NORTHWESTERN UNIVERSITY

Using Computational Models to Create Perceptually Relevant User Interfaces for Nonvisual Artifacts

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Technology and Social Behavior

By

Noah Liebman

EVANSTON, ILLINOIS

March 2021

ABSTRACT

Using Computational Models to Create Perceptually Relevant User Interfaces for Nonvisual Artifacts

Noah Liebman

Our experience of the physical world is mediated by our senses, but while most people have five senses, interactions with computer systems are largely limited to the visual sense. When working with nonvisual artifacts, like sound, on computers, such artifacts are typically transformed, or re-encoded, into something visual. Determining how to generate visual representations of nonvisual artifacts to support people's goals is a key design challenge. In many cases, simple encodings work well, but in other cases they fail to account for the complexities of how people's nonvisual senses work. This dissertation uses computational models of human perception to introduce a new class of inter-medium encodings that can better capture information that is relevant for users. I demonstrate this in the context of mixing multitrack audio. First, I show that current visual representations of multitrack audio lack semantic relevance that can be helpful in the mixing process. Next, I use a computational model of auditory perception to infer the perceived loudness of each track of audio, and apply it in the design of MaskerAid, a system that visualizes

perceived loudness and frequency masking. I then describe and report on an empirical evaluation in which I found that MaskerAid meaningfully improved user performance toward the goal of producing mixes in which each track was clearly audible. I conclude with a discussion of the practical and theoretical implications of this work.

Acknowledgements

Despite its sole authorship, I could not have completed this dissertation without the help of many people. First, thank you to my advisor, Darren Gergle, who not only provided insights and feedback, talked through much of the content, and answered my endless questions, but also stuck with me when my research direction went off the rails in year four of a Ph.D. And to my committee, Bryan Pardo, Anne Marie Piper, and Mike Horn for their many insights, encouragement, and valuable feedback.

Thank you to the many faculty I've worked with over the years, both at Northwestern and at Michigan. I specifically want to acknowledge Liz Gerber for her feedback on an earlier version of Chapter 2, and Mick McQuaid for introducing me to the user-centered design process and supporting my classmates and me on the Cuebert project.

The students and staff of Darren Gergle's CollabLab and Bryan Pardo's Interactive Audio Lab provided much intellectual, logistical, social, and moral support. Thanks to the students of these labs who were sounding boards and pilot testers: Prem Seetharaman, Ethan Manilow, Mark Cartwright, and Scott Cambo. And a huge thank you to Marcia Hilliard, manager of the CollabLab, who not only provided invaluable logistical support, but was always happy to lend a friendly, caring, supportive ear.

As user research in a very specific domain, I want to thank my many research participants, many of whom are busy professionals who generously gave of their time. Thanks to everyone at Columbia College who supported my ability to run a study there, and especially to Benj Kanters, who — in addition to jumping through administrative hoops to make this happen — spent hours talking about, giving feedback on, and showing genuine interest in this work.

A big thank you to Chad Reid, Laura Kroll, and everyone at Shure for the support, opportunities, and flexibility they gave me while having faith that I would finish this dissertation eventually. Thanks also to Shure for allowing me to recruit participants from among its employees, and allowing me to use their facility to run the study in Chapter 4. And to Ilana Emmett, my project manager who kept things moving along, helped keep me and this dissertation organized, and pushed me toward the finish line after having stalled out for a while.

And finally, thank you to my family. To my sister, for not mocking me too much when she graduated first, and to my parents, Judy and Marty Liebman, for instilling in me a love of music, audio, and learning, and supporting me in infinite ways along the way.

Table of Contents

List of Figures

CHAPTER 1

Introduction

In a quiet room, a phone vibrates instead of beeps.

The water in a tea kettle reaches a boil, and the kettle begins to whistle.

A blind person uses their cane to navigate city sidewalks.

In all of these cases, information is being translated, or re-encoded, from one sense to another. From the sound of a phone beeping to touch; from the feeling of heat and the sight of steam to sound; from sight to the feeling of the cane on the ground and the sound of each tap and its reverberations. Such re-encodings are useful in the world at large, and are especially useful in the context of human–computer interaction.

People experience the world around them through their senses. While most people have five senses, people spend a substantial amount of time with computers, which are almost entirely visual. Of course people listen to audio through their devices, and research has tried to make tactile (e.g., [**46**]) and auditory (e.g., [**29**]) interfaces, but reality is that computers remain predominantly visual, while the world is multisensory[1]. As a result, most computer-based work requires visual representations of any artifacts being acted upon.

When working with naturally visual objects like text documents or images, generating visual representations in a way that make sense to people is relatively straightforward. However, there are many artifacts and phenomena which are not naturally visible, such

[1] Sadly, Smell-O-Vision never lived up to its potential.

as electricity, temperature, and sound, which people still wish to view, simulate, and manipulate computationally. One of the key design challenges in such cases is generating visual representations of nonvisual artifacts in a way that makes sense to people.

HCI researchers have developed principles to help guide designers toward interfaces that are usable and relevant for users. A core design principle that emphasizes the extent to which the visual is privileged in interfaces is direct manipulation [**54**]. Direct manipulation tells us that interfaces are more straightforward and understandable when they allow users to act on representations themselves rather than on disjointed parameters of the object being manipulated. A key attribute of direct manipulation is that it applies to visual representations. Shneiderman tells us, "Direct manipulation systems offer the satisfying experience of operating on *visible* objects," (emphasis added). In other words, an object being manipulated must be made visual in some way, even when the underlying object being manipulated is not visual.

As Shneiderman realized, user interfaces that enable the manipulation of nonvisual objects must represent them in some way. This is because in order to manipulate nonvisual objects, they must undergo what Parkes calls a "manipulable inter-medium encoding" [**49**], representing one medium (such as audio) in another (a visual interface) to enable manipulation.

Manipulable inter-medium encodings often rely on directly translating underlying physical properties of the artifact in question to a visual form. In the case of sound, for example, this means graphs of amplitude and frequency, typically called a waveform. However, human understanding of a stimulus does not always correspond to such physical properties in straightforward ways: how hot or cold the air feels depends not only on temperature, but on wind and humidity. Similarly, the loudness of a sound depends not only on its amplitude, but also on its spectral content, its duration, and the makeup of other sounds in the environment.

When a representation does not align with a person's goals, either because of perception or interpretation, there is what Hutchins, Hollan, and Normal call a *gulf of evaluation* [**34**]. If a sound is represented by a waveform, but people are trying to manipulate perceived loudness, there is a gulf between user goals and the representation.

Innovative designer and former electrical engineer Bret Victor calls tools that help make the invisible visible "media for thinking the unthinkable" [**62**]. In referring to sound and light whose wavelengths are outside human perception, he says,

> "These sounds that we can't hear and this light that we can't see, how do we even know about these things in the first place? Well, we build tools. We build tools that adapt these things that are outside of our senses to our human bodies, to our human senses."

I believe that many tools that visually encode nonvisual artifacts do an inadequate job adapting them to our human senses. *In this thesis, I address this inadequacy by applying computational models of human perception to manipulable inter-medium encodings in order to create representations that are more perceptually meaningful to people.* I designed and implemented MaskerAid, a system that uses models of human auditory perception to create novel visualizations for use when mixing multitrack audio.

Mixing multitrack audio is an interesting domain in which to study manipulable inter-medium encodings for two reasons. First, the ways multiple tracks of audio and their many adjustable parameters interact to affect the overall sound is quite complex, such that

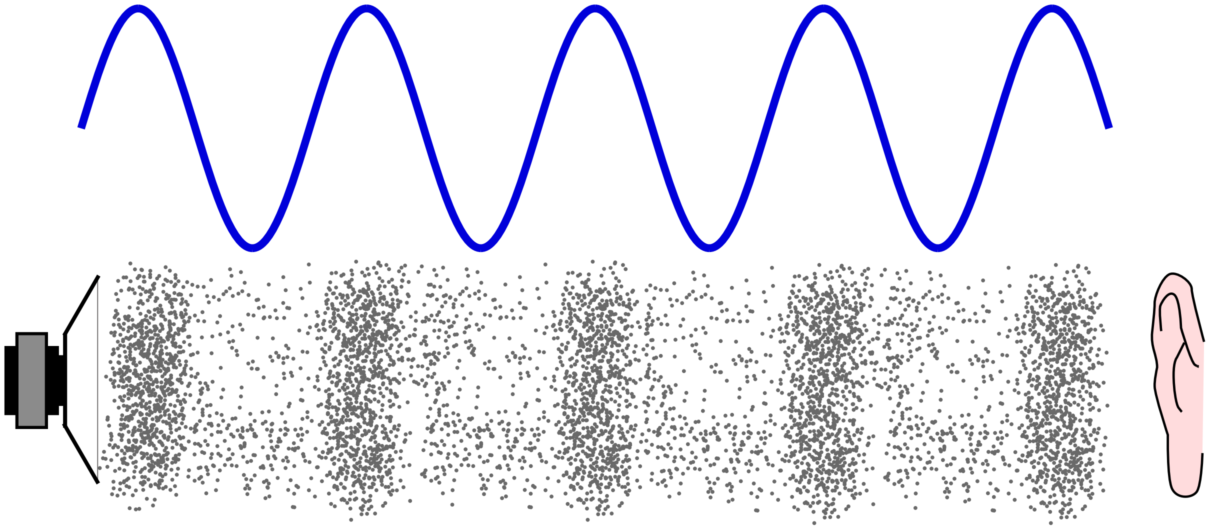


Figure 1.1. A sound wave, shown as a typical plot of sound pressure level (SPL) over time (top), and as compressions and rare factions of air molecules (bottom). CC0 Wikimedia Commons user Pluke.

the underlying physical properties of the audio can deviate significantly from a listener's perceived experience. Second, audio is transient — it does not persist over time — yet mixing interfaces show not just instantaneous information, but information over time. This offers opportunities to study both static and dynamic visualizations.

Before explaining multitrack mixing, it is important to understand some of the fundamentals about sound. When a bell is rung, the bell resonates, vibrating the air around it. These vibrations, called sound waves (Figure 1.1), then propagate through the air. When they reach our ears, we interpret these vibrations as sound. Sound waves have two primary physical characteristics. The *amplitude* of a wave measures the maximum pressure of the displaced air. This is related to how loud something sounds [**44**]. The *frequency* of a wave measures how quickly it oscillates between maximum (compression) and minimum (rare faction) pressure. It is related to the pitch and timbre of a sound [**43**].

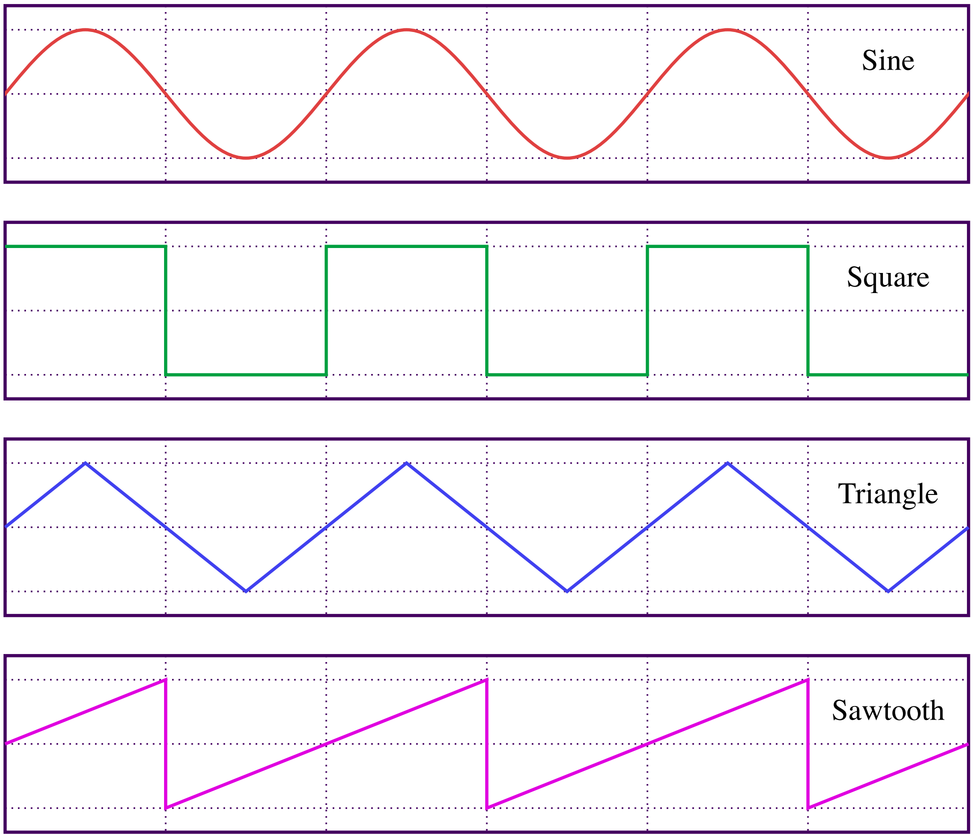


Figure 1.2. Examples of waveforms with the same fundamental frequency but with different harmonics, and therefore different timbres. CC BY-SA Wikimedia Commons user Omegatron.

The simplest sound is a sine wave at a single frequency, though such pure tones rarely occur in nature. The frequency of such a sound is perceived as pitch. Natural sounds are composed of multiple frequencies. When these other frequencies are integer multiples (*harmonics*) of the primary frequency (the *fundamental*), humans continue to perceive the sound as having the same pitch, but with a different timbre (Figure 1.2). If the frequencies in a signal are more randomly distributed, it is perceived as noise [**44**].

These two physical characteristics, amplitude and frequency content, determine how loud a wave sounds to a human listener. At a given frequency, greater amplitude does mean louder, but different frequencies at the same amplitude don't necessarily sound equally loud. This phenomenon is captured in equal loudness curves [**58**], sometimes called Fletcher–Munson curves (Figure 2.3) after their initial discoverers. These curves describe the sound pressure necessary at each frequency to maintain constant loudness.

So far, the discussion has been limited to single sounds, but we usually hear multiple sounds at the same time. When this happens, one sound can affect the perceived loudness of another in a process called *masking* [**44**]. There are multiple kinds of masking, but this dissertation deals exclusively with frequency masking. Frequency masking is when the frequency content in certain bands in one signal overwhelm similar frequencies in another, making that other signal more difficult to hear. In practice, this means that, depending on the amplitude and frequency content of two signals, the presence of one may make another sound quieter than it would be in isolation.

When recording music, each audio source (e.g., guitar, bass, vocals) is typically recorded separately. These recordings, called tracks, are then combined together in a process called mixing. A major objective of mixing is to ensure that each track is heard in an artistically desirable way. The process involves adjusting multiple parameters of each track, including volume, effects, and other processing. The number of tracks and parameters to be adjusted make it a complex process: volume, left–right pan, multiband equalization, reverberation, dynamic processing, and effects on each of 60 tracks makes for a lot of variables. This complexity is compounded by sound changing over time and each variable being adjustable over the course of a song. Finally, because of how human auditory perception works, any change made to one track can affect the audibility of other tracks, possibly at another point in time. The large number of time-varying parameters, the nonlinear way audio signals combine perceptually, and the fact that audio is transient all make multitrack mixing a particularly challenging yet illuminating case for studying and creating computational model–backed manipulable inter-medium encodings.

Developments in computational modeling of the human auditory system, and computing that is now capable of running such models in real time or faster than real time, offer an opportunity to create perceptually relevant visualizations of audio. Using such models, computers are beginning to be able to detect many of the acoustic phenomena audio engineers listen for as they work. These models can tell whether a song's vocal track can be heard over the music, or whether a bass guitar and kick drum are likely to combine in a way that muddies up a mix [**40**]. In this thesis, I will be applying such models to the creation of novel visualizations and user interfaces.

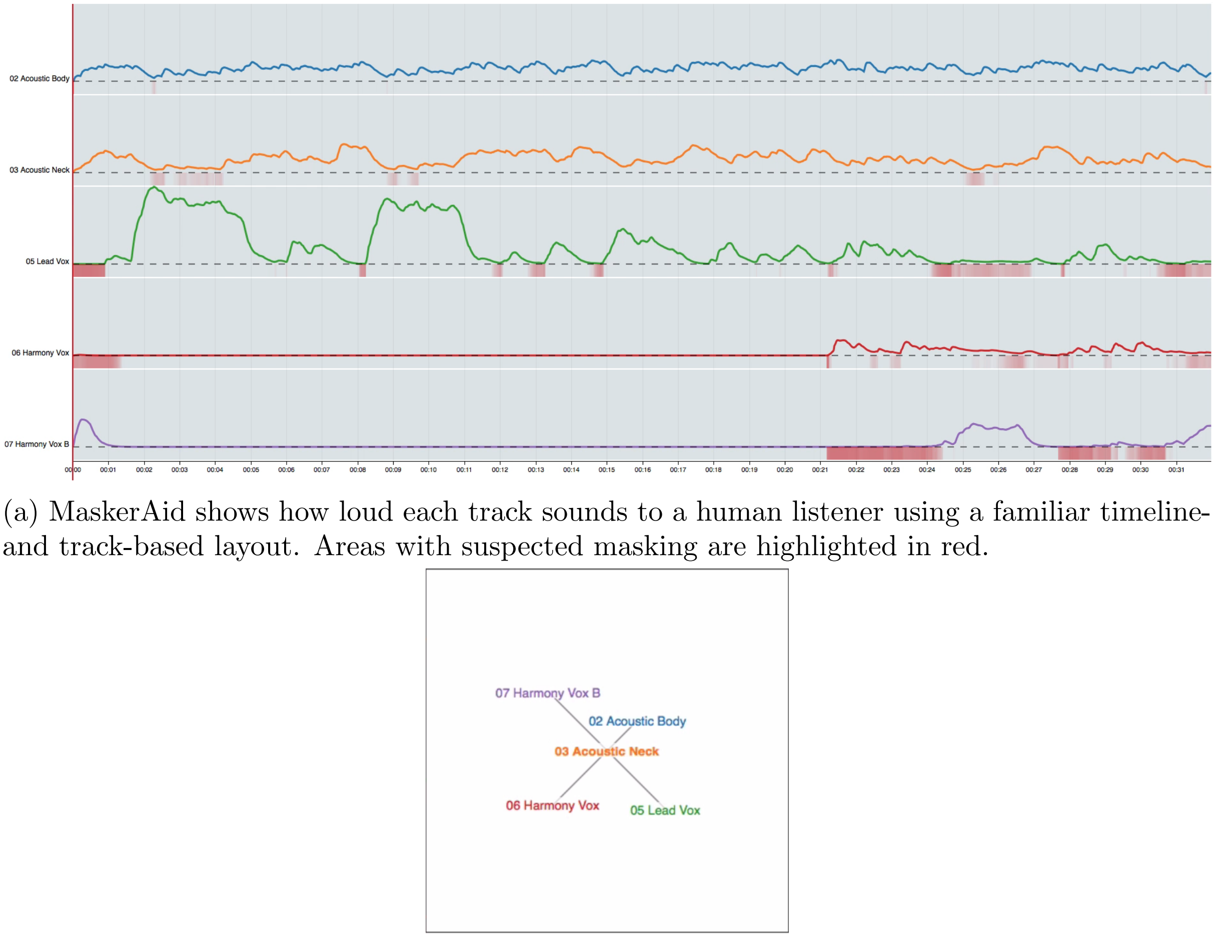
To summarize, in this thesis I address the problem of making nonvisual artifacts manipulable in visual user interfaces — manipulable inter-medium encodings — in ways that more naturally correspond to people's perceptual experience. I do this in the context of tools for professional audio engineers engaged in the task of mixing multitrack audio. Mixing is a task that is vital to the production of high-quality music, yet is particularly challenging for audio engineers and interface designers because of the complex perceptual relationships between audio tracks and the transient nature of audio itself. Mixing also pushes the limits of traditional visual interfaces because the artifact of interest — audio — is nonvisual.

This research was carried out in three phases, corresponding to the user-centered design process [**13**]: user research, design, and evaluation. The proximate goal was to design and test MaskerAid, a user-motivated interactive prototype of an audio visualization system. This is in service of the ultimate goal, which is to demonstrate the value of computational models of human perception to the creation of useful visual representations of nonvisual artifacts.

Phase 1 is a qualitative field study to assess user needs. The goal of this study was to understand the tools, workflows, and processes of professional audio engineers as they complete their work. As I will show in Section 2.4, there are a number of way that current visual representations of audio fail to support audio engineers in their work. This is largely because of the differences between what those representations show and what information is directly useful to mixing engineers.

Phase 2 describes the design and development of MaskerAid. This includes several desgin proposals motivated by the user research described in Phase 1. These designs are visualizations that are largely driven by perceptual models in order to better align what is displayed with what audio engineers need to hear. The final design (Figure 1.3) shows the perceived loudness of each track of audio as it would sound within the context of all other tracks, highlighting where a track may be difficult to hear because of frequency masking. It also interactively shows which tracks are likely to be mixing a selected track in a given time range.

Phase 3 is an evaluation of MaskerAid to assess the extent to which it enabled audio engineers to mix more effectively. It is a laboratory experiment that looks at whether the mixes produced when using MaskerAid have less frequency masking, and are therefore more transparent, and were created with less effort than those that were made with a standard baseline user interface alone. The results are mixed, with MaskerAid seeming to



(a) MaskerAid shows how loud each track sounds to a human listener using a familiar timeline- and track-based layout. Areas with suspected masking are highlighted in red.

(b) MaskerAid shows how far apart each track is to a selected track using spatial distance

Figure 1.3. An overview of MaskerAid's two views

reduce the amount of masking in resultant mixes, but taking more time to complete. The longer time participants spent with MaskerAid may be accounted for by novelty effects.

Chapters 2 through 4 map to these three phases. I then conclude with a short discussion of the contributions of this work. This includes contributions to the fields of audio engineering and HCI more broadly, including accessibility. Through this thesis, my aim is to help the field of HCI develop a deeper understanding of the roles computational models of human perception can play in intelligently translating nonvisual artifacts into manipulable visual objects.

CHAPTER 2

User Research

2.1. Introduction

When music is recorded, each instrument (e.g., guitar, bass, and vocals) is typically recorded independently. Mixing is the process of combining these individual recordings, known as tracks, into a single recording that can be played on the stereo speakers of a home theater, car stereo, or headphones. The primary function is to ensure that each track is heard in an artistically appropriate way by the listener [**36**]. This is accomplished by adjusting the volume (known as the level) of each track, and by applying various types of processing, effects, or other manipulations to tracks or groups of tracks.

Mixing is a complex and time-consuming process that involves many potentially changing variables. A typical pop song may have 60 or more separate tracks of audio. In addition to the level, each track often has many other parameters to be adjusted; for example, left-right pan, multiband equalization, reverberation, and dynamic compression. Many of these effects can have a dozen or more parameters. There are thousands of possible parameters to be adjusted, and they may be tweaked and manipulated hundreds of times over the course of a three-minute song. To further complicate matters, because of how human auditory perception works, changes to one track can affect the audibility of other tracks.

This presents a significant design challenge: representing and manipulating thousands of parameters that affect aural and temporal information, a challenge also recognized by Mycroft et al. [**47**]. Designing interfaces for interacting with audio is challenging because interfaces are visual, while the information they must represent is auditory. Visual information has the advantage of being able to persist over time. Audio, on the other hand, has one non-temporal dimension (amplitude) and is transient over time, and the relationship between amplitude and perceived loudness can be non-obvious.

