez, M., Vexo, F., and Thalmann, D. (2004). Conducting a virtual orchestra. *Multimedia*, IEEE, 11(3):40–49. 7, 8, 11
- Schooler, E. M., Casner, S. L., and Postel, J. (1991). Multimedia conferencing: has it come of age? In *Hawaii Int. Conf. on System Sciences*, volume 3, pages 707–716. 65
- Segen, J., Gluckman, J., and Kumar, S. (2000). Visual interface for conducting virtual orchestra. In *15th Int. Conf. on Pattern Recognition*, volume 1, pages 276–279. 8, 11, 12
- Seidman, R., Funk, P., and Jim Isaak, L. L. (2006). *Fluency with Alice: Workbook for Fluency with Information Technology: Skills, Concepts, and Capabilities*. Prentice Hall. 12
- Seo, B., Htoon, M. M., Zimmermann, R., and Wang, C.-D. (2010). Spatializer: a web-based positional audio toolkit. In *Advances in Computer Entertainment Technology*, pages 57–61. 71

- Shaomei, L., Yinghong, K., Haiyan, L., and Xiaoya, A. (2010). Image-based mapping using google earth images. In *ICACC: Proc. 2nd Int. Conf. on Advanced Computer Control*, pages 282–285. 61
- Snider, M. (2014). ‘songpop’ chimes in atop Facebook game chart. <http://www.usatoday.com/story/tech/gaming/2012/12/05/song-pop-top-facebook-game/1746781/>, [Online; accessed 15-November-2015]. 51
- Stewart, R., Levy, M., and Sandler, M. (2008). 3D interactive environment for music collection navigation. In *Proc. DAFX: 11th Int. Conf. on Digital Audio Effects*, Espoo. 56, 60
- Taylor, J. (1997). The emerging geographies of virtual worlds. *Geographical Review*, 87(2):172–192. 53
- Taylor, T. L. (1999). Life in virtual worlds: plural existence, multimodalities, and other online research challenges. *American Behavioral Scientist*, 43(3):436–439. 83
- Thompson, C. W. (2011). Next-generation virtual worlds: Architecture, status, and directions. *IEEE Internet Computing*, 15(1):60–65. 45
- Tsui, C. K., Law, C. H., and Fu, H. (2014). One-man orchestra: Conducting smartphone orchestra. In *SIGGRAPH Asia 2014 Emerging Technologies*, SIGGRAPH ’14, pages 11:1–11:2, New York. ACM. 8
- Usa, S. (1996). Automatic performance control apparatus. US Patent 5,585,584.

- Waldo, J. (2008). Scaling in games and virtual worlds. *Communications ACM*, 51(8):38–44. [45](#), [57](#)
- Weerasinghe, P., Marasinghe, A., Ranaweera, R., Amarakeerthi, S., and Cohen, M. (2013). Emotion expression for affective social communication. In *ICBAKE: Proc. Int. Conf. on Biometrics and Kansei Engineering*, pages 148–153, Tokyo. [92](#)
- Weerasinghe, P., Ranaweera, R., Amarakeerthi, S., and Cohen, M. (2010). “Emo Sim”: Expressing voice-based emotions in mobile interfaces. In *HC-2010: Proc. 13th Int. Conf. on Humans and Computers*, Aizu Wakamatsu. [92](#)
- Yankelovich, N., Kaplan, J., Provino, J., Wessler, M., and DiMicco, J. M. (2006). Improving audio conferencing: are two ears better than one? In *CSCW: Proc. Conf. on Computer Supported Cooperative Work*, pages 333–342, New York. [66](#)
- Yankelovich, N., McGinn, J., Wessler, M., Kaplan, J., Provino, J., and Fox, H. (2005). Private communications in public meetings. In *CHI: Extended Abstracts on Human Factors in Computing Systems*, pages 1873–1876, New York. [66](#), [77](#)
- Yankelovich, N., Walker, W., Roberts, P., Wessler, M., Kaplan, J., and Provino, J. (2004). Meeting central: making distributed meetings more effective. In *CSCW: Proc. Conf. on Computer Supported Cooperative Work*, pages 419–428, New York. [66](#)
- Yokomatsu, Y. (2007). Primassa: Polyphonic spatial audio system with matrix

mixer and speaker array integrated with CVE. Master's thesis, University of Aizu. [40](#), [41](#)

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *MultiMedia*, 19(2):4–10. [24](#)