Before the development of digital audio, mixing was done using an analog mixing board (alternately known as a “mixing console” or “desk”, Figure 2.1) and external (“outboard”) effects hardware. Over time, the design and layout of the mixing board has remained largely consistent. Until digital, the layout of a mixing board was necessarily dictated by the paths of audio signals through the circuitry. Digital made such a direct mapping unnecessary, but the legacy design, and the usability inherent in the physical constraints of analog, meant the old design persisted.

Today, the variety of tools available to mixing engineers has exploded. Digital audio software has become the heart of the studio, and is often used in conjunction with analog mixing consoles, digital hardware control surfaces, and other analog outboard effects. With such a variety of representations and interfaces, understanding what works for audio engineers, what hinders them, and why, is important for driving forward the state of the art.

The work presented in this chapter is a qualitative user research study designed to identify challenges audio engineers face. As user research, it represents Phase 1 of the user-centered design process. I conducted semi-structured interviews with, and observations

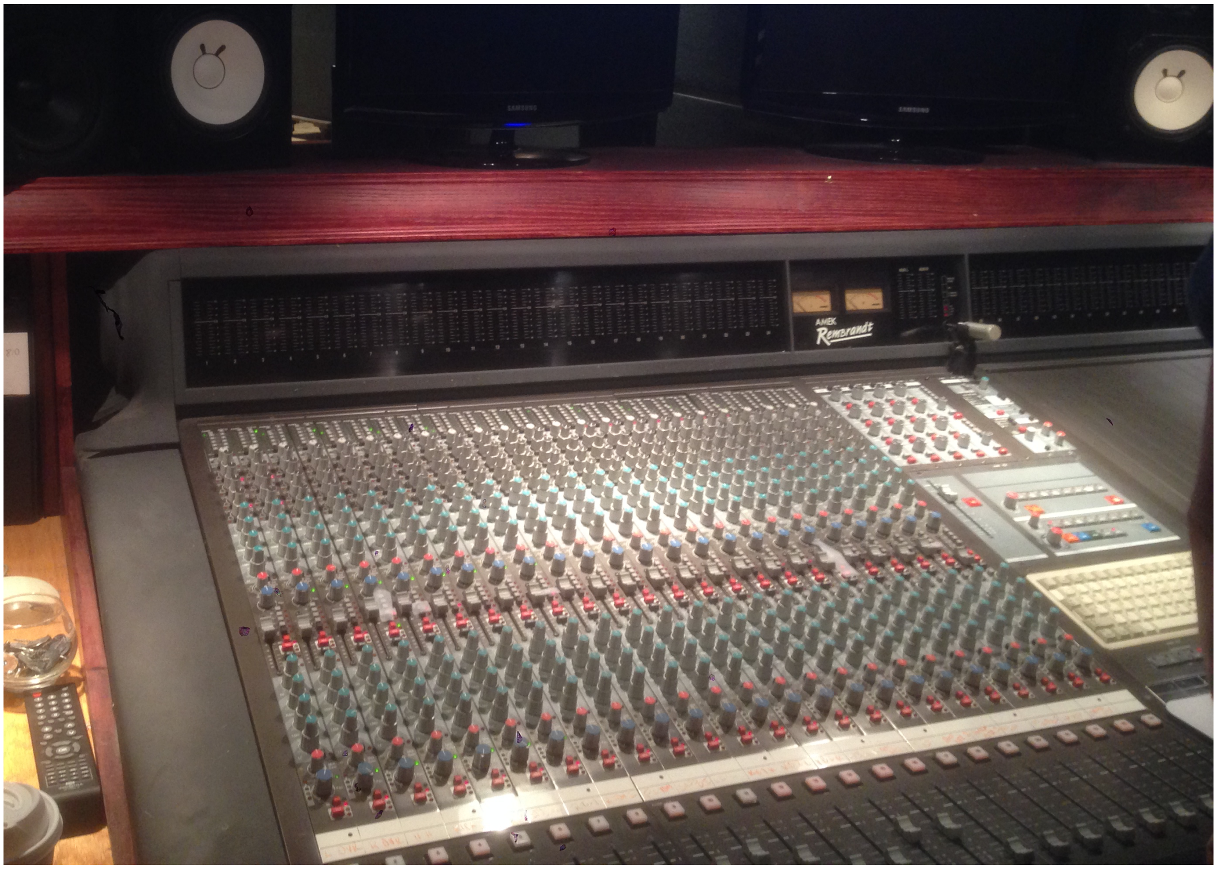


Figure 2.1. The analog mixing board at P1 and P2's studio

of, 10 studio mixing engineers. These interview and observation sessions aimed to identify shortcomings in current mixing tools and practices in order to reveal design opportunities for improved tools. The analysis revealed that there is often a misalignment between task and tool: when mixing, many representations show physical properties of audio or its processing, but what matters to the mixing process is how a track sounds to a listener. This shortcoming, which I discuss in more detail later, is used to motivate the designs described in the next chapter. In addition to this finding, I identified other design challenges having to do with the relative prominence of controls like faders and panning knobs; pre-fader visualizations; and balancing the tension between listening intently and looking at user interfaces and visualizations.

2.2. Background

This section provides background on three topics. First, I introduce direct manipulation, the theoretical framework through which the qualitative data is considered. Next, I describe how sound is generally represented in user interfaces. And finally, I provide an overview of the tools and practices used in mixing in order to introduce the vocabulary necessary to understand the data.

2.2.1. Direct Manipulation

HCI researchers have developed principles to help guide designers toward interfaces that are usable and relevant for users. One such design principle is *direct manipulation* [**54**]. Direct manipulation tells us that interfaces are more straightforward, understandable, and even delightful, when they allow users to act directly on representations of the object of interest rather than on disjointed parameters of the object being manipulated.

In an effort to identify why direct manipulation has cognitive benefits, Hutchins et al. describe direct manipulation as bridging two gulfs between a user's goals and an interface. The *gulf of execution* is the distance between how someone conceives of a task and how they perform it; the *gulf of evaluation* is the distance between how a person interprets an interface's output and the actual output of the system [**34**]. I am focusing primarily on the gulf of evaluation; that is, the distance between representations of audio in mixing tools and the audio itself.

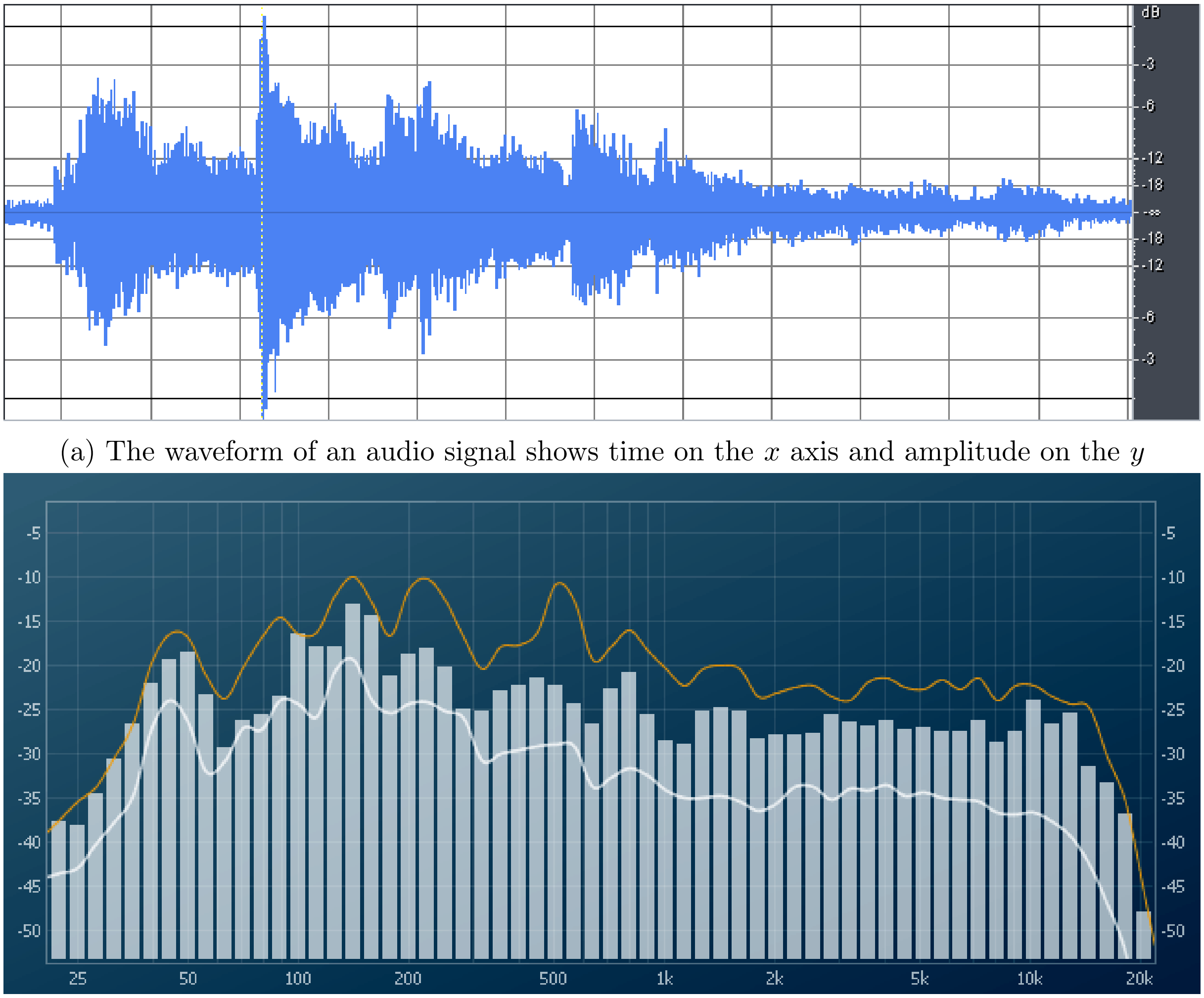
Each of these gulfs is made up of two kinds of distances. *Semantic distance* is the gap between the direct meaning of a representation and meaning that is relevant for a user's goals and intentions. For example, a user may be trying to examine a vehicle's acceleration, but is only given a graph of velocity over time. It is possible to infer acceleration, but it is semantically distant from what is presented. A much less semantically distant representation would show a graph of the value of interest: acceleration. *Articulatory distance* is the gap between the meanings of expressions and their physical form. A time-series graph and a numeric data table may have the same meaning, but the physical form of the data table has greater distance from its perceived meaning.

In analyzing the qualitative data from the study, I found that direct manipulation is a useful theoretical perspective for framing the results. I use semantic distance and articulatory distance to critique various representations of audio from the data.

2.2.2. Representing Sound

User interfaces that enable the manipulation of audio must represent that audio in some way. As Shneiderman tells us, "Direct manipulation systems offer the satisfying experience of operating on *visible* objects," (emphasis added). In other words, direct manipulation only makes sense applied to visual representations, even when the underlying object being manipulated is auditory. In order to manipulate audio, it typically undergoes what Parkes calls a "manipulable inter-medium encoding" [**49**], representing one medium (audio) in another (a visual interface) to enable manipulation[1].

[1] Nonvisual interfaces for interacting with audio are possible, including purely tactile interfaces, but such designs failed to make the leap from analog hardware to modern, in-the-box systems.



(a) The waveform of an audio signal shows time on the x axis and amplitude on the y

(b) The frequency spectrum of an audio signal shows frequency on the x axis and amplitude on the y

Figure 2.2. Visualizations of some of the most fundamental properties of an audio signal

While there are many potential ways to represent audio, two fundamental physical properties of sound make for useful starting points: sound pressure amplitude (broadly perceived as volume/loudness) and frequency (perceived as pitch and timbre). These two properties can be visually represented by simple plots. Plotting amplitude over time is a waveform; plotting amplitude over frequency is a frequency spectrum plot (Figure 2.2).

These representations are used throughout the professional audio world, yet there is not necessarily a direct relationship between these physical properties and how something sounds. For example, it has long been understood that higher frequencies are perceived as louder than lower frequencies of the same amplitude [**26**] (Figure 2.3). Additionally, how loud something sounds to a listener depends not just on the amplitude and frequency content of the signal in question, but the amplitude and frequency content of other signals playing at the same time. Just as you have to speak more loudly on a crowded train than in a quiet room, the presence of one sound can change how loud another seems to be. This is a phenomenon called *frequency masking*[2] [**44**]. When the goal is understanding how loud something will sound, these quirks of auditory perception introduce distance between the meaning of either of the two representations of Figure 2.2 (amplitude or frequency), and the goal of the interpretation (loudness). In other words, there is semantic distance in the gulf of evaluation between these representations and perceived audio.

While I have described these representations of audio in the abstract, they are, in fact, in the tools used by audio engineers every day. The next section introduces how these tools are used in practice.

2.2.3. Current Tools and Practices in Mixing

One cannot understand the work of multitrack mix engineers without some familiarity with tools currently in use in studios. Design and innovation of tools for audio production come from both equipment makers and academic research. This section focuses primarily

[2] I used the terms “frequency masking” and “masking” interchangeably although there are other forms of auditory masking, most notably temporal masking.

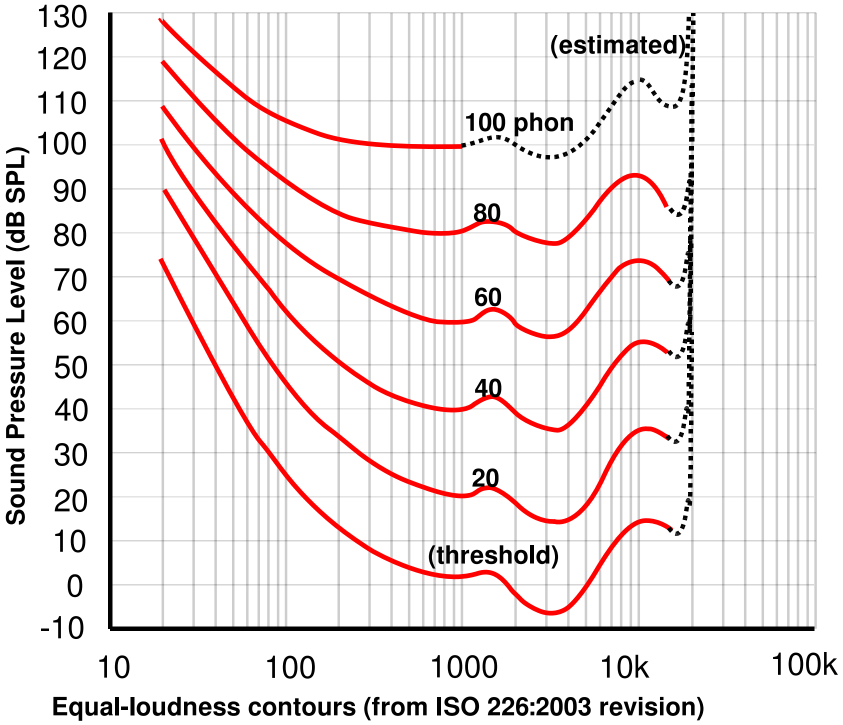


Figure 2.3. Equal loudness contours, often called Fletcher–Munson curves, show how much sound pressure is needed to maintain a given loudness across the frequency spectrum. Each curve shows that more sound pressure is generally needed for lower frequencies to be perceived as loudly as higher frequencies. (Reproduced from Wikipedia user Lindosland.)

on industry. While I will not be detailing how to mix here, [**36**] is a good reference on the subject.

Music production and mixing have gone through multiple revolutions: live miking to multitrack, live mixing to automation, analog to digital, console to computer-based [**3**]. Each time, the design and layout of the mixing board has remained largely consistent. Until digital, the layout of a mixing board was necessarily dictated by the paths of audio signals through the circuitry [**60**]. This meant that there was a one-to-one mapping of controls to functions; each knob performed exactly one function. As a result, the current state of the system was always visible.

Until sophisticated software gave engineers the option of mixing entirely on the computer (which they call “in the box”), the *mixing board* (alternately known as a “mixing console” or “desk”, Figure 2.1) and external (“outboard”) effects hardware is how mixing was traditionally accomplished. Despite its often imposing size, a mixing board is relatively straightforward. It is laid out in columns, each column (“channel strip”) being for a single track of audio. An audio signal comes in at the top of a channel strip, goes through some amount of optional filtering and routing (controlled by knobs), is panned left or right with another knob, and finally comes to the fader at the bottom, which controls how loud that track is in the overall mix. The output of each track’s channel strip is then summed and sent to a master stereo output.

Digital audio made such a one-to-one mapping between controls and functions — and the usability inherent therein — unnecessary, yet the designs of digital tools for mixing multitrack audio continue to be largely inspired by their decades-old analog counterparts [**59**], leaving significant room for innovation. Today, the variety of tools available to mixing engineers has exploded. Digital audio software has become the heart of the studio, and is often used in conjunction with analog mixing consoles, digital hardware control surfaces, and other analog and digital outboard effects. However, such software is typically designed to mimic the behavior, and even the look, of analog equipment (Figure 2.4) even though digital audio offers many new opportunities for interface and visualization design.

The advent of *Digital Audio Workstation* (DAW) software marked a shift in the way audio was manipulated. It largely combined editing and mixing into a single tool, and

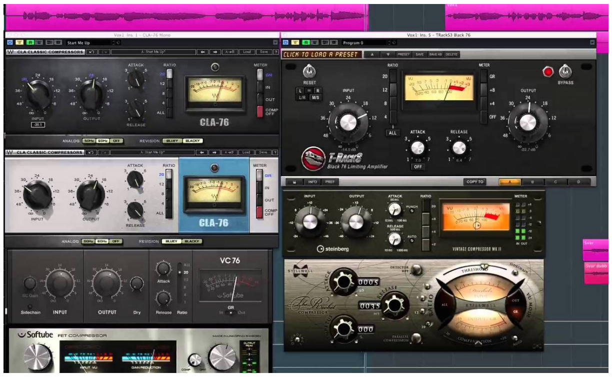
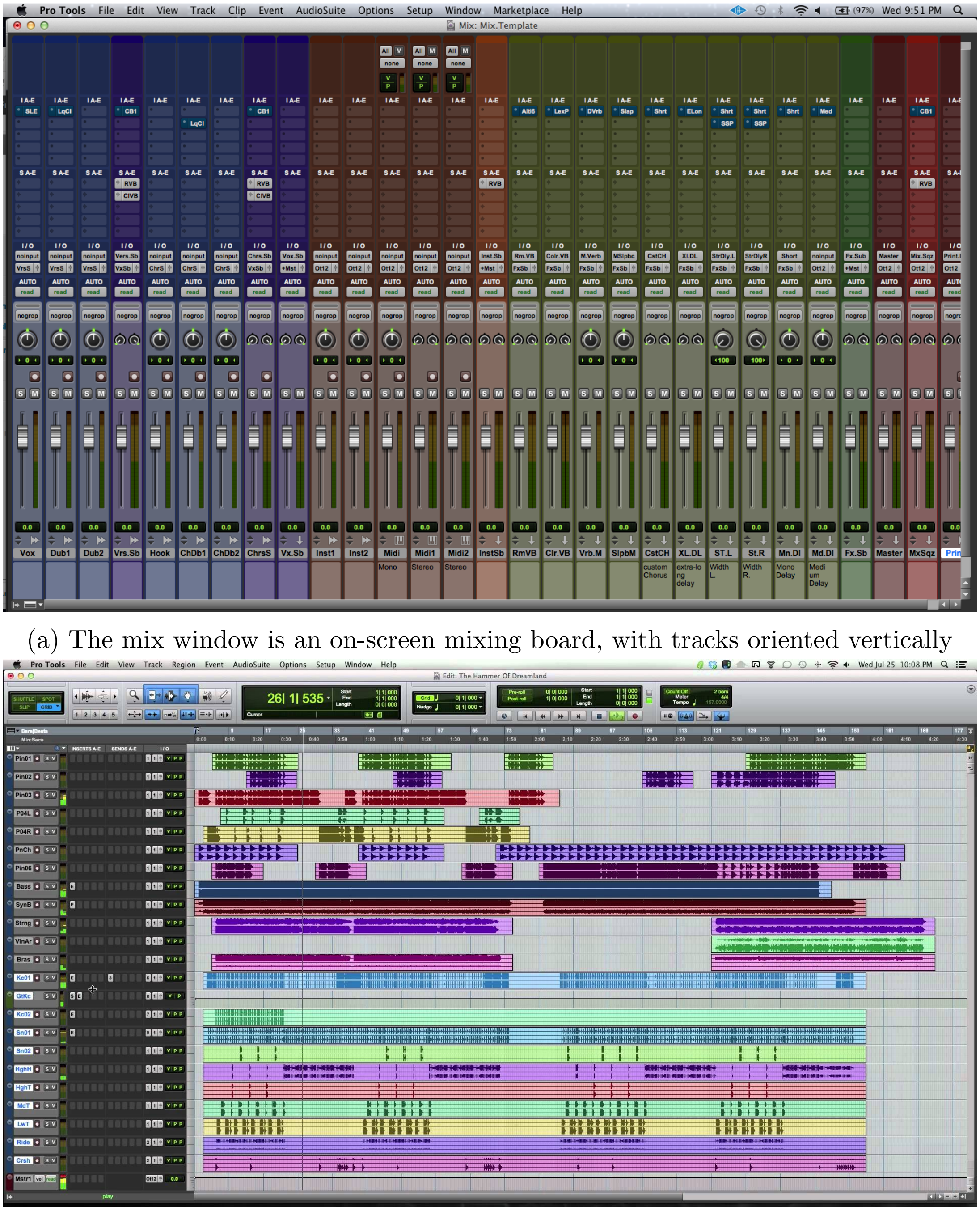


Figure 2.4. Many plug-ins use skeuomorphic designs that mimic analog hardware

made precise manipulations easy and replicable. Even with optional hardware control surfaces that bring the affordances of physical faders, the mixing process is very different with a DAW because it is possible to adjust the levels and effect parameters of each track to precise values at precise times. It also introduced the idea of working with on-screen representations of an audio signal's waveform or with an equalizer's frequency response curve. This is a powerful way to work, but, as I will discuss shortly, it breaks down in ways predicted by Hutchins et al., providing an illuminating case for exploring the use of perceptual models to improve representation and reduce the gulf of evaluation.

The most common DAW among engineers I studied was Avid Pro Tools (Figure 2.5). DAWs are used for both editing and mixing, so they typically have two views: mix, and edit. The mix window is an on-screen representation of a mixing board, while the edit window is a time-based waveform representation of each track [**23**]. As I will show, both representations are used during mixing.

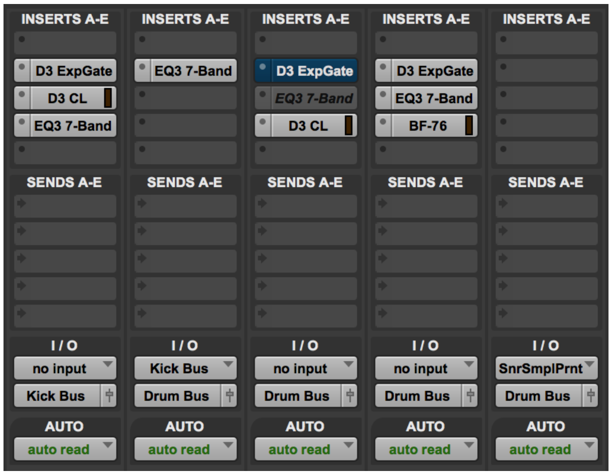
Mixing involves manipulating tracks in a few major ways: *level*, *equalization* (EQ), *dynamic processing*, and *panning*. Level is simply the overall volume of a track. Bear in mind that level is quite distinct from perceived loudness. Just like the volume control


Pro Tools File Edit View Track Clip Event AudioSuite Options Setup Window Marketplace Help
Wed 9:51 PM
Mix: Mix.Template
Vox Dub1 Dub2 Vrs.Sb Hook ChDb1 ChDb2 ChrsS Vx.Sb Inst1 Inst2 Midi Midi1 Midi2 InstSb RmVB Clr.VB Vrb.M SlpbM CstCH XL.DL ST.L St.R Mn.Dl Md.Dl Fx.Sb Master MxSqz Prin
Pro Tools File Edit View Track Region Event AudioSuite Options Setup Window Help
Wed Jul 25 10:08 PM
Edit: The Hammer Of Dreamland
26| 1| 535
Pin01 Pin02 Pin03 P04L P04R PnCh Pin06 Bass SynB Strng VlnAr Bras Kc01 GtKc Kc02 Sn01 Sn02 HghH HghT MdT LwT Ride Crsh Mstr1

(a) The mix window is an on-screen mixing board, with tracks oriented vertically

(b) The edit window shows waveforms over time, with tracks oriented horizontally and stacked vertically

Figure 2.5. The Pro Tools mix and edit windows



INSERTS A-E	INSERTS A-E	INSERTS A-E	INSERTS A-E	INSERTS A-E
D3 ExpGate	EQ3 7-Band	D3 ExpGate	D3 ExpGate	
D3 CL		EQ3 7-Band	EQ3 7-Band	
EQ3 7-Band		D3 CL	BF-76	
SENDS A-E	SENDS A-E	SENDS A-E	SENDS A-E	SENDS A-E
I / O	I / O	I / O	I / O	I / O
no input	Kick Bus	no input	no input	SnrSmplPrnt
Kick Bus	Drum Bus	Drum Bus	Drum Bus	Drum Bus
AUTO	AUTO	AUTO	AUTO	AUTO
auto read	auto read	auto read	auto read	auto read

Figure 2.6. Plug-ins are added to a track using these insertion points above the fader in the mix window

on consumer audio equipment, level affects how much of a track's energy is allowed to pass, but the actual amount of energy depends on the amplitude of the underlying signal. EQ involves altering the frequency content such that certain frequencies are made more or less prominent. This alters the harmonic makeup of the sound, affecting its timbre. Dynamic processing affects how loud or soft a track is allowed to get. Compression[3] and limiting are examples of dynamic processing. Panning is where a track is placed from left to right in stereo space.

In a DAW, effects like EQ and compression are applied to a track using what is called a *plug-in*. Analogous to outboard effects hardware, plug-ins are independent pieces of software that include their own user interfaces and audio processing capability. Some are included with DAW software, while many others are sold as independent products. They are added to tracks using insertion points, shown in Figure 2.6.

[3] Not to be confused with data compression.

In order for designers to determine what representations of audio best facilitate users' goals, it is important to understand what mix engineers do, how they think about audio, and how they use their equipment. Therefore, I conducted the qualitative study described in the next section to better understand these professional practices.

2.3. Method

2.3.1. Positionality statement

I grew up surrounded by music and audio. My father is a pianist and composer who ran a recording studio for much of my childhood. My mom is a flute player from a family where getting together meant playing music together. Falling asleep to my dad playing Bach or Chopin on the piano was the norm, and singing three-part harmony with my family was not unusual. In fifth grade I asked my dad to show me how to use a mixing board. After that, I was hooked on live sound. I ran sound in school through middle and high school, and went on to get a bachelor's degree in electrical engineering focusing on signal processing and wireless communication systems. During my master's in HCI, I combined my interests in audio and HCI by leading a project to research and design for audio engineers in live theater. That project was published at NIME and is cited in this dissertation.

I do not have significant experience mixing in recording studios. That said, I still have some familiarity with the concepts, tools, and processes. Realistically, this may have biased me by making more familiar and expected findings more salient in my own mind. At the same time, my ability to speak somewhat knowledgeably about mixing may have increased my rapport with participants and their willingness to be interviewed. As a

	Field	DAW	Years experience
P1	Music	Pro Tools	< 10
P2	Music	Pro Tools	> 40
P3	Mastering & music	Pro Tools	~ 20
P4	Commercial	Pro Tools	< 10
P5	Music	Reason	< 10
P6	Commercial	Pro Tools	~ 10
P7	Mastering	Sequoia	~ 10
P8	Music	Pro Tools	~ 10
P9	Music & education	Pro Tools	20–30
P10	Music & education	–	~ 40

Table 2.1. Audio engineers interviewed

professional identity, audio engineering can be exclusive. My ability to claim some of that identity probably helped me access participants, as did my being demographically similar (white cis male).

2.3.2. Participants

My interview subjects represented a diverse range of experiences and specialties, from just four years out of school to having been in the business for over forty years. They worked in a variety of fields, including music production, commercial sound design, and mastering, all in Chicago (Table 2.1). They were recruited by drawing on personal and professional connections, and by reaching out on online sound engineering forums.

2.3.3. Procedure

My primary data source was a series of in situ observations and semi-structured interviews with audio engineers. I conducted ten interviews, lasting an average of two hours each ($\mu = 118$ minutes, $\sigma = 35.5$). These interviews focused on the mixing process, workflows, and tools. Interviews were semi-structured, in that I created a discussion guide, but let the conversation flow naturally.

The discussion guide started by asking participants about their background and type of mixing they do, then got into questions about process and tools. I asked engineers to try to explicitly articulate what they are listening for when they are listening to a track or a mix, and to reflect on how they balance looking at a user interface with listening to the audio. They would often also share explicit problems they encounter in their daily work. The full discussion guide is reproduced in Appendix A.

In addition to verbal interviews, I observed them doing their work in their usual mixing spaces, probing to make sure I understood what they were doing and why.

To gain further understanding of current audio engineering practices and tools, I watched YouTube videos of interviews with mix engineers explaining what they do and how they do it[4]. I also read reviews and promotional material for products, especially those mentioned either by engineers in my interviews or in aforementioned videos.

2.3.4. Data analysis

Audio from interview and observation sessions was recorded and transcribed; in the one case where a participant did not want to be recorded, I took detailed notes. I applied a qualitative analytic approach that combined deductive and inductive coding. Inductive themes were derived from post-session memoing and iterative coding. Once patterns became evident in these bottom-up codes, I began applying deductive codes. Deductive themes were drawn from HCI theory, design principles, and my experience with audio

[4] Many such videos are published by *Sound on Sound* magazine and the Sweetwater catalog.

production. This ultimately led me to identify existing HCI theory, specifically direct manipulation [**34**], as a useful lens through which to consider the data. Codes included examples of semantic and articulatory distance, the roles of visualization and auditory perception, and different ways of manipulating audio (e.g., levels, EQ, dynamics).

2.4. Findings

What follows are findings based on data from 10 interview and observation sessions. These findings cover significant breadth, so are broken down into four categories: semantic distance, articulatory distance, execution and feedback, and looking and listening. These begin with the most theoretically substantive, then zoom out to more general findings. Much of the semantic distance I identified derives from tools almost always representing physical properties of audio or its processing rather than the perceptual properties that are directly relevant to audio engineers. Additional findings have to do with the relative prominence of controls, resulting in varying levels of articulatory distance, the challenges of pre-fader visualizations, and the tension felt between looking and listening.

2.4.1. Semantic Distance

Semantic distance is present when there is distance between the actual meaning of a representation and how a user uses the representation. In this case, the actual meaning of existing representations largely has to do with physical properties of audio signals, not how each track sounds to a human listener.

Of course, there is one representation that has no gaps at all between meaning, presentation, and goal: the audio itself. Listening alone can be frustrating for certain tasks, though, because sound is unable to persist over time, but being able to see sound over time is important for parts of the production process (P2, P4, P9, P10).

Engineers described the main technical goal of mixing as making sure everything that should be audible is audible. With no adjusting, certain sounds tend to mask other sounds. This is especially true if they share similar frequency content.

The waveform — a graph of amplitude over time — is the dominant representation of sound in a DAW; waveforms make up the bulk of the edit window (Figure 2.5b). While dominant, waveforms are not useful for much of mixing. In describing waveforms, P4 said, "There is definitely a visual element to it, but it's somewhat minimal." He went on to describe two specific uses for waveforms. The first, for zoomed-out waveforms, is to try to make two sections of audio approximately the same volume. This use was echoed by P3 and P9.

> "If I was to bring in a couple different sound effects, it's nice to see their waveforms because then you can see if one is really f—ing loud and one is not that loud. Cause then you can just even it out visually before you even hit the play button."
> –P4

P4 then described his primary use for zoomed-in waveforms: cleaning up minor glitches. This was also echoed by P9. However, this is actually more a part of the editing process than mixing.

> "If I have to draw out some weird sound, that's nice to be able to zoom in and see exactly where that sound is, rather than just scrubbing around

> and listening.”
>
> –P4

In general, uses for waveforms tended to be more about general trends in loudness over time, such as knowing when an instrument starts or stops. With the exception of drawing out glitches, no uses of waveforms in the mixing process (as opposed to editing) depend on their detailed form; they are simply a way of visualizing loudness. Yet waveforms are an approximate, naïve way of visualizing loudness, because there is distance between a waveform and the perceived loudness to a listener. This distance is the result of nuances of human auditory perception. What the perceptual system perceives as loudness is a combination of physical amplitude, frequency content, and other sounds in the listening environment. Because a user is trying to interpret physical amplitude alone as loudness, there is a difference between the direct meaning of a waveform and the information someone is trying to glean from it. This distance is semantic in nature because what differs is the actual *meaning* of a waveform from the meaning users are attempting to infer from it. In other words, when the goal is to know how loud something sounds, seeing amplitude only takes a viewer part of the way there, so there is semantic distance.

An important thing to understand about the waveforms displayed in DAWs is that they represent the audio on the raw track, before any processing by plug-ins or level adjustments by the fader. In other words, what is shown on the timeline is the “pre-fader level” (PFL). This means changes made to the waveform are heard, but not displayed. A relatively new feature called “clip gain” increases or decreases the pre-fader level of a segment of audio, making the change evident on the timeline. This reduces the semantic distance between the waveform and its (approximate) meaning as loudness because level always impacts the perceived loudness of a clip, but only clip gain makes a waveform appear proportionally bigger or smaller. For this reason, engineers expressed an appreciation for it (P3, P4, P5, P9). Some even found that it reduced how much automation they did because they could match volume on short segments with pre-fader gain as opposed to having to change the level or compression with automation (P4).

> "I typically won't automate compression settings, 'cause now you can do it with clip gain, and kinda even it out before it hits the compressor so nothing's too much."
>
> –P8

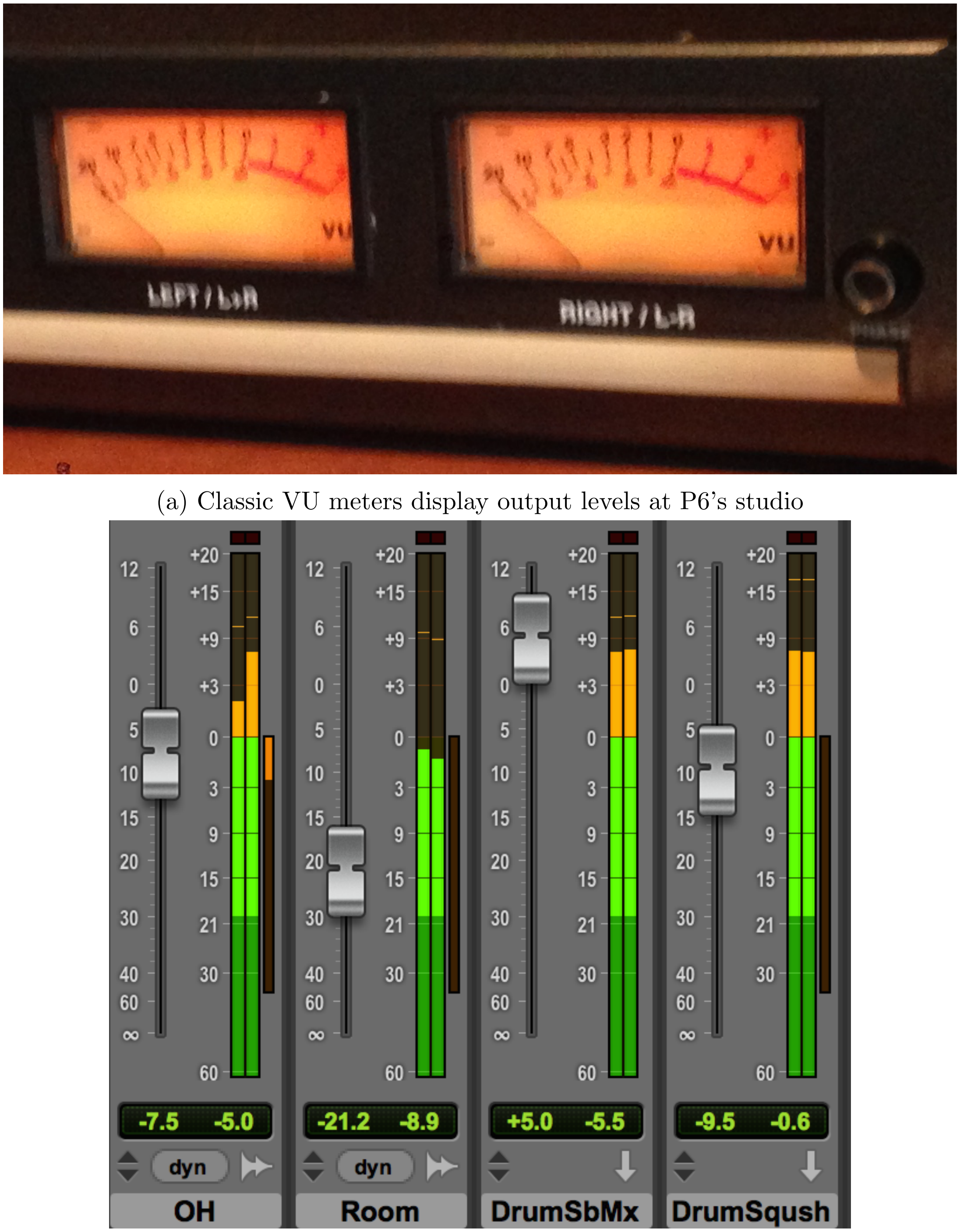
Audio is also represented in real time using several types of meters (Figure 2.7). These generally show one of several types of values related to the amplitude. In mixing, meters are used extensively for calibration, when the electrical amplitude of a signal is the important value, but not for understanding how audio sounds (P3, P4, P8).

In talking about his analog mixing board, P8 said

> "I also get the metering that this board has to offer, too.. . . Calibrating everything, knowing what's happening to individual things."
>
> –P8

More common in mastering and broadcast settings are meters designed to show a more perceptual measure of loudness. The purpose of this is compliance with regulatory and distribution channel loudness specifications (P3, P4, P6)[5]. While such meters can show

[5] For example, Spotify's documentation says, "Target the loudness level of your master at -14 dB integrated LUFS and keep it below -1 dB TP (True Peak) max." [**57**] LUFS is a standard loudness measure.



(a) Classic VU meters display output levels at P6's studio

(b) Meters next to the faders in the Pro Tools mix window

Figure 2.7. Examples of meters

loudness, they are not used to show the loudness of individual tracks, nor do they allow interaction or loudness adjustments; they are just meters.

When mixing, the user objective of making each track audible often involves altering the loudness of individual tracks by adjusting their levels, but meters on individual tracks show electrical/acoustic amplitude, not loudness. Using amplitude to infer loudness is an example of semantic distance.

In addition to adjusting level, another main way of reducing masking to make two tracks distinguishable is by using EQ to make sure that the two tracks sound good while leaving room for each other in the frequency spectrum. Engineers use words like "sculpting" (P1), "carving" (P3, P4), or "scooping" (P10) to describe this process. As these words suggest, it is a mostly subtractive process, removing excess acoustic energy that is not musically integral, especially when the heart of another instrument's sound is in that frequency range. Yet engineers do not generally use frequency-domain representations like spectrum analyzers.

> "When you think about the entirety of the mix and if you're dealing with even just like a lead vocal. . . or like an acoustic guitar might have a little bit too much information because you're so close to it when you record it. But you need to kinda sculpt out just enough, so that you can hear it with the bass and it's not fighting each other."
>
> –P3

When they do look at spectrum analyzers, engineers have a very acute awareness that what they are looking at does not directly correlate to what they hear; there is semantic distance between the representation and the goal. For example, as mentioned earlier, higher frequencies are perceived as louder than lower frequencies at the same level. Two participants explicitly mentioned Fletcher–Munson curves (P4, P8), which plot the relationship between frequency and sound pressure level at a given perceived loudness [**26**]. The distance between engineers' interfaces and the audio they hear is semantic because, thanks to human perception, the information shown in the interface has a different meaning than the information they need in order to mix. Awareness of such properties of human perception helps them bridge this distance.

The shape of the frequency response curve is important for giving engineers a sense of their EQ (Figure 2.9). Through experience and training, they learn what frequencies or bands are associated with particular aural phenomena (P2, P5, P9). Table 2.2 shows a variety of aural phenomena, which frequencies they are associated with, and how adjusting those frequencies will likely impact the sound. Even with this knowledge, participants would consistently scan or sweep through a range of frequencies to find where they wanted to cut (or occasionally boost). This was done with live audio feedback. To ensure that they could hear what frequencies were being affected, they would often make exaggerated movements, significantly boosting a filter while sweeping through the spectrum to make it more salient, then cutting once they found the desired spot. Such sweeping was almost always done through direct manipulation of the frequency response curve, but other parameters would often be tweaked using on-screen knobs. All of this was typically done with no visual feedback about the spectral content of the underlying audio; listening was vital.

Frequency	Uses
50 Hz	(1) Increase to add more fullness to lowest frequency instruments like foot, toms, and the bass. (2) Reduce to decrease the “boom” of the bass and will increase overtones and the recognition of the bass line in the mix. This is most often used on loud bass lines like rock.
100 Hz	(1) Increase to add a harder bass sound to lowest frequency instruments. (2) Increase to add fullness to guitars, snare. (3) Increase to add warmth to piano and horns. (4) Reduce to remove boom on guitars & increase clarity.
200 Hz	(1) Increase to add fullness to vocals. (2) Increase to add fullness to snare and guitar (harder sound). (3) Reduce to decrease muddiness of vocals or mid-range instruments. (4) Reduce to decrease gong sound of cymbals.
400 Hz	(1) Increase to add clarity to bass lines especially when speakers are at low volume. (2) Reduce to decrease “cardboard” sound of lower drums (foot and toms). (3) Reduce to decrease ambiance on cymbals.
800 Hz	(1) Increase for clarity and “punch” of bass. (2) Reduce to remove “cheap” sound of guitars.
1.5 kHz	(1) Increase for “clarity” and “pluck” of bass. (2) Reduce to remove dullness of guitars.

Table 2.2. Frequency chart of the sort commonly found online. It shows the frequencies associated with a variety of aural phenomena, and the impacts of changes to those frequencies. Reproduced from [**20**].

Frequency	Uses
3 kHz	(1) Increase for more “pluck” of bass. (2) Increase for more attack of electric/acoustic guitar. (3) Increase for more attack on low piano parts. (4) Increase for more clarity/hardness on voice. (5) Reduce to increase breathy, soft sound on background vocals. (6) Reduce to disguise out-of-tune vocals/guitars.
5 kHz	(1) Increase for vocal presence. (2) Increase low frequency drum attack (foot/toms). (3) Increase for more “finger sound” on bass. (4) Increase attack of piano, acoustic guitar and brightness on guitars (especially rock guitars). (5) Reduce to make background parts more distant. (6) Reduce to soften “thin” guitar.
7 kHz	(1) Increase to add attack on low frequency drums (more metallic sound). (2) Increase to add attack to percussion instruments. (3) Increase on dull singer. (4) Increase for more “finger sound” on acoustic bass. (5) Reduce to decrease “s” sound on singers. (6) Increase to add sharpness to synthesizers, rock guitars, acoustic guitar and piano.
10 kHz	(1) Increase to brighten vocals. (2) Increase for “light brightness” in acoustic guitar and piano. (3) Increase for hardness on cymbals. (4) Reduce to decrease “s” sound on singers.
15 kHz	(1) Increase to brighten vocals (breath sound). (2) Increase to brighten cymbals, string instruments and flutes. (3) Increase to make sampled synthesizer sound more real.

Table 2.2. Frequency chart of the sort commonly found online. It shows the frequencies associated with a variety of aural phenomena, and the impacts of changes to those frequencies. Reproduced from [**20**].

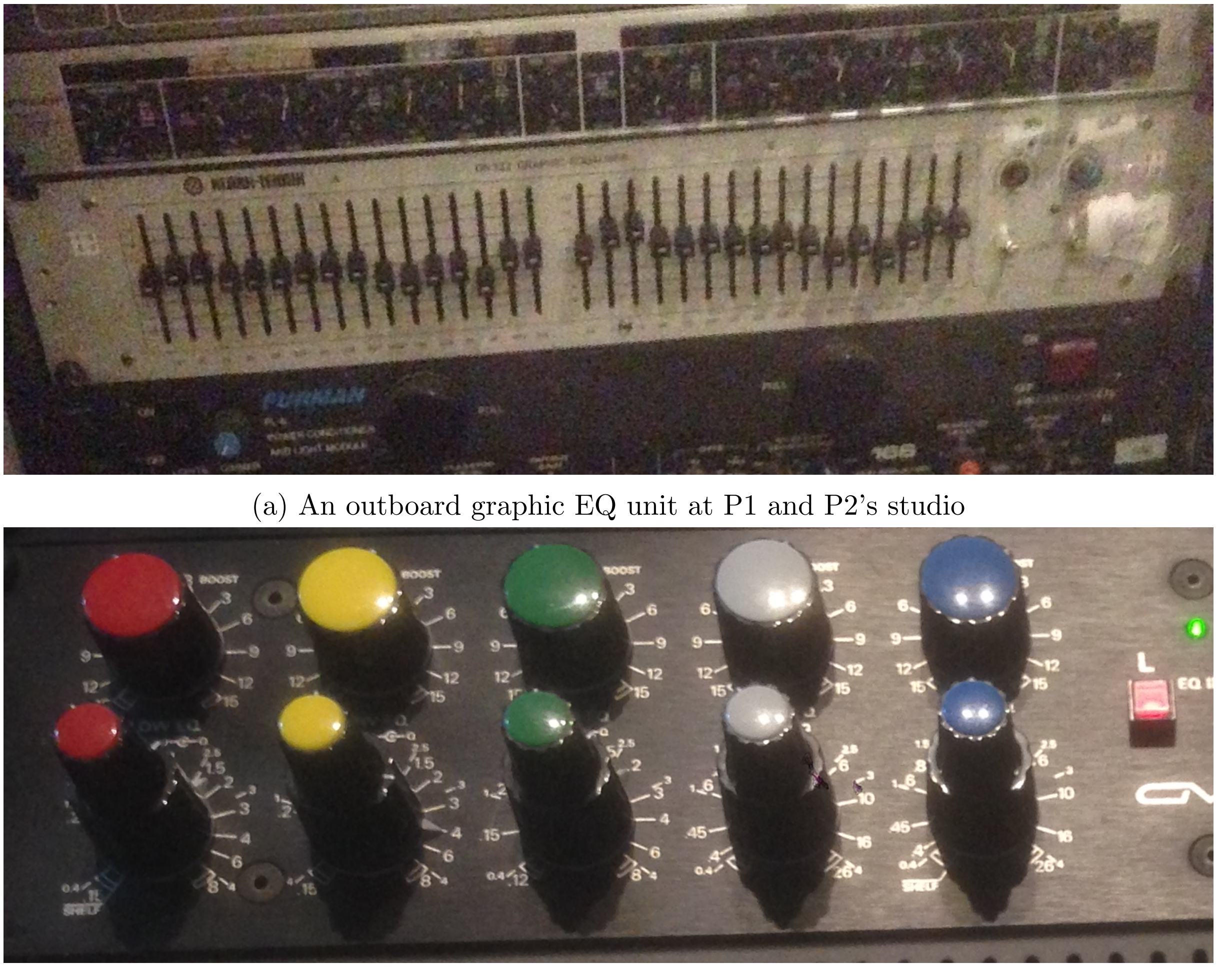
2.4.2. Articulatory Distance

There were also many instances of articulatory distance. Articulatory distance is the distance between the meaning of a representation of sound and its physical form.

EQing is a common task in mixing, editing, and mastering. In mixing, it is used to reduce masking to help make tracks distinct. The process relies on entirely different representations of audio than the waveforms described in the previous section. Working in the frequency domain suggests a need for frequency-domain representations (representations of pitch and harmonics), yet, unintuitively, it is relatively rare to see audio represented in the frequency domain during the production process except while EQing. (There was much more attention to spectrum analyzers during mastering (P3, P7).) This makes this part of the mixing process extremely listening intensive.

There are two traditional styles of user interface for analog equalization: graphic and parametric (Figure 2.8). Graphic EQ is made up of a bank of closely-spaced filters, with the gain of each controllable by a slider. This has the effect of "drawing" the frequency response curve of the EQ filter with the placement of the sliders. Parametric EQ generally has three knobs for each filter: frequency, gain, and Q (filter width). The ability to adjust the frequency and Q makes parametric equalizers more flexible in some ways, but do not give the visual feedback about the shape of the response curve.

These two interface styles represent very similar information, yet have very different physical forms — knobs versus a series of sliders. Parametric EQ gives engineers fine-grained control over individual filter settings, but does not directly provide a sense of the shape of the overall frequency response curve. On the other hand, graphic EQ offers a sense of the frequency response curve, but individual filters are less configurable. Because



(a) An outboard graphic EQ unit at P1 and P2's studio

(b) A parametric EQ unit at P3's studio

Figure 2.8. Types of EQ

of their different physical forms, they also have different articulatory distances. The words participants used to describe EQing — “scooping”, “sculpting”, and “carving” — combined with the standard design of frequency spectrum plots and many on-screen EQ interfaces, point to frequency response curves being the way people picture their equalization. While parametric EQ offers sonic advantages over graphic EQ, its lack of response curve means it has greater articulatory distance from how people picture equalization.

On the software side, many EQ plug-in designers have taken a hybrid approach by showing the frequency response curve generated by filters whose parameters are adjustable with knobs (Figure 2.9). In addition, the shape of the filter's response curve is often directly manipulable by dragging points on the curve. A common behavior was to take advantage of this hybrid interface by manipulating the frequency response curve directly, then fine tuning or adjusting the Q with the knobs (observation of P3, P4, P6, P7, P8). These EQ designs suffer from the problems of semantic distance that arise from Fletcher–Munson curves, frequency masking, and other perceptual issues discussed in the previous section, but the hybrid design that includes the frequency response curve gives them less articulatory distance than other EQ interfaces.

The most prominent controls on a mixing board are the faders (Figure 2.1). The prominence of faders on a mixing board implies that the primary function of mixing is to adjust levels. This is sometimes true, but observation revealed that this is often not the case (P2, P3, P8). P8 described using the faders to mix certain types of music, but taking a more set-it-and-forget-it approach for others:

> "With jazz records, a lot of the times, I will mix moving the faders. A lot of other times, I'll just calibrate them all at the same level and just sum through 'em."
>
> –P8

The physical form of a mixing board, and faders in particular, sometimes aligns with how it gets used, and sometimes does not. This means the amount of articulatory distance

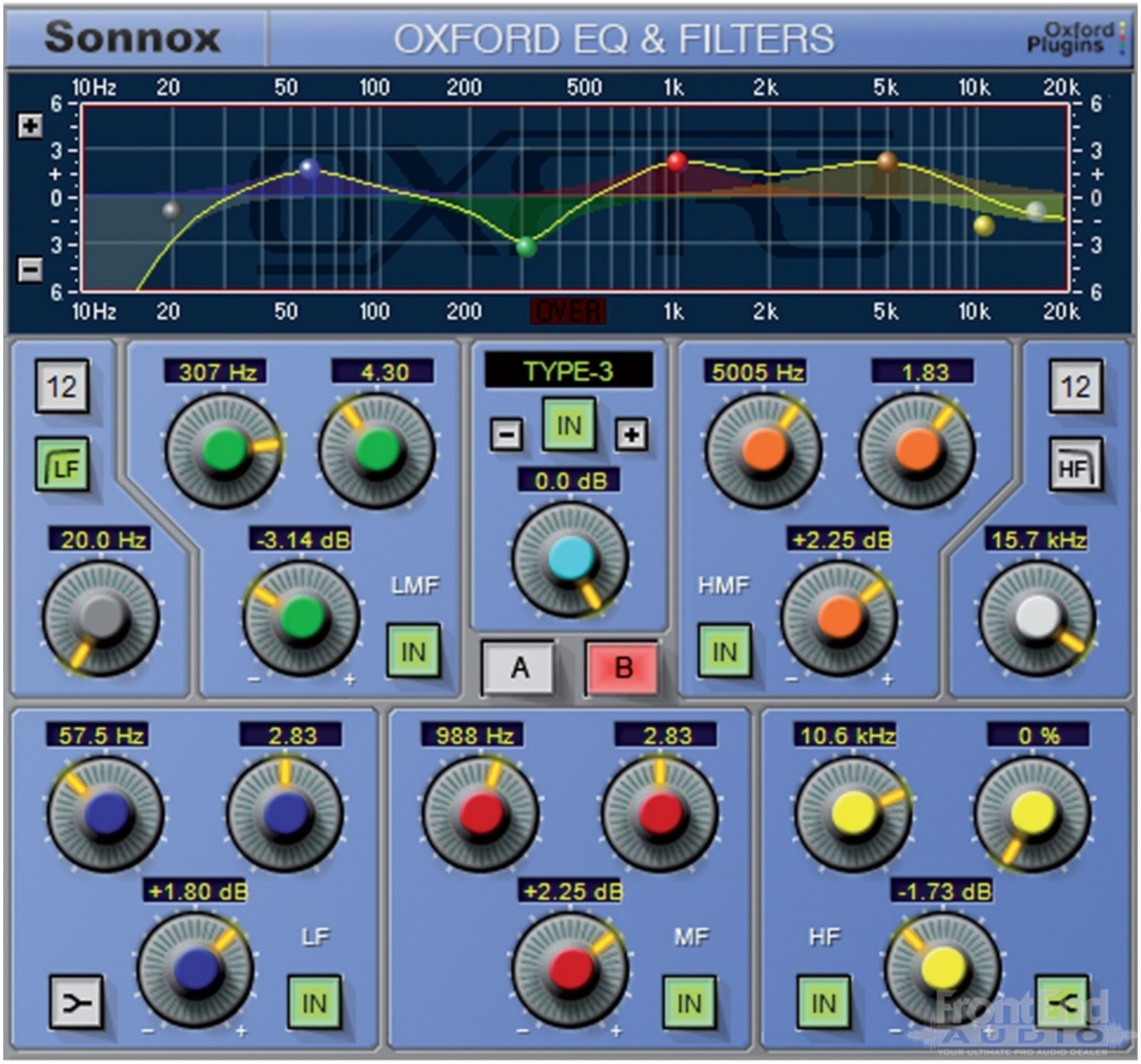


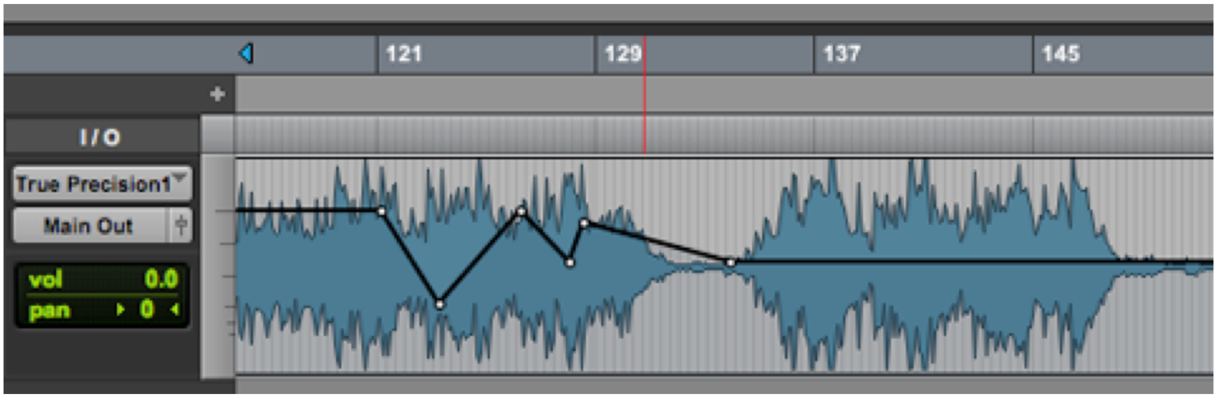
Figure 2.9. The basic EQ plug-in used by P6. It is seven-band parametric EQ where each of the seven filters is controllable by a set of knobs and button in the bottom portion while the shape of the overall frequency response curve is shown at the top. Dragging the circles on the graph at the top moves the corresponding knobs below.

between the form of a mixing board and its uses can vary from project to project and person to person.

The faders on a mixing board — or their metaphorical counterparts in a DAW's mix window — are another representation of tracks' relative volumes. As a representation, faders do not account for time, nor do they account for the loudness of the underlying audio. This makes them more useful for making coarse assessments and relative adjustments, though they do not seem to give the engineer a tremendous amount of information. They are quite useful for *interacting* with levels (P5, P8, P9, P10).

Another common manipulation of audio during mixing is panning. Along with EQ and level, it is a common approach to making tracks distinguishable by reducing masking. Panning means shifting the tracks' left–right balance so they do not occupy the same space in the stereo image. However, crafting a stereo image — for example, placing the guitar on the left and vocal on the right — is part of the creative process, and it may not be artistically desirable to impose spatial restrictions to avoid masking. It also may not solve the problem for tracks that are too similar, and the benefits are lost for listeners with mono sound systems (e.g., many built-in phone speakers).

Panning is usually only represented in knobs, making it relatively invisible despite its significant impact on a mix. Aside from these knobs, panning's lack of representation in major visualizations of audio means that, without listening, panning's impact on a mix cannot be readily inferred from the user interface. Once again, the physical form does not align with user goals, which means there is articulatory distance in the representation of panning.



True Precision1
Main Out
vol 0.0
pan 0

Figure 2.10. An automation line on the Pro Tools edit window timeline showing how level changes over time

Most parameters — levels, EQ, panning, etc. — in a mix stay constant through the course of a song, but sometimes it is necessary to make tweaks as a song progresses. This is accomplished using what is called automation. Automation lets an engineer specify when and how a parameter should be altered over time, and is represented as a line on the same timeline as the waveform (Figure 2.10). For example, a track has a volume line that can be raised or lowered. This will be discussed more in Section 2.4.3 below.

This line is a simple, effective way of representing the value of a parameter over time. In other words, there is minimal articulatory distance between this method of visualization and the value it represents.

2.4.3. Execution & Feedback

How users interact with a system, and how the system responds, are fundamental aspects of the usability of any system. How interactive the representations discussed are, and the extent to which they offer feedback to engineers, are valuable lenses through which to evaluate the tools of multitrack mixing.

In most cases, audio responds to manipulations in real time. This enabled certain behaviors described above, such as sweeping through EQ curves to find the right center frequency for a filter. Additionally, the parametric (knob-based) EQ and visual representations of filter response were linked together such that changing one would update the other.

While some visual representations update in real time, others do not, if they update at all. Again, because the edit window timeline only shows pre-fader waveforms, any non–time-based changes such as EQ or level are not reflected in the edit window. Even time-based changes such as those controlled by automation are not reflected directly in the waveform, although the automation lines can be shown on the timeline. The inability to see the effects of dynamic compression on a waveform led to work-arounds by a number of engineers to make that comparison (P5, P7, P8) and another expressed the desire to be able to do so (P9).

Further, plug-ins exist largely in their own visual worlds. For example, if multiple EQ plug-ins are applied to a track, there is no way to see the aggregate filter on that track, or if multiple dynamic plug-ins are introducing gain reduction, it is possible to see the gain reduction from each, but not the total gain reduction relative to the original audio. Another way in which representations are not linked together is the fact that waveforms in the timeline can only show pre-fader, pre-insert waveforms.

The ability to interact directly with visual representations of audio was quite limited. The most direct forms of manipulation came from automation, clip gain, and EQ. Automation represents the value of a parameter as a line on the timeline. To modify it (for example, to raise the volume), an engineer can drag it up (Figure 2.10). This can be done for many of the parameters of a track, but is most frequently used to adjust levels. Clip gain uses a contextual fader to boost or cut the gain of a region right in the timeline. As mentioned before, some EQ plug-ins allow the frequency response curve to be directly manipulated, but a filter's response curve is not a representation of the audio itself. An exception is an EQ plug-in from FabFilter which allows an engineer to see the spectra of two tracks at once, which also allows engineers to directly boost or cut frequencies from the spectral data [**1**]. However, I did not observe this interaction. This may be because engineers preferred other EQ plug-ins for sonic or other interface reasons (P5, P9).

In talking about the differences between working with just a computer and working with hardware controls, multiple engineers considered the ability to touch and manipulate multiple things at once a significant advantage of hardware controls (P2, P8, P9). Only one engineer (P8) did a significant amount of mixing on an analog mixing console (in addition to Pro Tools), but several did have small control surfaces (P2, P3, P4, P6, Figure 2.11). Control surfaces are computer peripherals that have faders and knobs that are used to control DAW software, but they do not process any audio themselves. In terms of affordances and control, P10 commented that, "A mouse is not a fader." Control surfaces fill that gap.

Before automation, mixes were performed live; that is, audio would play off a multitrack tape machine, and engineers would move the faders, adjust the EQ, and do any other necessary processing in real time as the output of the mixing board was recorded to stereo (or mono) tape. As P1 said, "We [engineers] were the automation." Engineers who use control surfaces use them to partially recreate this experience by using the faders on

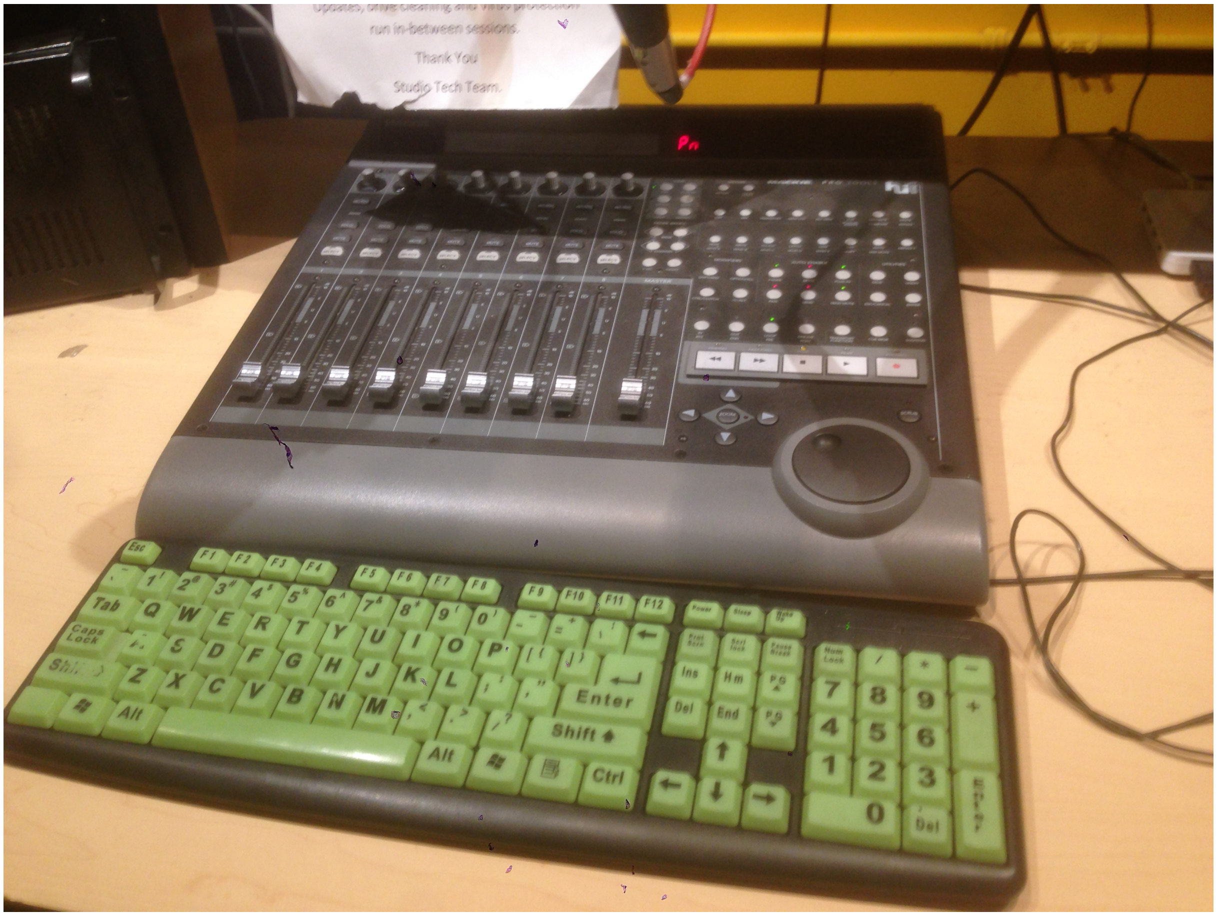


Figure 2.11. A control surface is a small mix suite in P2's studio. Unlike a mixing console, so audio passes through a control surface; it functions as a glorified keyboard/mouse for a DAW.

the control surface to record the automation, which can then be tweaked later (P2, P3, P6).

P5 described how, in mixing a documentary-style interview, he “rode the fader” up and down to balance swells in background music with the interview dialog before going back to the on-screen interface to tweak it.

> “Usually I’ll fine tune it, go in and tweak the automation by hand [on screen] a little bit, but that initial pass it’s nice to be able to have a fader to connect you to it a little bit more.”
>
> –P5

A more experienced mixer who learned on analog hardware even felt that being able to mix with faders can give different artistic results:

> “I do think that my mixes in Pro Tools working with the mouse are probably tamer and are probably more thought out or contrived than a mix on a console.”
>
> –P9

2.4.4. Looking and listening

Even with all of these visual representations, the actual object of interest is still sound. The tension many engineers feel between looking and listening is brought into focus by computer interfaces because mixing boards provided tactile feedback that made it possible to look away from the equipment rather than possibly being distracted by it (P1, P9).

Hardware controls, as opposed to on-screen controls, are a way to reduce how much looking is required. A common task is to compare two versions of the same audio that have been processed in different ways. Engineers recognized that auditory memory is short, so they relied on quick ways to toggle back and forth between versions (P3, P7, P9). This was sometimes done with on-screen buttons, but mastering engineers (P3 and P7) had hardware switches to switch between the mastered and unmastered versions.

Some take significant measures to ensure that they are truly listening and not just looking, because as P7 points out, visuals are “a natural crutch”. In her mastering studio, P7’s speakers are hidden from view to ensure that clients and engineers orient themselves to the listening sweet spot without looking at the left-to-right positions of the speakers. When he’s finishing a mix, P9 has spent so much time looking at it that he feels the need to step away:

> “When I get to the end of a mix, I’ll turn the monitor off so I’m not looking at the music anymore.”
> –P9

This tension between seeing and hearing is not new: P4 shared a story of engineers in the 1960s covering their basic VU meters (Figure 2.7a) with tape in order to hide them. While it may be apocryphal, the fact that it was told indicates that listening is emphasized in the course of training young engineers. Indeed, P9, who sometimes teaches as well as mixes professionally, said of his students:

> “They’re watching the music, and I have to remind them all the time: face the speakers! Position yourself so you’re in between the speakers.”
> –P9

There was some disagreement about how severe visual distractions can be. An engineer who has been in the business for a long time suggested that more experienced engineers may be better able to incorporate visualizations into their work because they have a better sense of both what they are looking for and what they are listening for (P2). Further research is needed to test this claim.

2.5. Conclusion

In this work, I conducted a qualitative needs analysis of audio engineers who mix multitrack audio. This investigation revealed several challenges that mixing engineers face that may be addressed by new or improved designs. I identified four main classes of challenges: semantic distance, articulatory distance, execution and feedback, and looking and listening.

I found that the main visualizations of audio — waveforms, meters, and frequency spectrum plots — are often not that useful during the mixing process. This is because of the large amount of semantic distance between these visualizations and what engineers are trying to accomplish. As detailed above, much of the semantic distance present between visual representations of sound and the goal of mixing was a result of the nuances of human auditory perception. Because the physical properties of sound — amplitude and frequency — do not map straightforwardly to human experience, visualizations of those properties also fail to do so.

Additional semantic distance is a result of what data these visualizations are representing: pre-fader audio, before any adjustments have been made or effects applied. This makes them better suited to editing than mixing. Yet in the results described above, I found evidence that engineers want to see the effect of their mixing reflected in visualizations, and that such visualizations could improve their workflows. Taken together, these two findings will motivate the designs proposed in the next chapter.

Sources of articulatory distance include the prominence of controls like faders and panning knobs relative to their importance in people's workflows, and parametric EQ interfaces that lack a visual frequency response curve. Challenges to execution and feedback stem from the isolated nature of plugin architecture, which isolates feedback and interaction within each plugin. People also liked control surfaces for input, though many did not have one. Finally, people felt an unresolved tension between listening critically to their mixes and looking at visual feedback from user interfaces.

In the next chapter, I offer design proposals based on the shortcomings identified in this chapter and the established design principle of direct manipulation for improving how audio data is visually displayed in mixing interfaces. Specifically, I propose using computational models of human auditory perception to represent audio in ways that are better aligned with how engineers perceive sound, thereby reducing the semantic distance identified in this research.

CHAPTER 3

Design and Development

3.1. Introduction

In the previous chapter, I identified two significant sources of semantic distance between waveforms — a main visual representation of audio — and the way they are used in mixing. In other words, there is distance between the meanings of waveforms and what audio engineers want from them when mixing. First, waveforms represent unprocessed (i.e., pre-fader level) audio. During mixing, this introduces semantic distance by not taking into account mixing decisions made so far, making it hard to see the impact of changes. Second, waveforms are based on physical properties of sound — namely, amplitude. This introduces semantic distance because the human auditory system interprets sound in complex, nonlinear ways that make amplitude distinct from loudness.

One could partially bridge part of this gulf by showing a waveform as it would appear after adjustments have been made, not just what the unprocessed input audio looks like. Recall that waveforms in a DAW show pre–plug-in and pre-fader amplitude. Some participants (P8, P9) directly stated that post–plug-in and post-fader waveforms that updated in real time would be useful, especially when adjusting dynamic compression. Such a design would help support interpretation by bringing the display closer to the user's goal; in other words, reducing semantic distance between the meaning of the output display and the user's goal of understanding how their mix sounds.

But since waveforms — even post-fader waveforms — show physical properties of an audio signal, waveforms can only go so far in bridging semantic distance because of the complexities of human perception. One very interesting avenue of exploration is the application of computational models of human auditory perception (e.g., [**31**]). These models can take audio signals and output how loud each one will sound in context based on each one's frequency content and amplitude. Such models could also help point out where frequency masking is likely to occur, both in time and in the frequency domain. The meaning users are trying to derive from waveforms is loudness, so by computing and representing loudness directly rather than amplitude, an interface can show the semantic value a user is interested in (loudness) rather than a value used to infer it (amplitude). This directness marks a reduction in semantic distance.

MaskerAid, the design described in this chapter, combines these two approaches. It shows post-fader audio — after processing has been applied — and uses computational models of perception to represent perceived loudness rather than physical properties of the audio. The design process described here comprises Phase 2 of the user-centered design process.

3.2. Background

This section provides the background necessary to contextualize and understand the designs discussed in Sections 3.5 and 3.3. First, I summarize the academic literature on interfaces for audio and visualization, focusing on multitrack mixing. Then I introduce computational models of auditory perception in detail.

3.2.1. Mixing in the Literature

While many developments in the design of mixing tools have come from industry, some have come from academic research. A common objective of researchers is simplifying mixing interfaces by using algorithms to partially automate the process. For example, Cartwright and colleagues created mixing interfaces that automatically generate families of mixes and allow users to explore them with just two or three degrees of freedom instead of the usual hundreds or thousands [**16**]. Similarly, Seetharaman and Pardo used descriptive adjectives to abstract away the more fundamental parameters of EQ and reverb [**53**]. These interfaces address the issue of mixing's high dimensionality, but at the expense of user flexibility and autonomy.

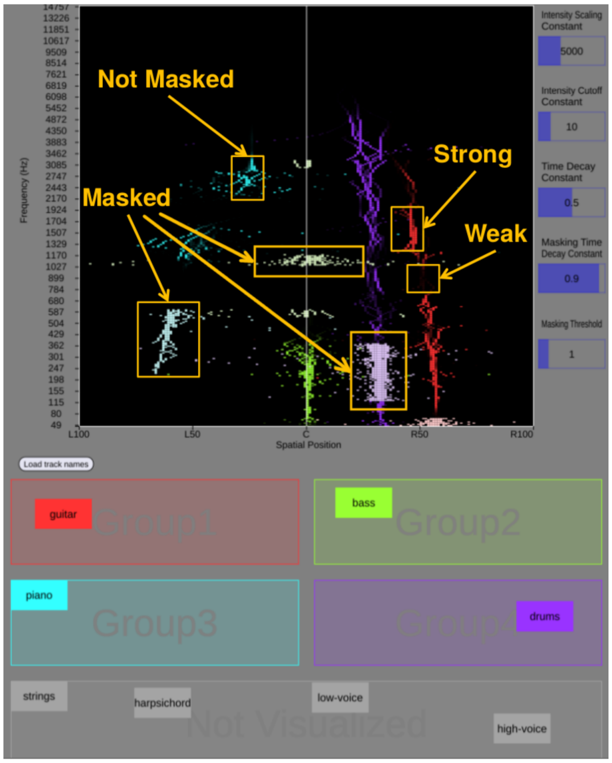
Others have made advances in fully automated mixing, removing the interface — and the artist–engineer — entirely e.g., [**50, 51, 65, 19**]. These developments are not targeted toward professional audio engineers, but rather toward the hobbyist community. They are unlikely to be directly applied by mix engineers in professional audio production because of the loss of flexibility and artistic autonomy.

Others have taken a spatial approach in which a stereo image and relative levels are manipulated by moving icons representing each track around a two-dimensional space e.g., [**15, 21, 48**]. For example, Carrascal and colleagues [**15**] created a mixing interface well suited to creating spatial audio environments using a large multitouch display, with tracks shown as point sources around a central listening position. An important drawback to such interfaces is the relative lack of frequency domain control. They addressed this by providing an equalizer, but it is separate from the spatial metaphor used by the rest of the interface. It also would not scale to many tens of tracks, as professional sessions often require.

A team at Technische Universität Berlin and Deutsche Telekom developed a spatial user interface to control the synthesized output of a spatial audition model [**30**]. While this sits at the intersection of user interfaces and perceptual models, the interface is used to control model parameters, not mix. My work uses models to generate visualizations used in user interfaces rather than using a visualization to tune model parameters.

Some research is aimed at tools for professionals, though not necessarily studio mix professionals. Cuebert was a prototype mixing board optimized for live sound reinforcement for musical theater [**39**]. It was designed using a user-centered approach, relying on interviews and observations. I follow a similar approach, but in a different professional audio context.

As discussed, visualizations of audio that are typically found in production tools are based on physical properties of sound waves. One prototype that is an important exception to this rule is MixViz [**27**] (Figure 3.1). By relying on computational models of human auditory perception, it is able to represent stereo, multitrack audio in a way that is more meaningful to human viewers. It shows the extent to which different tracks mask or otherwise interfere with each other both in frequency and in stereo space, allowing an engineer to EQ or pan appropriately. MixViz is a two-dimensional space, with left-to-right spatial position shown on the x axis and frequency shown on the y axis. Each of up to four groups of tracks has a color assigned, and is shown in the two-dimensional space. Groups that are masked are highlighted to indicate which tracks, frequencies, and positions are affected. Like a meter, MixViz updates in real time as audio plays. Although



Intensity Scaling Constant 5000

Intensity Cutoff Constant 10

Time Decay Constant 0.5

Masking Time Decay Constant 0.9

Masking Threshold 1

Figure 3.1. MixViz shows spatial and frequency masking in real time

not evaluated by representative users, it provides evidence that the use of perceptual models in mixing tools can enable new classes of tools and representations.

My approach uses the user-centered methodology of Cuebert [**39**], the perceptual algorithms of Glasberg and Moore [**31**], and audio visualization concepts for professionals inspired by MixViz [**27**] to increase the directness of direct manipulation [**54**] in multitrack mixing interfaces.

3.2.2. Visualizations

In order to work with audio over time, visualizations are necessary because they can persist. There are many ways to visualize multivariate systems (e.g., [**66**]), of which multitrack audio is an example. Of these, audio is most often represented with small multiples, which is a visualization technique in which many small versions of a visualization are repeated for different subsets of data [**61**]. This can be seen with the waveforms in Figure 2.5b. Small multiples can work well when comparing items against each other [**61**], but mixing is less a process of comparing, and more a process of combining. Parallel waveforms do not show how audio signals will sound together.

In addition to a multivariate visualization, a mixing environment is necessarily interactive. Interactive visualizations are often used to enable filtering so users can search and explore more data than would fit in a single static visualization [**24**]. An important distinction between most interactive visualizations and a mixing system is that the manipulations involved in mixing alter the underlying data being visualized, not just what is emphasized in the visualization.

In addition to visualizations of audio itself, representations of systems for manipulating audio are also necessary. For example, signal routing is often also represented as electrical signals flowing through analog hardware, or digital metaphoric [**9**] representations of analog hardware.

Most visualizations of audio itself are of its physical properties. At the most basic level there are two primary physical properties of sound: sound pressure amplitude (perceived as volume/loudness) and frequency (perceived as pitch and timbre) [**43**]. These two properties can be visually represented by simple plots. Plotting amplitude over time is a waveform; plotting amplitude over frequency is a frequency spectrum plot (Figure 2.2).

3.2.3. Computational Models of Auditory Perception

Recall from Chapter 1 that sound is made up of waves having amplitude and frequency. These properties correspond to the perceived sensations of loudness and pitch, respectively, but these relationships are not straightforward. Peculiarities of the human auditory system are responsible for the complex ways these physical properties correspond to perception within individual sounds, and how sounds influence each other when combined.

Therefore, one avenue to pursue to move beyond visualizing just physical properties of individual audio tracks is using computational models of auditory perception. Such computational models grew out of efforts to enable computers to hear like people. A primary example from auditory scene analysis [**63**] is source separation, in which a computer can algorithmically turn mixed audio into individual tracks [**63**]. The goal of my work is to use computational models of auditory perception to visualize the relationships between tracks in the context of multitrack mixing.

When a sound wave reaches the ear, it travels through the outer and middle ear before reaching the cochlea in the inner ear. At each stage, the signal undergoes a transformation, where the outer and middle ears function primarily as a series of filters that reshape the signal's frequency content [**44**].

Within the inner ear, the cochlea is the organ responsible for converting acoustic signals into electrical signals that can be interpreted by the brain. At a high level, the cochlea is a coiled-up cone whose component parts are tuned to vibrate in certain frequency ranges, starting with the high frequencies at the large end of the cone, going to low frequencies at the small end. Each of these frequency ranges is called a *critical band*,

and is often approximated as a rectangular filter and referred to as an *Equivalent Rectangular Bandwidth* (ERB). Along the way, the acoustic energy in each critical band is filtered again, this time with a filter whose frequency response depends on the amplitude of the signal in that band. The level dependence of these filters also results in dynamic compression.

Now that the original signal has been filtered by the outer and middle ears, then split up into frequency bands to be filtered again and compressed, the cochlea can convert each one of these bands to an electrical excitation pattern that is sent to the auditory cortex in the brain. For a single ERB, the excitation pattern is called the *specific loudness.* It is these excitation patterns that, when integrated over each ERB, can tell us the *loudness*, how loud something will sound to a listener.

This critical band–based understanding of human hearing can provide us with insight into frequency masking. When the frequencies of two tones are within a critical bandwidth — that is, they are affected by the same auditory filter — the louder of the tones masks the other. Depending on the extent of the masking, this could mean the less energetic signal is harder to hear than it would be in isolation, or it could be fully masked, meaning it is completely inaudible.

Natural sounds are not pure tones, but rather have frequency content at many frequencies. Despite this complication, the same principles still apply. If two sounds have significant overlap in frequency, some or all frequency components may be masked.

Researchers have developed computational models of the human auditory system based on this understanding. One of the most commonly used models of loudness is that of Glasberg and Moore [**45, 31**]. It works by breaking an audio signal down into subbands that have been experimentally shown to mimic the behavior of the human auditory system [**56**]. At each subband, or Equivalent Rectangular Bandwidth (ERB), the specific loudness N' is a function of audio excitation E in ERB number n whose constants have been determined experimentally (Equation 3.1). Loudness, then, is the integration over all ERBs of each specific loudness (Equation 3.2).

$$N'(n) = \begin{cases} C\left(\frac{2E(n)}{E(n)+T_Q(n)}\right)^{1.5} \cdot \left((G \cdot E(n) + A)^{\alpha} - A^{\alpha}\right) & \text{for } E(n) \leq T_Q(n) \\ C(G \cdot E(n) + A)^{\alpha} - A^{\alpha}) & \text{for } 10^{10} \geq E(n) > T_Q(n) \\ C\left(\frac{E(n)}{1.04 \times 10^6}\right)^{0.5} & \text{for } E(n) > 10^{10} \end{cases} \tag{3.1}$$

$$N = \int_{n_{\min}}^{n_{\max}} N' \mathrm{d}n \tag{3.2}$$

Human hearing has a remarkably large dynamic range, meaning it capable of hearing from very quiet to very loud sounds. To account for different behaviors of the auditory system at different excitation levels, Equation 3.1 is defined piecewise.

For sounds in the normal range of hearing (excitation $E(n)$ is between the threshold of hearing and 10^{10}), specific loudness is a function of excitation and several frequency-dependent constants. A is twice the peak excitation for a sine wave at T_Q; α accounts for dynamic compression in the cochlea, so is therefore less than 1; G is the gain of the cochlear filter, which is only a factor below 500 Hz; and C scales the specific loudness to the sone unit of loudness. Each of their values depends on frequency, so cannot be concisely enumerated. See [**45**] for more details.

Loudness	The intensity of an acoustic signal as perceived by a human listener
Specific loudness	The loudness of a signal in a single subband (ERB) of the auditory system
Partial loudness	The loudness of a signal as influenced by other, masking, sounds (i.e., signal in noise)
Specific partial loudness	The loudness of a signal in noise in a single ERB

Table 3.1. Definitions loudness terms

When the excitation level is very quiet, below the threshold of hearing, T_Q, specific loudness is multiplied by an additional term. This is because loudness falls off more quickly at very low excitation levels. The reason signals below the threshold of hearing have specific loudness is that a broadband signal with each ERB below the threshold can still be heard when integrated over all ERBs. On the other end of the dynamic range, when something is really loud, additional excitation contributes even more to the perception of loudness, so is modeled by a steeper relationship between excitation and specific loudness.

The definitions of specific loudness and loudness above assume a signal in quiet; that is, with no other signals playing at the same time. When a sound is heard in the presence of noise (i.e., background sounds), the perceived loudness of the target signal is called *partial loudness*. A sound in noise may be partially (or fully) masked, so partial loudness is almost always less than loudness. Partial loudness is a function of both the target and background sounds' excitation levels, and is again given in terms of the same experimentally determined constants (Equation 3.3). It, too, is defined piecewise, where each piece depends not only on the absolute excitation level of each signal, but on the relationship between the signal and masker. In addition to the terms used in computing specific loudness, specific partial loudness adds two more: T_N, the threshold of hearing for this signal in the presence of this masker; K, the minimum signal-to-noise ratio of the excitation required to be able to hear the signal when the masking noise has high excitation. C_2 is an alternative value of C when both the signal and the masker are loud. As with specific loudness, the values of these terms vary with frequency and can be found in [**45**].

In the context of this dissertation, they key take-away is that modeling human hearing is computationally complex, a fact that will become evident when discussing the MaskerAid prototype's performance.

$$N'_{\text{SIG}}(n) = \begin{cases} \begin{aligned} & C\left(\left(\left(E_{\text{SIG}}(n)+E_{\text{NOISE}}(n)\right)G+A\right)^{\alpha}-A^{\alpha}\right) \\ & -C\left(\left(\left(E_{\text{NOISE}}(n)(1+K)+T_Q\right)G+A\right)^{\alpha}-\left(T_QG+A\right)^{\alpha}\right) \\ & \times\left(\frac{T_N}{E_{\text{SIG}}(n)}\right)^{0.3} \end{aligned} & \begin{aligned} & \text{for } E_{\text{SIG}}(n) \geq T_Q \text{ and} \\ & \quad E_{\text{SIG}}(n)+E_{\text{NOISE}}(n) \leq 10^{10} \end{aligned} \\ \\ \begin{aligned} & C\left(\frac{2E_{\text{SIG}}(n)}{E_{\text{SIG}}(n)+T_N}\right)^{1.5} \\ & \times\left(\frac{\left(T_QG+A\right)^{\alpha}-A^{\alpha}}{\left(\left(E_{\text{NOISE}}(n)(1+K)+T_Q\right)G+A^{\alpha}\right)-\left(E_{\text{NOISE}}(n)G+A\right)^{\alpha}}\right) \\ & \times\left(\left(\left(E_{\text{SIG}}(n)+E_{\text{NOISE}}(n)\right)G+A\right)^{\alpha}-\left(E_{\text{NOISE}}(n)G+A\right)^{\alpha}\right) \end{aligned} & \text{for } E_{\text{SIG}}(n) < T_N \\ \\ \begin{aligned} & C_2\left(E_{\text{SIG}}(n)+E_{\text{NOISE}}(n)\right)^{0.5} \\ & -C_2\left(\left((1+K)E_{\text{NOISE}}(n)+T_Q\right)^{0.5}-\left(T_QG+A\right)^{\alpha}+A^{\alpha}\right) \\ & \times\left(\frac{T_N}{E_{\text{SIG}}(n)}\right)^{0.3} \end{aligned} & \begin{aligned} & \text{for } E_{\text{SIG}}(n) \geq T_N \text{ and} \\ & \quad E_{\text{SIG}}(n)+E_{\text{NOISE}}(n) > 10^{10} \end{aligned} \\ \\ \begin{aligned} & C\left(\frac{2E_{\text{SIG}}(n)}{E_{\text{SIG}}(n)+T_N}\right)^{1.5} \\ & \times\left(\frac{\left(T_QG+A\right)^{\alpha}-A^{\alpha}}{\left(E_{\text{NOISE}}(n)(1+K)+T_Q\right)^{0.5}-E_{\text{NOISE}}(n)^{0.5}}\right) \\ & \times\left(\left(E_{\text{SIG}}(n)+E_{\text{NOISE}}(n)\right)^{0.5}-E_{\text{NOISE}}(n)^{0.5}\right) \end{aligned} & \begin{aligned} & \text{for } E_{\text{SIG}}(n) < T_N \text{ and} \\ & \quad E_{\text{SIG}}(n)+E_{\text{NOISE}}(n) > 10^{10} \end{aligned} \end{cases} \tag{3.3}$$

Having methods to calculate loudness and partial loudness leads us to two working definitions of masking. For both definitions, masking is a ratio that represents the extent to which a target sound is masked. The first definition, given by [**56**], is the ratio of the loudness of the sum of the target and masking excitation patterns to the sum of the loudness of each of the target and masker in quiet (Equation 3.4).

$$M = \frac{N\left(E_t + E_m\right)}{N\left(E_t\right) + N\left(E_m\right)} \tag{3.4}$$

An alternative definition of masking, from Aichinger et al. [**6**], is called the masked-to-unmasked ratio (MUR, Equation 3.5). It is the ratio of a signal's partial loudness, N_{masked}, to its loudness in quiet, N_{unmasked}. As unmasked loudness approaches silence, there is a risk of dividing by 0. They therefore do not allow N_{masked} or N_{unmasked} to fall below the threshold of silence, 0.003 sones. In their experimentation, they determined that when the MUR falls below 0.1, a signal is masked to the point of being difficult to hear; in other words, fully masked.

$$MUR = \frac{\max(N_{\text{masked}}, 0.003)}{\max(N_{\text{unmasked}}, 0.003)} \tag{3.5}$$

Audio engineers think about and experience masking more from the perspective of partial loudness and the masked-to-unmasked ratio ([**6, 40**]), while audiologists think about masking in terms of the first definition. Therefore, in my work I use the masked-to-unmasked ratio.

Masking is not just a matter of level, but of the frequency content and spatial position of each signal. Ford et al. expanded Glasberg and Moore's model of partial loudness to account for spatial effects [**27**]. They also created a visualization around that model, which is an outcome similar to the goal of MaskerAid, except that it was transient instead of showing masking over time.

When currently used in mixing oriented applications, perceptual models of partial loudness and masking are largely used to simplify or automate the mixing process. Using the Glasberg and Moore loudness model, Ma and colleagues were able to accurately predict the appropriate level of music tracks relative to each other so they sounded equally loud [**40**]. Reiss and colleagues have focused on more fully automating mixing based on models of partial loudness [**51, 50, 42**]. In contrast to most previous applications of models of auditory perception to audio engineering, which target novice users by using the models in systems that automate and simplify the mixing process, the goal of this work is to design and develop tools for audio professionals.

MixViz [**27**], mentioned above, is an important exception to this rule. Unlike the other examples, it, like my work, is targeted toward audio professionals who could benefit from more powerful visualization, but who do not need simplification or abstraction. It focuses on frequency and spatial masking, and helping engineers see, in real time as their audio is playing, whether either is occurring and to which groups of tracks.

MixViz is distinct from MaskerAid, which I will introduce in the next section, though both approaches are potentially valuable for a mixing engineer. While MixViz and MaskerAid both visualize frequency masking, MixViz is more like a meter, showing masking in real time as audio is playing. MaskerAid, on the other hand, is more like a waveform, showing partial loudness and masking over time. If someone can use MaskerAid to assess relative loudness between tracks, identify instances of masking throughout a mix, and determine which tracks are causing masking on a particular track at a particular time.

Once they identify a specific instance of masking and want to resolve it by using EQ and panning, MixViz can provide useful frequency and spatial information about the masked and signals. In this way, the two complement each other.

3.2.4. Loudness Models in Industry

Most uses of loudness models in industry are for mastering, for example to normalize loudness for a broadcast or streaming service . Standard models include LUFS, a model standardized by the International Telecommunication Union (ITU) [**35, 11**], and the less sophisticated A-weighting. Har-Bal is a mastering application with the slogan "Eyes to the ears" [**2**]. Some of its features rely on loudness models, including loudness matching between songs, and automatically compensating for changes in loudness by adjusting level when the EQ of a track is changed. Most directly relevant to my work is the Masking Meter feature of iZotope's Neutron 3 plugin [**4**]. Unlike a traditional meter, the Masking Meter shows potential masking as an overlay on the spectrum plot/frequency response curve of the EQ filter (Figure 3.2). This helps users adjust the EQ to carve out spectrum to leave space for another track. It works in the frequency domain, so does not help users identify where in time masking may be occurring. It also does not attempt to identify which track or tracks is masking the selected track. This is in contrast to MaskerAid, which does identify masking in time and can help determine which tracks are causing that masking, but does not show specific frequency information.

The exact masking models iZotope is using are proprietary, but iZotope employees have published on identifying masking using the same Glasberg and Moore models I use, using an approach very similar to the masked-to-unmasked ratio [**67**].

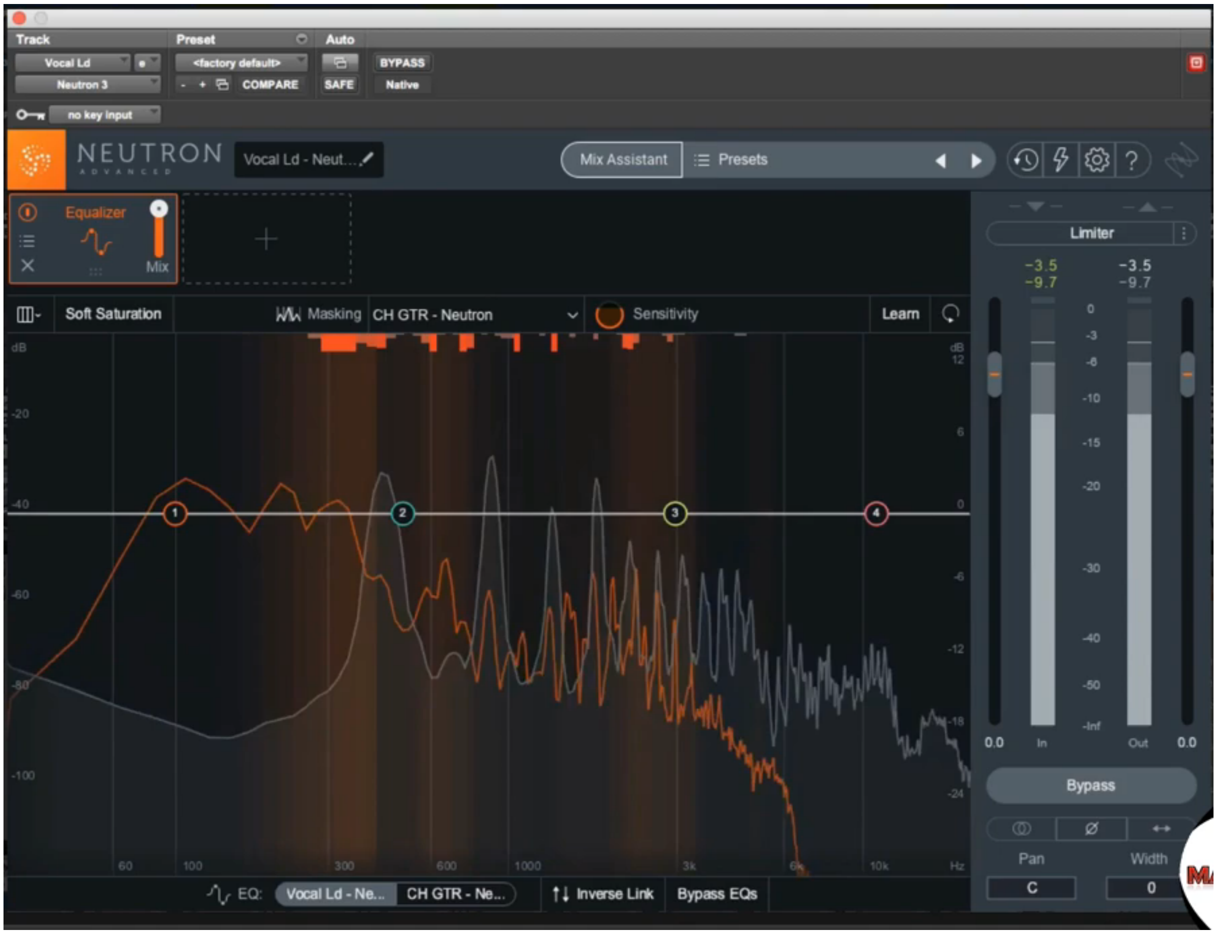


Figure 3.2. iZotope Neutron's Masking Meter highlights the spectrum in orange where masking from a selected other track is likely occurring. The intensity of masking is also indicated by an upside-down bar graph at the top.

3.3. Design Concepts

To start developing design concepts, I drew from user research. In Chapter 2, I identified two important shortcomings in how digital audio is represented visually. First, while waveforms are used in specific ways, the information gleaned from them is often inferred through expert knowledge (e.g., awareness of equal loudness/Fletcher–Munson curves) rather than applied directly (signal amplitude). While level relationships between tracks are made clear by faders in the mix window of a DAW, the loudness relationships between tracks depend on much more than their relative levels. The second is that most representations do not account for changes to a mix; they are pre-fader level (PFL), only showing unmodified input audio.

These two observed shortcomings point to a key insight: most visual representations of audio show properties of individual tracks, but what is important when mixing is the *perceptual relationships between tracks*. These relationships often play out through the related concepts of masking and partial loudness. Remember that partial loudness is the perceived loudness of a target sound in the presence of background sounds, and masking is the extent to which background sounds makes it difficult to hear a target sound. Partial loudness and masking are functions of the target and background sounds' amplitude (affected by level and dynamic processing), spectral content (affected by EQ), and spatial positioning (affected by panning) [**27**], which means there are many ways to manipulate partial loudness and masking.

Increases in processing power have dramatically improved the sound quality and increased the capabilities of computer-based audio workstations. And, as discussed in Section 3.2.3, computational models of the human auditory system are able to predict partial loudness and masking between audio sources [**45, 31, 40**]. It is now possible to apply some of that processing power toward new representations of audio that use computational models of auditory perception.

The design process began by drawing on this and other insights from Chapter 2 to create several potential concepts. Each concept involves the application of perceptual models to represent *perceptual relationships* between tracks, thereby reducing semantic distance between the representation and user goals. I do this by using these models to create manipulable inter-medium encodings of audio that better reflect the perceptual experience of listening. Each concept is also motivated by a real-life mixing scenario.

To address the issue of waveform usefulness, I initially proposed the following designs: Smart Timeline, Perceptual Waveforms, and Stacked Timeline. To improve the visibility of masking overall, in a more time-averaged way, I proposed the Pairwise Masking Map. In all of these cases, I envisioned a system in which each of a large number of tracks would update in real time as mix parameters are adjusted.

3.3.1. Design Concept 1: Smart Timeline

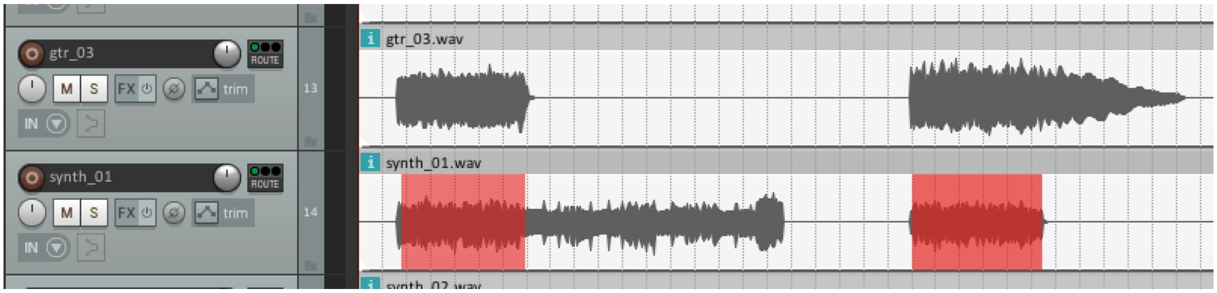


Figure 3.3. The Smart Timeline highlights areas where it thinks masking is likely to occur

> *You're mixing a song with an acoustic guitar and a singer. You've made the artistic decision that the singer should always be easily heard, and that, while you'd like the guitar to be audible throughout, it's ok if occasionally the singer overpowers it. You start out by quickly setting the overall levels and applying some EQ and a touch of reverb to sweeten the vocal. Before you go any further, you notice that the vocal track is*

> *highlighted in red for about one second a minute into the song. You go investigate, and sure enough the guitar got a little overly enthusiastic at the beginning of the second time through the chorus. You add some light compression to the guitar track, which got the red to fade a bit, but to make sure the vocal is perfectly clear, you use automation to bring the level down a bit more in that spot. As soon as you do that, the red disappeared entirely and the vocal comes through beautifully.*

Without Smart Timeline, identifying a single area where the guitar masked the vocal would have necessitated a careful listen through the entire song. While engineers should still listen carefully as they mix, Smart Timeline makes it possible to identify potential problems earlier in the process.

The waveform timeline of the edit window is a main view in any DAW. The purpose of this design concept is to highlight potential masking problems on the timeline, enabling a mix engineer to see how an adjustment to one track impacts the audibility of all other tracks across time. Aichinger et al. [**6**] found that when a track's masked-to-unmasked ratio (Equation 3.5) falls below 10% it is considered hard to hear in a mix. Therefore, in this design a track will be highlighted red under the following conditions:

$$red(t) = \begin{cases} 0 & \text{if } MUR > .1 \\ 1 & \text{if } MUR \leq .1 \end{cases} \tag{3.6}$$

This design can be thought of as a sort of spell check for masking: it can bring potential masking to the user's attention, but it does not require corrective action by the engineer.

3.3.2. Design Concept 2: Perceptual Waveforms

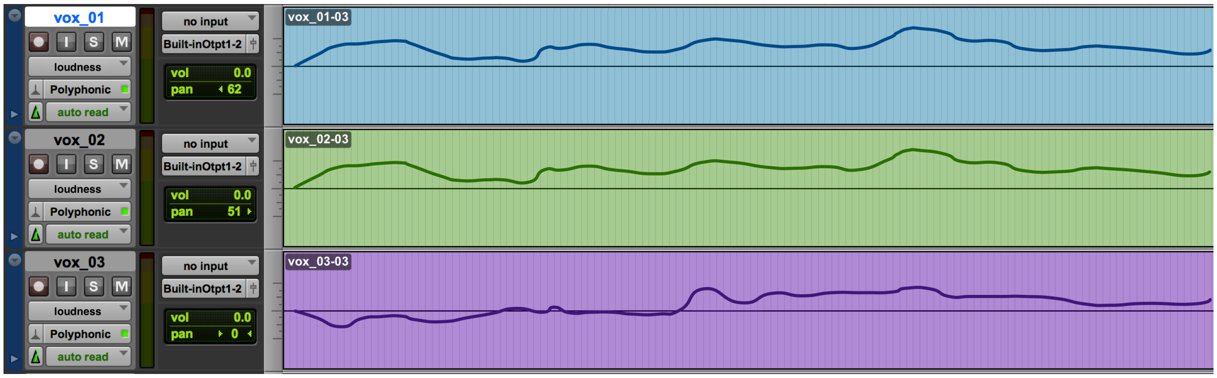


Figure 3.4. Perceptual Waveforms show the partial loudness of each track; that is, how loud each track sounds in the context of the mix

You're mixing a rock song that starts out with a tight, three-part vocal intro. The performance during tracking was good, and all three singers blended very nicely. Now that you're mixing, though, they all sound on top of each other. You pan them apart to give them some space, which improved things, but the middle part still sounds like it's being swallowed up by the other voices. A regular waveform makes it look like all three voices are equally loud, but the perceptual loudness plot clearly shows that the middle part is quieter in context. Once the band comes in and one of the voices drops out, the masked track becomes unmasked. You apply some EQ to one of the other voices to carve out some spectrum, and use automation to only apply that EQ to the intro so the rest of it sounds as natural as possible.

The challenge in this scenario is that the three tracks sound similar, so it is easy to hear that they sound on top of each other, but is harder to identify which tracks are harder to hear.

In addition to overlaying information about potentially problematic issues on the standard edit window's waveforms, the waveforms themselves could be replaced with plots of more perceptually relevant information. Instead of a plot of signal amplitude, a timeline could show a graph that better corresponds to how a track is perceived in context. For example, this could show how loud each track sounds in context — its partial loudness — over time. In such a design, it would be evident which tracks *sound* louder or softer across a mix, not just which have more acoustic energy or which are mixed at a higher level.

3.3.3. Design Concept 3: Stacked Timeline

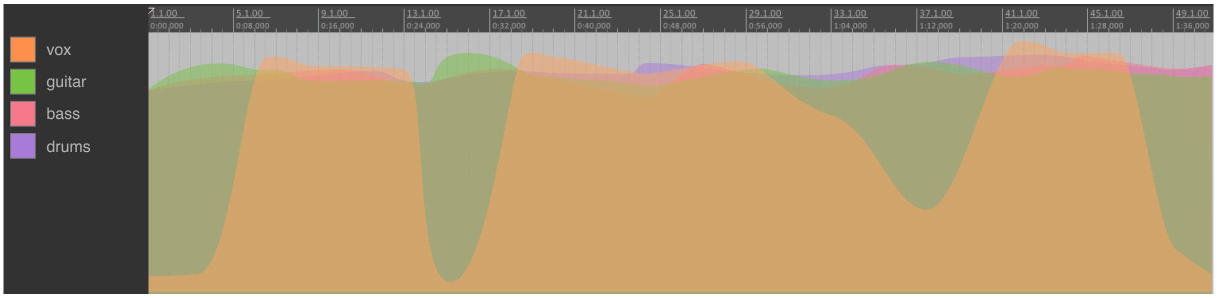


Figure 3.5. The Stacked Timeline shows where and how much tracks or groups of tracks stick out of a mix

> *You're mixing a live recording of a big band jazz tune, and want to be careful about the interplay between the rhythm section, the brass players, the sax players, and the soloists. You group your tracks to create your*

> *layers in the stacked timeline. You see right away that you need to bring the brass levels down across the whole tune. You also see that you need to use automation to bring up the sax solo because it was only as loud as the rest of the band, and add more compression to the trombone solo because her attacks on the high notes make it cut over the rest of the band too much. As you make these adjustments in specific spots, you can see whether they're having the desired effects during the rest of the tune.*

The challenge here is that mixing requires keeping track of many tracks and their changing loudness over time. Being able to see those changes over time can provide useful insight.

Rather than a timeline that shows each track in its own horizontal lane, this concept is to show a loudness-based timeline with all tracks, or groups of tracks, stacked[1]. Its purpose is to help engineers get an overall view of which tracks (or groups or stems) are most prominent in a mix, how that changes over time, and whether something that should be prominent is ever overwhelmed by the rest of a mix. This design is unlikely to scale beyond four or five layers, which is why it may require grouping tracks together. The height of each layer is determined by its partial loudness.

3.3.4. Design Concept 4: Pairwise Masking Map

While the previous three designs show information over time, it can also be valuable to show information summarized over time. This could be as an aid to an engineer in

[1] This design is similar to one described in [**32**], though I conceived it independently.

establishing a rough mix before adding automation to vary it over time, as a way to inspect a portion of a mix, or as another way to understand an overall mix.

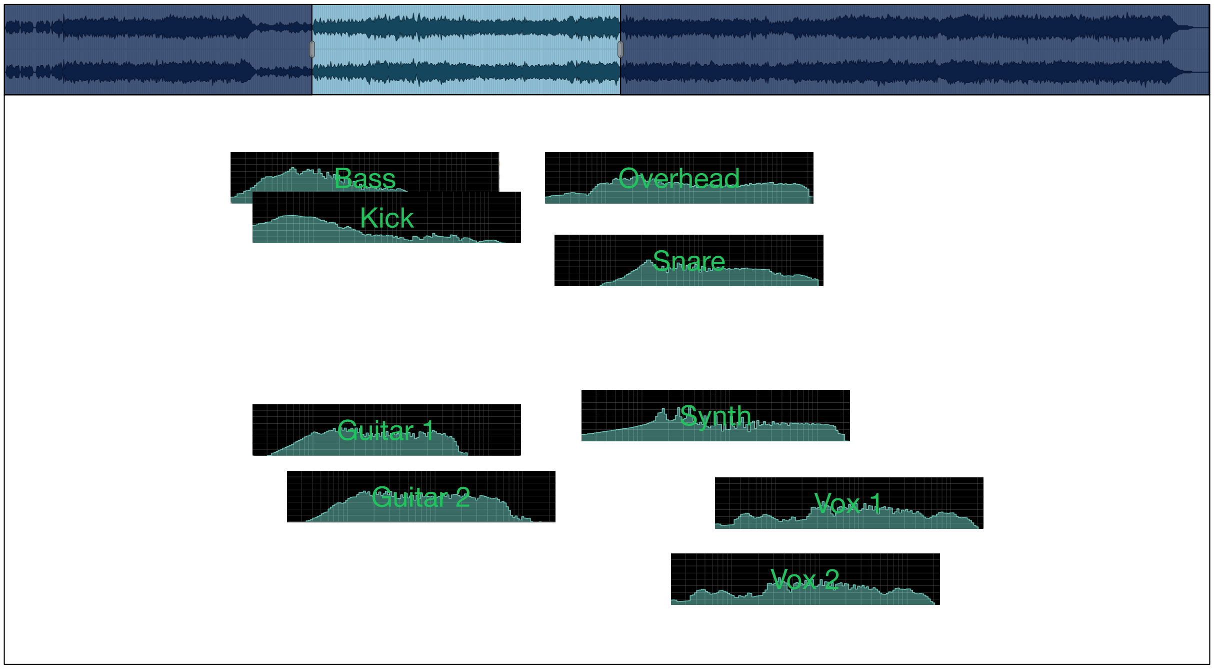


Figure 3.6. The Pairwise Masking display shows how much each track masks each other track

> *You're mixing the cast recording of the latest hit musical. Each track sounds great by itself, but something sounds mushy when you play it all together and you can't place why. You look at this visualization and see that the percussion section is right on top of the distorted guitars, meaning that the drums are partially masking the guitars. That's it! You high-pass filter the guitars to get them out of the way of the bass drum, and it's clearer already.*

When mixing, it can be clear when something sounds amiss, but it is not always evident why, especially when there are many tracks.

For a selected portion of time, this shows each track in a dimensionless space where the closer tracks are to each other, the more one masks the other. In network science terms, it is a force-directed graph of masking between each pair of tracks. The attractive force between any two nodes (tracks) is based on how much the two tracks mask each other. This is partial loudness, but where partial loudness is a track in the presence of another individual track rather than the rest of the mix. The intent is for this view to be able to update live as an engineer changes the selected time, or changes any mix parameters on any track. Unlike the previous three designs, this design considers pairwise masking between any two tracks rather than partial loudness relative to the rest of the mix. Because existing models of masking compute masking at individual time frames, a single value for each pair must be calculated by first computing masking at each frame, averaging those across time for in direction, then computing a statistic (e.g., the mean) across each direction of each pair. This design is less similar to existing, well understood interfaces. For that reason, it may be less well received by experts than the other three proposed designs.

3.4. Design Exploration and Iteration

When designing data visualizations, it is helpful to design with real data. Without this, it is difficult to create designs that will be useful and feasible. Therefore, before prototyping any of the designs described above, I generated partial loudness data from a multitrack song session. I did this using the same software libraries that model auditory perception as would go into the final prototype.

With sample data acquired, the first design I implemented was the Stacked Timeline. To try to communicate how prominent one track sounds relative to another or the mix as a whole, I experimented with different semitransparent gradient fills (Figure 3.7). The intent was for tracks that were sonically masked to also be masked visually. This turned out to be unworkable because sometimes a track that should be masked was stacked on top of those that were masking it, making it visible. This type of z stacking issue was common the stacked variant of the design.

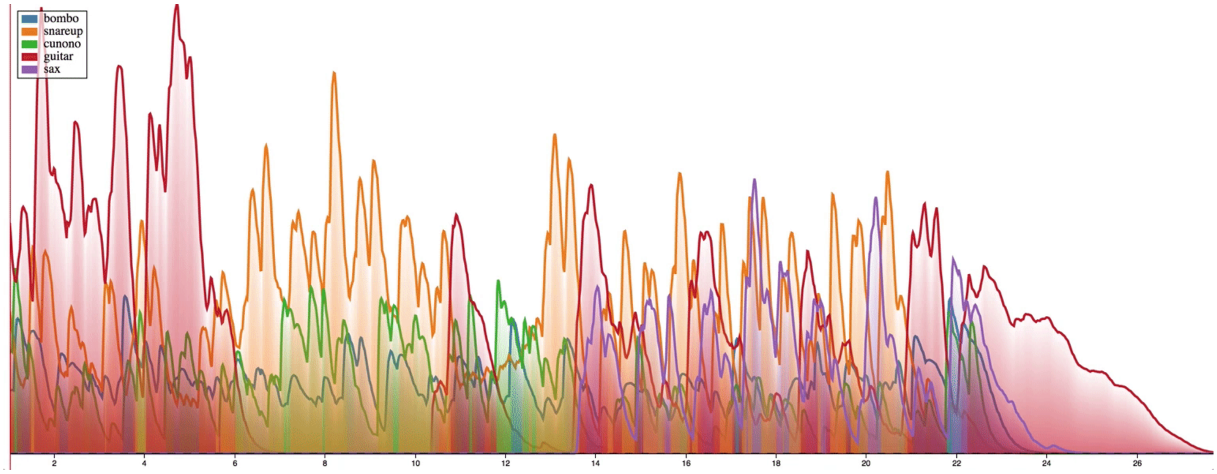


Figure 3.7. A prototype of the Stacked Timeline design. At each time point, the tracks are stacked so the loudest is on top.

I therefore abandoned the stacked design in favor of something more like the Perceptual Waveforms design. Both designs show partial loudness, so showing each track in its own lane was a good way to address the issues that came up in the stacked design while still communicating most of the same information to a user. Once each track was in its own lane, it became possible to incorporate the masking highlighting of the Smart Timeline proposal. Placing the highlighting over each track, much as in the Smart Timeline mockup, would have been too salient and disruptive for something that, like spell check, should be easy to ignore. Moving the highlights to underlines opened up the possibility of using the underline lane to make a small plot of masked-to-unmasked ratio (Figure 3.8). That turned out to be not useful for answering the question, “”How masked is this track?”, and therefore did not bridge the semantic distance of the gulf of evaluation. Visual interference with the main partial loudness visualization was also a problem.

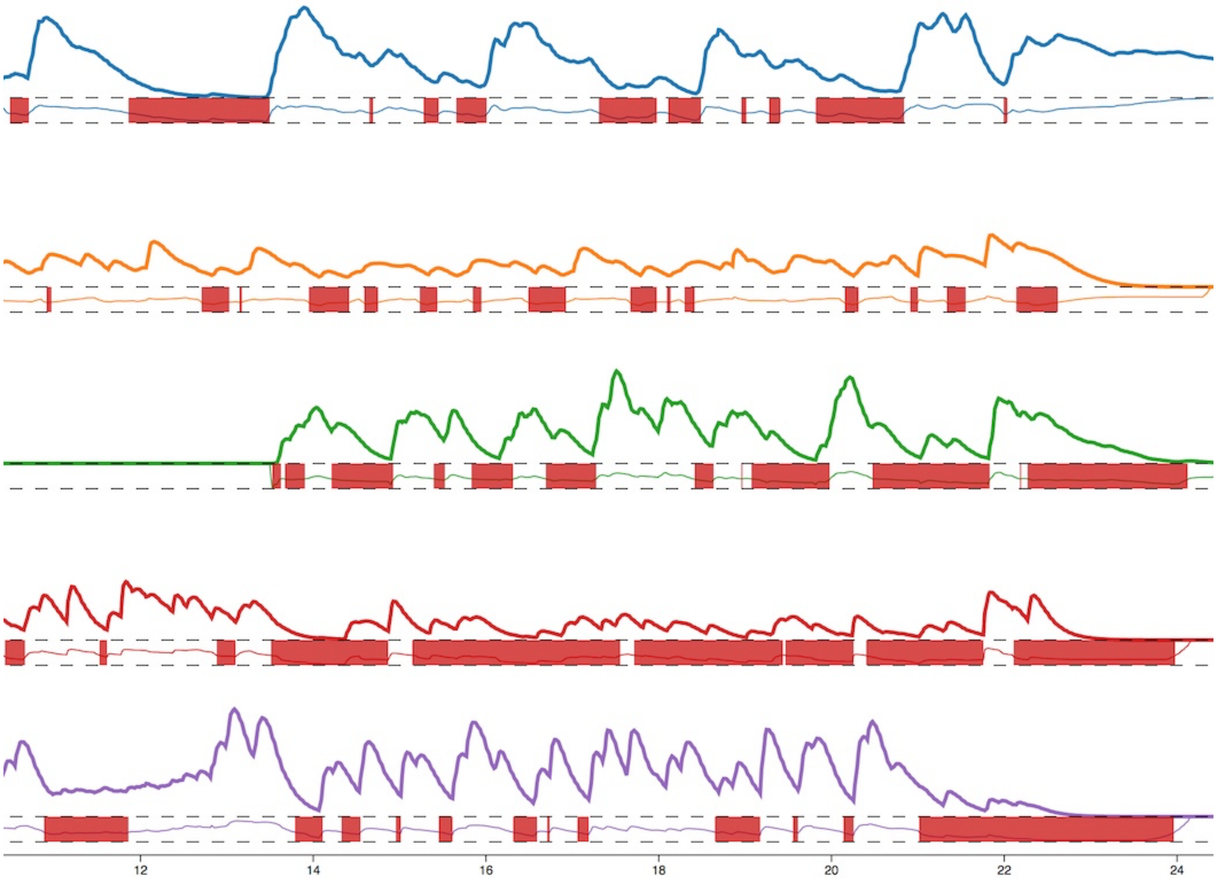


Figure 3.8. This prototype shows the masked-to-unmasked ratio in a small plot below the partial loudness curves. It also shows binary cutoffs for masked/not masked.

When developing the masked-to-unmasked ratio, Aichinger et al. found that listeners consider a ratio below 0.1 to be masked [**6**]. While I did not conduct formal experiments, in my usage this seemed a somewhat arbitrary cutoff; what a user considers too masked for their purposes depends on aesthetic judgment. I therefore added a slider to allow the user to change the masking threshold. It was interesting to see the underlining change while adjusting the threshold, but ultimately felt like giving too much control to the user while still defining masking as a binary. To address this, I change the binary underline to a gradient where there was no underline when the masked-to-unmasked ratio was above 0.1, and was at maximum opacity (50%) when the ratio got all the way down to zero.

This completed the design of the perceptual timeline, but there was still an important piece of information missing: if a track was masked, which other track or tracks were masking it? To visualize this, I drew from the original Pairwise Masking Map design, with an important change. The original design intended to show all tracks in relationship to all other tracks at once, which would require averaging how masked track A is in the presence of B with how masked track B is in the presence of A. Given that masking is likely to occur when one track overpowers another, masking is very often only an issue in one direction. Averaging across both directions would be meaningless at best and deceptive at worst.

Instead, the Masking Map treats the selected track as the signal of interest, with each of the other tracks considered a separate masker. This means the distance from the selected track in the center to one of the other tracks is determined by the MUR of just those two tracks; that is, the ratio of the partial loudness of the selected track in the presence of the other track to the loudness of the selected track by itself (Equation 3.7).

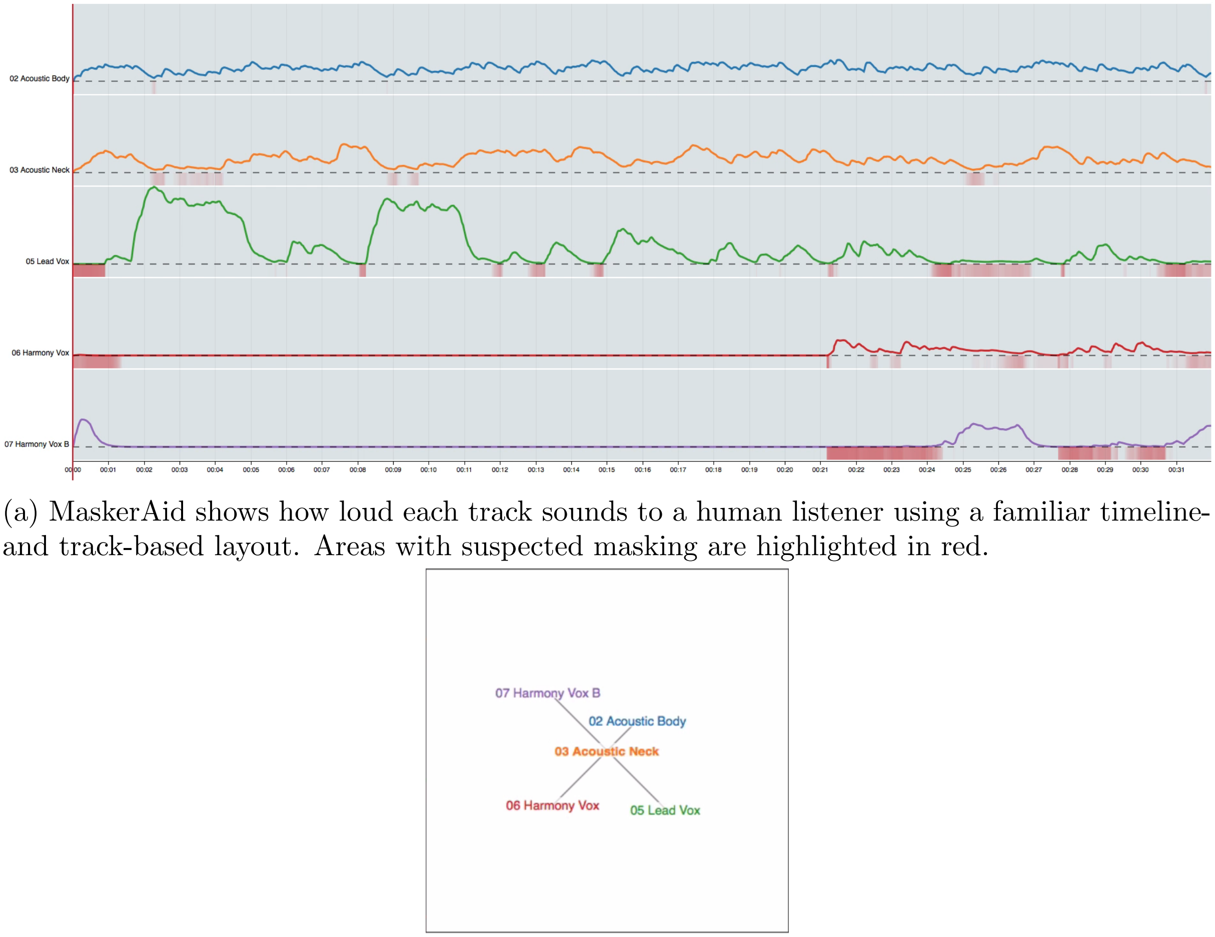
$$distance(signal, masker) \propto \frac{\max\left(N_{\text{signal masked by masker}}, 0.003\right)}{\max\left(N_{\text{signal}}, 0.003\right)} \tag{3.7}$$

3.5. MaskerAid

This dissertation introduces MaskerAid, a visualization system for multitrack audio (Figure 3.9) that was the result of the design process described above. Briefly, MaskerAid offers the following unique functionality: shows loudness over time for each track in a standard DAW-like timeline, highlights potential frequency masking on the timeline for each track, and interactively shows which tracks are causing masking on a selected track during a selected time.

The user research described in Chapter 2 found that knowing how loud each track sounds is important to the mixing process, but that existing representations, including waveforms, fail to directly represent this. Therefore, MaskerAid uses computational models to do the translation from amplitude to the more semantically relevant value of loudness. Like MixViz [**27**], MaskerAid visualizes the output of Glasberg and Moore's partial loudness model [**31**] (see Section 3.2.1), but unlike MixViz, it does so over time. MaskerAid uses the familiar timeline- and track-based layout mixing engineers are used to. Instead of waveforms, however, which show the amplitude of sound pressure, MaskerAid shows partial loudness, which is how loud a track sounds not in isolation, but when heard in the context of each of the other tracks.

Audio is transient and changes over time. As a result, changing a track by, for example, adjusting its level, can have a different effect at one point in a song than another. This is why MaskerAid shows loudness across time. It does this using a presentation familiar



(a) MaskerAid shows how loud each track sounds to a human listener using a familiar timeline- and track-based layout. Areas with suspected masking are highlighted in red.

(b) MaskerAid shows how sonically on top of the selected track each other track is using spatial distance

Figure 3.9. An overview of MaskerAid's two views

to people who are used to waveforms: a line graph where time is the horizontal axis and partial loudness is the vertical axis (Figure 3.9a). Unlike loudness meters, which show loudness instantaneously[2] as audio is playing, by showing loudness across time, MaskerAid can help a user understand the impact making a change to a mix at one point in a song will have on the rest of a song. It also makes it easier to identify likely problem areas

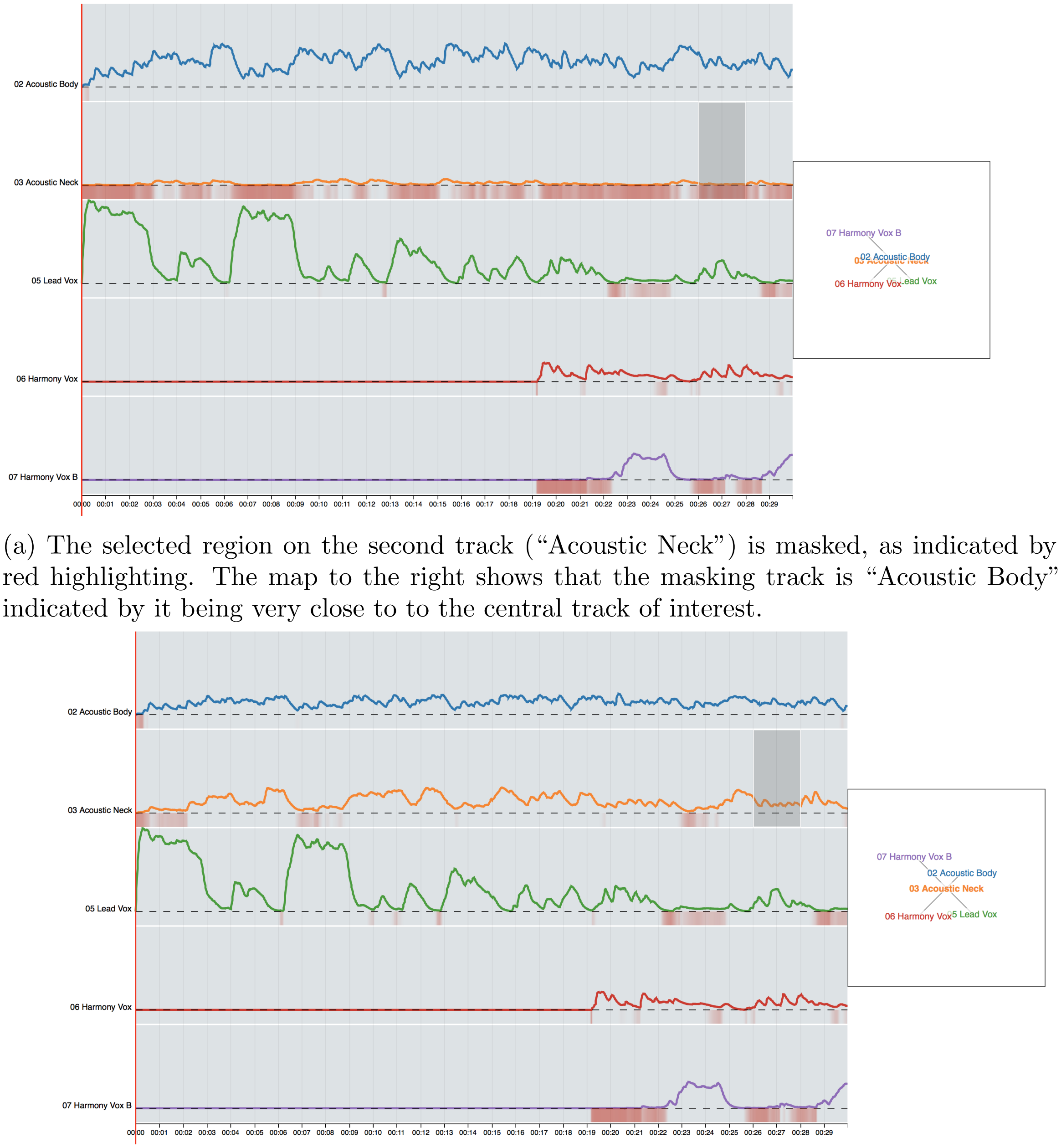
[2] Not to be confused with *instantaneous loudness*. See Section 3.6.1 and [**31**].

at a glance. This objective of being able to see the impact of changes across time also requires updates to be live. As a mix changes, MaskerAid's representations of loudness and masking should update fluidly.

In addition to showing the partial loudness of each track across time, MaskerAid — as its name implies — helps a user identify instances of potentially undesirable frequency masking. Much like how spell check underlines likely misspellings, MaskerAid underlines regions of tracks where the masked-to-unmasked ratio falls below 10%, as Aichinger suggests [**6**]. Unlike spelling, masking is not a binary true or false. Therefore, the underline is a gradient whose opacity increases as the predicted amount of masking increases.

Being able to identify where a track is masked is only part of being able to fix it. A user could likely resolve a masking issue by increasing the level of (or otherwise manipulating) the masked track, but it may be more desirable to modify the track or tracks doing the masking. To help with this, MaskerAid includes the ability to identify which track or tracks are most likely masking a selected track (Figure 3.10). To see what tracks are masking another, a user selects a slice of time of a single track. That track is then shown at the center of a space, with all other tracks some distance away. That distance is determined by how much the other track masks the track in the center. The farther apart they are, the less a track is masking the center track. The more on top of each other they are, the more on top of each other they are sonically. The design uses a two-dimensional layout because that makes it less prone to track names overlapping when they contribute similar amounts of masking to the selected track.

This design is currently implemented as a working, interactive prototype that sits adjacent to a DAW. Ultimately, I envision replacing waveforms right in a DAW with live



(a) The selected region on the second track ("Acoustic Neck") is masked, as indicated by the red highlighting. The map to the right shows that the masking track is "Acoustic Body", as indicated by it being very close to to the central track of interest.

(b) The mix has now been adjusted to reduce the masking on "Acoustic Neck". As a result, the highlighting is gone and the previously offending track, "Acoustic Body", is farther away from the center of the map on the right.

Figure 3.10. Tracks shown in MaskerAid with significant masking (3.10a) and minimal masking (3.10b)

updating, perceptually driven curves of partial loudness. This does not mean waveforms are never useful. Recall that mixing is the process by which individual tracks of audio are combined into an aesthetic and intelligible whole. Before that can happen, though, each track has to be cleaned and corrected, excess recorded material needs to be removed, and different takes combined together. These editing tasks are best achieved working with waveforms. As Chapter 2 showed, waveforms are less useful — or at least semantically more distant — during mixing than editing, but are still important for editing.

3.6. Implementation

As we saw in Chapter 2, engineers spend most of their time in a Digital Audio Workstation (DAW). In order to best integrate into existing workflows, new tools should integrate into a DAW, or at least not force people to mix outside of a DAW. On the other hand, my experience with web technologies (HTML, CSS, and JavaScript) allowed for rapid prototyping and development. MaskerAid, then, is built adjacent to a regular DAW, using data from the DAW. It gets data from the DAW, but visualizes it in a web browser.

Working with web technologies in conjunction with a DAW presented some challenges because DAW software is not generally designed to interact with the web. After a false start[3] I settled on using the DAW Reaper. Reaper is a full-featured DAW that has an important feature most others lack: a web server. It can run a web server which provides an API into certain DAW features. This allows web applications running locally to send requests to Reaper to find out the current state of a mix session.

[3] I tried to use AppleScript to get data out of ProTools and into a browser. It worked as long as ProTools was not playing back.

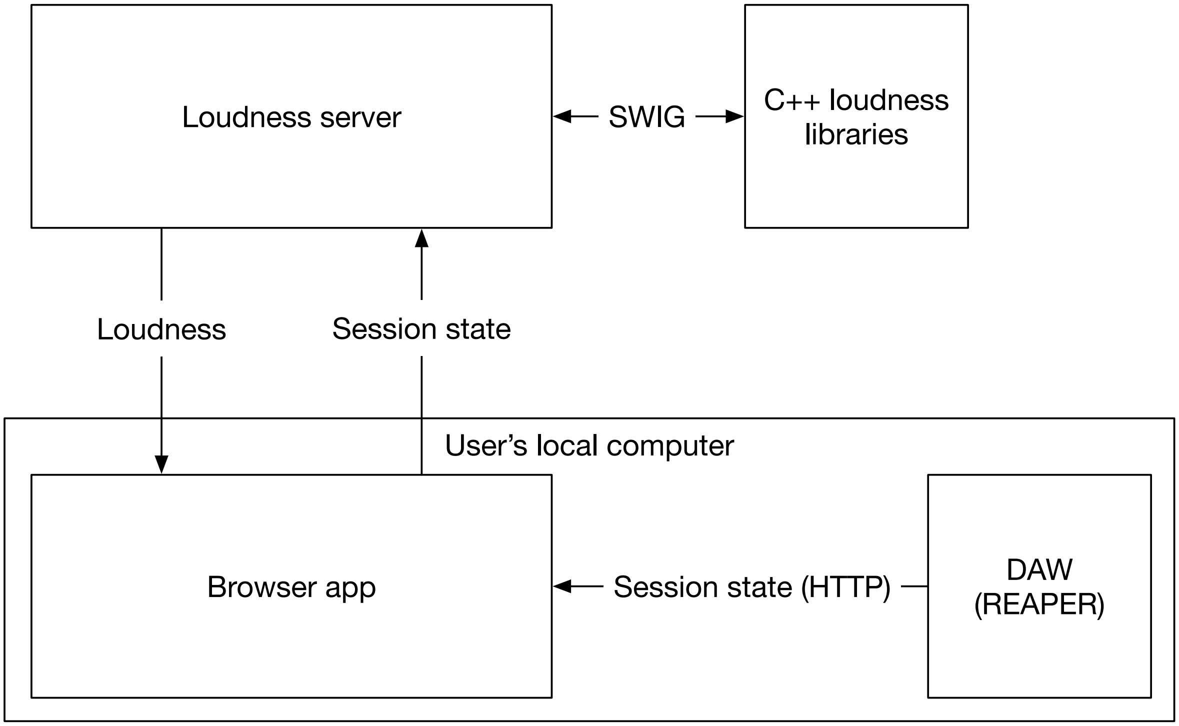


Figure 3.11. The MaskerAid system architecture is made up of three main components: a DAW, the browser-based web front end, and the loudness server. The loudness server interfaces with the loudness models [**64**]

The DAW constitutes a significant portion of the overall system, but it is only one of three major components. These three components are the DAW, the browser, and the loudness server (Figure 3.11). Having selected a DAW that was capable of providing the necessary data in a usable format, this section will discuss the other two components and how all three work together.

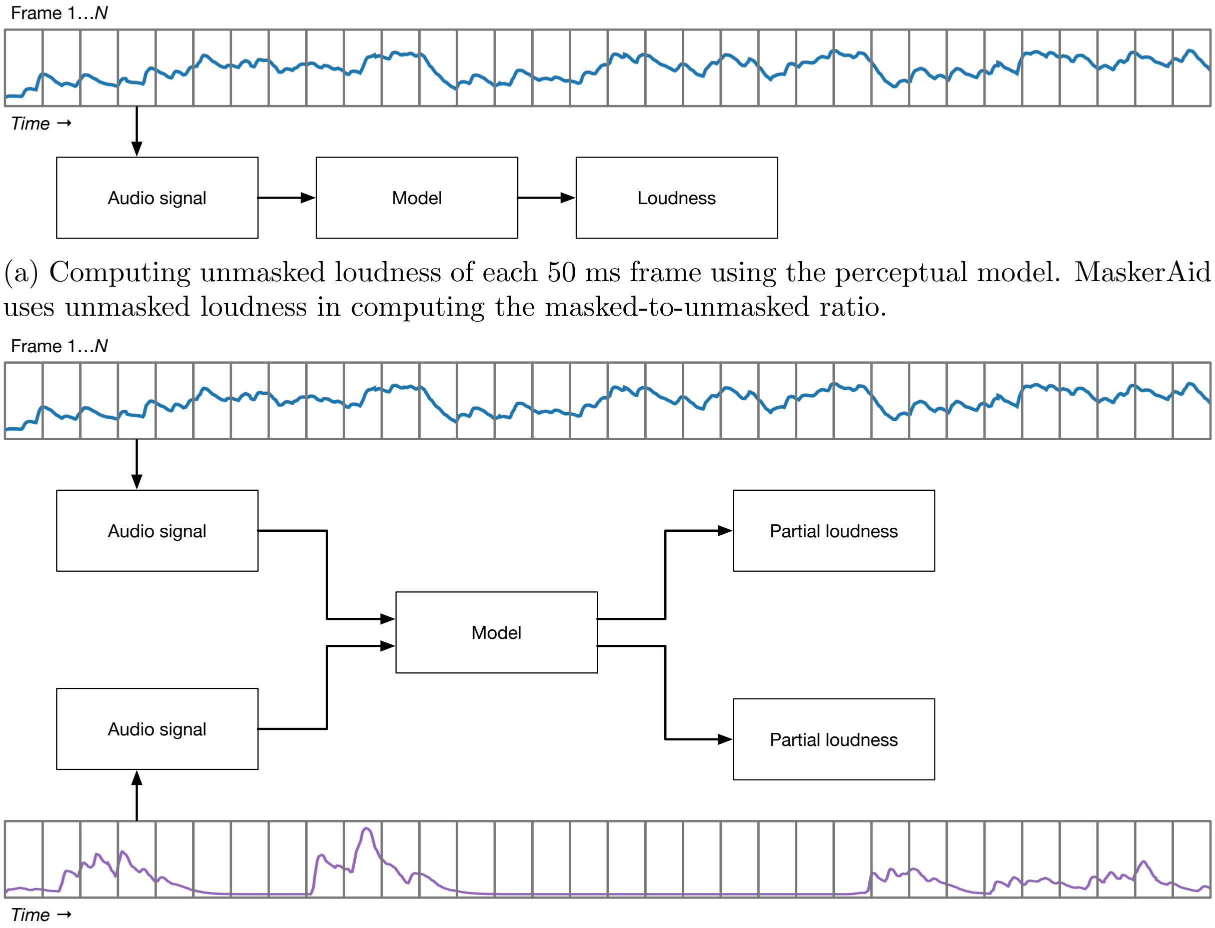
3.6.1. Loudness Server

The loudness server is a service that takes multiple tracks of audio and their mix parameters (levels over time), and returns the loudness and partial loudness of each track over time. This service is built on Ward's open-source implementation of Glasberg and Moore's 2002 time-varying loudness model [**64**].

MaskerAid works by slicing each track into 50 ms frames. Each frame of each track is run through the loudness model, yielding two values: loudness (Figure 3.12a), and partial loudness (Figure 3.12b). Partial loudness is presented directly as the loudness curve. Loudness, on the other hand, is used as the denominator in calculating the masked-to-unmasked ratio that is used to determine whether a frame is masked (see Section 3.2.3, Equation 3.5).

The loudness model has outputs for three partial loudness parameters: instantaneous loudness, short-term loudness, and long-term loudness [**31**]. Instantaneous loudness is an intermediate value useful for calculating short- and long-term loudness, but is not something that would actually be perceived by a listener. Glasberg and Moore describe short-term loudness as "the loudness perceived at any instant". While it may seem like this is the parameter that a visualization of loudness over time should show, a real listener is not typically listening to music in slices that are shorter than a few-hundred milliseconds. Since the goal is to create aesthetic music, MaskerAid shows long-term loudness. This is in line with Ward's decision to use long-term loudness in [**65**], although they also suggest that different instruments or styles may be more accurately represented with short-term loudness, or a value in between.

Architecturally, unprocessed audio files are stored on the loudness server so that large audio files do not have to be transferred with each request. Requests include the state of a mix, which is then applied to the audio on the server.



(a) Computing unmasked loudness of each 50 ms frame using the perceptual model. MaskerAid uses unmasked loudness in computing the masked-to-unmasked ratio.

(b) Computing partial loudness of each 50 ms frame using the perceptual model. MaskerAid directly shows partial loudness, in addition to using it in computing the masked-to-unmasked ratio.

Figure 3.12. MaskerAid calculates loudness for each 50 ms frame by running each frame through Ward's implementation of Glasberg and Moore's loudness model

3.6.2. MaskerAid Visualization Front-end

The MaskerAid front end implements the visualization described in Section 3.5. It relies heavily on the JavaScript library D3 [**10**] and other web technologies, including canvas and Web Workers.

In addition to being the user interface, the web front end plays an important technical role as the intermediary between the DAW and the loudness server. It detects changes in the state of a mix in the DAW, then sends the updated session information to the loudness server to get the partial loudness for each track. Similarly, when a user selects part of a track in order to use the pairwise masking map, the front end sends the request along with the current session state to the loudness server.

3.6.3. Limitations

One of the factors contributing to a feeling of directness in direct manipulation systems is responsiveness [**54**]. To feel direct, a system has to feel as if it is responding quickly to human input. Unfortunately, the computational complexity of the loudness model means that the current implementation does not reach the ideal of fluidly updating the visualization as a user changes a mix.

Computing partial loudness is computationally expensive, even with Ward's optimizations. His algorithm was designed to be real time, which means fast enough to keep up with audio as it plays. The goal of MaskerAid, though, is to compute partial loudness for an entire song nearly instantaneously as adjustments are made — much faster than real time. To get as close to this goal as possible, there were several design decisions made with an eye toward performance.

One of the main performance levers of the loudness algorithm itself is its rate, or frame size. This determines the temporal resolution of the resulting partial loudness curves. In trying with different values, I found that a rate of 50 ms balanced performance and temporal resolution for the current application.

The other main optimization technique was parallelization. The loudness server slices the audio into 16 overlapping segments — overlapping because of widowing in the loudness model's filter banks — and computes their loudness simultaneously on 16 processor cores. This sped up each loudness request from about 12 seconds to about two seconds.

3.7. Conclusion

Tools such as MaskerAid are not intended to devalue the work that audio engineers do, because being able to hear problems in a mix would still be of utmost importance, and the technical knowledge of how to resolve issues would still be required. In quoting a colleague, P10 said that audio engineers see themselves playing an important role as translators:

> "We are the bridge between the objective — the technology — and the subjective — the art."
>
> –P10, quoting a colleague

A virtuosic violinist does not have to be a master acoustician to create beautiful music, but an audio engineer is an artist who needs to understand the underlying electrical and acoustic phenomena. Having that knowledge is part of the professional identities of many audio engineers (P1, P2, P3, P8, P9, P10), so any designs that reduce the amount of technical knowledge necessary need to be carefully considered in order to not make engineers feel devalued. I believe that the proposed designs offer assistance through improved visibility, but still rely on engineers' technical knowledge and artistic talents. Ultimately, what audio engineers are looking for is not something that is easy to use; they are looking for what sounds good. It does not matter how well received a user interface is if the audio processing is not able to produce the sound they have in mind.

CHAPTER 4

Evaluation

4.1. Introduction

MaskerAid, described in the last chapter, was designed to help audio engineers mix more effectively by providing more relevant information to engineers as they work. It uses computational models of auditory perception to show how loud each track sounds in the context of all other tracks. It also accounts for the current state of the mix rather than showing pre-fader loudness. MaskerAid does these things with the goal of helping engineers achieve more transparent mixes that are less plagued by problematic frequency masking. In this chapter I describe a laboratory experiment I designed and carried out to assess the extent to which it achieved this aim. I also examine whether it accomplishes a secondary goal of improving an engineer's critical listening skills.

This evaluation serves as the final phase in the iterative design process outlined in Chapter 1. The results are mixed, with MaskerAid resulting in mixes that are more transparent, but on which people spent more time compared to not using MaskerAid. I will then discuss these results, offering design, experimental, and theoretical implications.

4.2. Research Questions

By designing MaskerAid in a way that reduces the gulf of evaluation, the fit of the tool to the task should be improved compared to traditional mixing tools alone. This is why mixes mixed with MaskerAid should be of higher quality, and should have been produced more efficiently, than mixes without MaskerAid. In an art like sound engineering, quality is quite subjective. Since the focus here is on frequency masking, we can more narrowly define quality. In the present study, participants were instructed to make each track as audible as possible. This means minimizing frequency masking, creating what Aichinger called a *transparent* mix [**6**]. Research question 1 asks,

RQ1: Does MaskerAid enable users to achieve more transparent mixes with less effort?

I approach this in several ways, first addressing mix transparency and effort separately, then together. I measure transparency using Glasberg and Moore's loudness models.

Time, number of manipulations that went into making a mix, and density of manipulations are all ways to try to infer effort. When evaluating a system that is designed for people, how much effort someone feels they expended also matters. As described in section 4.5.3, I used the NASA Task Load Index [**33**] to gather self-reported effort.

Participants were given the goal of creating the most transparent mixes as possible. As they pursue that aim, we would expect each manipulation to — on average — increase the transparency of the mix. In other words, each change to the mix should move the mix in the right direction. At some point, the amount of improvement from each manipulation is likely to level off. This is for two reasons. First, there is a practical ceiling on how transparent a given mix can get. And second, someone is likely to start with big, sweeping changes to get to a rough mix, then move to making smaller, more localized changes to dial in the final mix. By looking at mix transparency over time in this way, it is possible to see whether MaskerAid influenced how quickly people approach their maximum level of transparency.

MaskerAid visually reveals frequency masking. The ability to see frequency masking in addition to hearing it may help reinforce what frequency masking sounds like. By reinforcing what frequency masking sounds like, MaskerAid may help improve users' critical listening skills for frequency masking. To investigate this, research question 2 asks

RQ2: Does MaskerAid help improve critical listening skills for frequency masking?

By making frequency masking visually salient, exposure to MaskerAid over time can reinforce what masking sounds like. If MaskerAid influences mixing practices for subsequent mixes by helping an engineer to train their ears for frequency masking, there should be differences in outcomes such as mix transparency versus the pre-MaskerAid condition. To answer this question, I compare the pre- and post-MaskerAid conditions without considering the MaskerAid condition at all.

4.3. Method

In order to answer these research questions, I conducted a lab experiment.

4.3.1. Experimental design

The study used a within-subject A-B-A experimental design. Participants mixed three 30-second, five-track excerpts from multitrack song sessions. Each of these three mixes is considered a trial. The first and last trials — which I will refer to as *pre* and *post* — did not show MaskerAid, only traditional DAW-based mixing interfaces; the middle trial — called *MaskerAid* — did allow users to see and interact with MaskerAid. The reason for this A-B-A design is to establish a baseline level of performance from which to assess learning (RQ2).

4.3.1.1. Task. Each trial consisted of mixing a 5-track, 30 second excerpt of a song. The focus of MaskerAid in general, and RQ1 in particular, is frequency masking. Therefore, participants were instructed to try to make each track clearly audible at all times. In other words, they were asked to make the mix as transparent as possible, minimizing masking. Because of limitations in the implementation, participants were also instructed to mix using only level: in its prototype version, MaskerAid is unable to respond to EQ, panning, or other plugins, though participants were able to automate level.

Each trial was limited to 15 minutes, but participants could stop any time before this limit if they felt satisfied with their mix. They were allowed to stop when they felt satisfied because how long a mix takes can be a proxy for effort. The 15-minute time limit imposed a need for efficiency, which better replicates the real world where studio time is expensive. The time limit was also for logistical reasons, as a session generally took about 90 minutes.

4.3.1.2. Song stimuli. Each of the three trials used a different song stimulus. I took care to satisfy several criteria when selecting the stimuli. First, in order to ensure they could be processed quickly enough by the prototype system, a song needed to have enough interesting content in just five tracks or stems [1]. Second, within the 30 second excerpt, there needed to be enough variety that changing or fixing something at one part of a song would not be a good solution for other parts of the excerpt. And third, I selected

[1] Stems are audio tracks that may be composed of multiple recorded tracks. For example, a drum set may have been recorded with 10 microphones going to 10 tracks, but was mixed down to just one or two tracks to make mixing the overall song easier.

songs of fairly different genres: one was a pop-rock song, one was more metal, and one was folk/acoustic. The order of these three stimuli was fully counterbalanced between participants.

4.3.2. Procedure

Participants were brought into the laboratory for about 90 minutes. The procedure is summarized in Table 4.2. They were seated at workstations with a single 24–27” monitor (Figure 4.1). Monitors were split such that Reaper was on the left half of the screen and a browser window alternately containing questionnaires, MaskerAid, or a placeholder was on the right. To accommodate participants, the study ran at multiple physical locations, so the physical setup differed between groups of participants. Those at Columbia College and Northwestern University used external USB audio interfaces and school-supplied headphones, while others used the computer’s built-in audio output and supplied their own high-quality headphones.

Before beginning the mixing tasks, participants first completed a questionnaire to assess their demographics and background. An important part of this initial questionnaire is an assessment of their experience mixing, comfort with different DAWs, and familiarity with a number of terms that relate to frequency masking and auditory perception. (The complete questionnaire is reproduced in Appendix B.) They then watched a five-minute video that introduced MaskerAid, demonstrated its use, and showed the basics of volume automation in Reaper. All participants had experience mixing in various DAWs, but not everyone had used Reaper before. I therefore provided a quick reference sheet with some basics of mixing and automation in Reaper.

2 minutes	Intro and consent
2 minutes	Demographics and mixing experience
2 minutes	Concept familiarity pretest
6 minutes	Demo of MaskerAid, introduction to REAPER, loudness, and masking
2 minutes	Introduce the task
15 minutes	Mix first song — no MaskerAid
5 minutes	Inter-song questionnaire: task load index and self-assessment of performance
15 minutes	Mix second song — MaskerAid
7 minutes	Inter-song questionnaire: task load index, self-assessment of performance, MaskerAid-specific questions
15 minutes	Mix third song — no MaskerAid
5 minutes	Inter-song questionnaire: task load index and self-assessment of performance
4 minutes	Concluding questionnaire
10 minutes	Post-task interview
90 minutes	Total

Table 4.2. Summary of the procedure, with how long each step took

Participants were then given the task instructions described above and asked to mix the first song. As explained in Experimental Design, they mixed the first song without the use of MaskerAid. To maintain consistency, when MaskerAid was hidden, they still only had Reaper on the left half of the screen. When they decided they were satisfied with the mix, they completed a set of questionnaires. These questionnaires asked participants to reflect on their experience mixing, including how well they thought they were able to mix, the extent to which they believed they resolved frequency masking, and whether the mixing tools distracted from listening. (More details on the questionnaire is in the Measures section below.) They then responded to the NASA Task Load Index (TLX) [**33**].

They then moved on to mix the second song. For the second song, participants could see MaskerAid on the right half of their display. As with before, they had up to 15 minutes to complete the mix. Once they felt satisfied or ran out of time, they completed a set

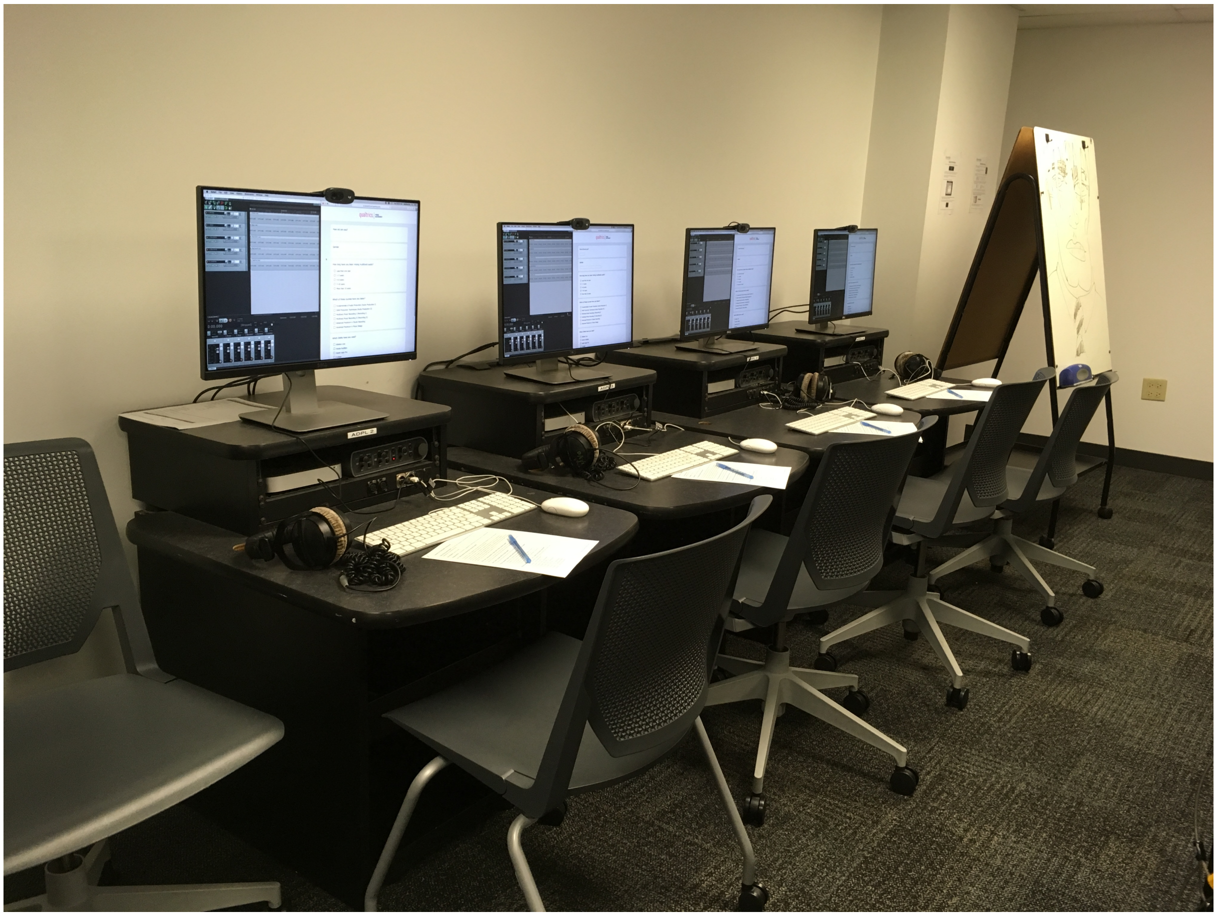


Figure 4.1. The experimental setup at Columbia College Chicago

of questionnaires. In addition to the questions that were asked after the pre trial, this questionnaire added a set of questions about MaskerAid in particular. (Again, details of these questions are in the Measures section.)

Finally, participants mixed the third song. As with the first song, they did not have access to MaskerAid. They concluded with the same set of questions as the pre trial.

The questionnaire for the third trial was followed by two measures asking about MaskerAid: the System Usability Scale (SUS) [**12**] and Technology Acceptance Model (TAM) [**18**] (details in Measures).

The session concluded with a 10–15 minute semi-structured interview. This was a participant-driven, open-ended discussion of their experience with and reaction to MaskerAid, though I did have a series of questions to direct the conversation (see Appendix C for the discussion guide.)

4.4. Participants

Evaluating interfaces designed for expert users requires access to individuals with domain expertise. This presents a challenge because expert users may be so familiar with existing tools, and so set in their ways, that they may not be interested in new interfaces [**22**]. Students, people who have formal training as audio engineers, and serious hobbyists should provide less biased results. I recruited participants from three sources: students in the Audio Arts and Acoustics program at Columbia College ($N = 11$), students in the Sound Arts and Industries graduate program at Northwestern University ($N = 3$), and people employed by professional audio equipment maker Shure ($N = 10$).

Twenty-four individuals participated, which is four in each of the counterbalanced stimulus order conditions. They ranged from 19 to 37 years old ($\mu = 27.1$, $\sigma = 6.00$), and were predominantly male (21 male, 2 female, 1 non-binary). This is in line with the audio industry is a whole, in which women are only 5% of the workforce [**5**]. Participants from Shure tended to be older ($M_{\text{Shure}} = 32.7$) than those at either university ($M_{\text{Colum}} = 22.8$, $M_{\text{NU}} = 24.0$), which is expected of a professional environment as opposed to a university.

All 11 Columbia students had taken their introductory audio production course, and 10 had taken the next is the audio production sequence, DAW Production Techniques. Only one had take the most advanced course, Advanced Practicum in Music Design.

All three Northwestern students were in or had completed the Master's in Sound Arts and Industries program.

Participants indicated which of several popular DAWs they had used, and how comfortable they were with each. In terms of use, all participants reported having used at least two DAWs (median 3). For comfort, most indicated that they were "somewhat comfortable" (4/5) with at least one DAW. (Each participant rated their comfort with a number of DAWs. The median rating on their highest-rated DAW was 4/5, and the mean was 3.75/5.) This means that participants were quite comfortable using DAW software.

4.5. Measures

I measured several types of outcomes: *overall task success*, *process-level behavioral data*, *questionnaire-based assessments of experience*, and a *qualitative interview about participants' experience*. I also measured process-oriented variables to try to understand how and why my designs work (or fail to work) as expected.

4.5.1. Task Success

A participant's instructions were to make each track as distinctly audible as possible throughout — in other words, to maximize transparency by minimizing frequency masking. Therefore, task success can be measured by how much masking is present in a finished mix versus its initial, unmixed state. Masking is calculated using the masked-to-unmasked ratio. As described in Section 3.3.1, when the ratio of a track's masked-to-unmasked ratio (Equation 3.5), which is its partial loudness (loudness in the presence of other sounds) to its loudness in isolation, is below 10% at time t, that track is considered masked at time t [**6**] (Equation 3.6).

Each track is analyzed in 50 ms frames. Before being mixed, a frame is either unmasked (masked-to-unmasked ratio above 10%), masked (masked-to-unmasked ratio below 10%), or silent (unmasked loudness is below the threshold of hearing, 0.003 sones). Similarly, after being mixed a frame is either masked, unmasked, or silent. All measures described here ignore frames that are considered silent when unmixed. This is acceptable because frames that became audible that were silent before mixing account for a total of 36 frames across all 72 mixes (each song was made up of 7,520 frames), which is 0.007% of all frames.

Equation 4.1 shows the proportion of unmasked frames, U. This can be a useful measure on its own, but is not comparable across songs because each song has different initial (unmasked) frame counts and different numbers of non-silent frames.

$$U = \frac{N_{\text{unmasked}}}{N_{\text{unmasked}} + N_{\text{masked}} + N_{\text{silent}}} \tag{4.1}$$

Comparing the proportion of unmasked frames from before a song is mixed to after is a measure of improvement, I (Equation 4.2):

$$I = \frac{U_{\text{mixed}}}{U_{\text{unmixed}}} \tag{4.2}$$

Event	Description
Play state change	User changed between stopped, paused, and playing
Playhead move	User moved the playhead (cursor) in the DAW
Session change	User changed the mix. This means moving a fader, changing a track's mute or solo state, and changing automation.
Pairwise	User made a time selection for the pairwise loudness visualization
Pairwise close	User closed the pairwise loudness visualization

Table 4.3. Logged behavioral events

This measure allows for comparison across songs. Both of these values — the proportion of unmasked frames, U, and the relative improvement in U, called I, are useful for addressing RQ 1 because they operationalize the notion of improved mix transparency.

Using the same perceptual models to drive MaskerAid and to test it limits what claims I am able to make. I can say whether or not people were able to correct issues detected by the models, and I can say whether people corrected more detected issues with MaskerAid, but I cannot say whether people corrected more issues overall or produced more subjectively transparent mixes. This is an acceptable trade-off because I am not setting out to validate the perceptual models.

4.5.2. Process Measures

To gain further insight into how and when my design concepts are used and when they are effective, I collected process-level data. This is mostly data about the actions participants took as they mixed, and the resulting state of the mix. The system logged timestamps and details for the events listed in Table 4.3.

(1) It was easy for me to achieve an acceptable mix
(2) I feel confident that frequency masking is not a major problem with this mix
(3) I was able to create a mix where every track was clear
(4) The tools I just used to mix distracted from my critical listening
(5) I felt rushed to finish the mix in the allotted time

Figure 4.2. Statements participants were asked to agree with after each trial

These events can be used to compare aggregates across trial conditions and to explore when during a song and when during a trial certain events occur. This can help reveal how much participants are changing and exploring a mix in different experimental conditions, which can help get at notions of effort and efficiency. The session change event also logs the state of a session (all levels on all tracks), so it is possible to reconstruct all loudness and changes to masking/transparency made along the way.

4.5.3. Questionnaire-Based Feedback

After each mix, participants were asked to respond to a series of statements about their experience. The statements get at participants' confidence, effort, self-efficacy, and distraction (List 4.2). They were rated on a seven-point scale, from strongly agree to strongly disagree.

I used the NASA Task Load Index (TLX) after each trial to measure different aspects of effort (mental demand, physical demand, temporal demand, performance, effort, and frustration). These six factors are reliable, with Cronbach's α of 0.80 among my participants, so I will be using the mean of all six in later analysis.

After the trial with MaskerAid (trial 2), I asked questions specific to the task and the technology being studied. These questions asked whether the visualizations were informative, the extent to which there was a clear relationship between what was shown

(1) There was a clear relationship between what was visualized on screen and what the audio sounded like
(2) The visualizations helped me make mixing decisions
(3) The visualizations distracted me from mixing

Figure 4.3. Statements participants were asked to agree with after using MaskerAid

on screen and what the audio sounded like, and whether they were helpful or distracting (List 4.3). As with the more general statements, these were rated on a seven-point scale, from strongly disagree to strongly agree.

After all trials, I administered two standard scales to assess the overall experience of using MaskerAid. To measure usability, I used the System Usability Scale (SUS) [**12**]. This scale is ten items long, and captures overall user experience. The SUS has been empirically validated and found to have a single factor and be highly reliable, with a Cronbach's α between 0.85 and 0.911 [**8**]. Among my respondents, Cronbach's α was 0.702. This is lower than expected based on the published validation, and may be a result of the concept of usability not being as applicable to a system that, in its current form, is more visualization than interactive system. The questions not applying to the MaskerAid, a visualization system, could result in a lack of construct coherence.

I am introducing a new technology into an existing work environment. Therefore, I measured how well participants felt it would be accepted using the Technology Acceptance Model (TAM) [**18**]. The TAM asks people how they feel their work will be affected by the introduction of new technology. It measures two factors, usefulness and ease of use. Davis found Cronbach's α for these two factors to be 0.97 and 0.91, respectively. Among my respondents, usefulness had a Cronbach's α of 0.95, and ease of use had an α of 0.835.

4.6. Analysis and results

The two main questions I set out to answer are whether MaskerAid enables an engineer to more easily create more transparent mixes by fixing frequency masking (RQ1), and whether MaskerAid helps improve critical listening skills for frequency masking (RQ2). The analysis and results here find that the answer to RQ1 is a qualified yes, but that there is little evidence for RQ2. To accommodate the large amount and variety of data, the results below are presented under four conceptual headings: mix quality, time and effort, learning, and usability. The first two of these speak to RQ1, learning speaks to RQ2, and usability considers MaskerAid overall. Analysis and results are summarized in Table 4.4.

4.6.1. Analytical approach

The two research questions drove the A-B-A study design. Participants did not see MaskerAid in the pre and post trials, so when comparing MaskerAid to non-MaskerAid (A versus B) I treated the two trials as a combined "A" trial. Only when the pre and post trials needed to be separate, as in rank analysis and the RQ2 investigation, did I treat them that way.

All statistical models described below include three types of control variables. First, individuals bring their own skills, abilities, and approaches to mixing. Therefore, the models include the participant as a random effect. Second, participants have different levels of experience. To account for this, I added three control variables to the models that function as an overall experience measure. First was the sum of each participant's self-reported familiarity with several concepts relevant for a frequency masking–related

Concept	Hypothesis	Operationalization	Results
Mix quality	Mixes made with MaskerAid will be more transparent (i.e., have less frequency masking) than those made without MaskerAid.	Unmasked improvement as a function of trial, comparing MaskerAid to no-MaskerAid (combined pre and post)	NS, with improvement in the expected direction. $t = 1.69, p = 0.098$
		Proportion of unmasked frames as a function of trial, comparing MaskerAid to no-MaskerAid	NS, with proportion in the expected direction. $F = 3.75, p = 0.059$
		Rank test checking for MaskerAid being better than pre	MaskerAid is significantly more likely to perform better than the pre trial. $Z = 1.97, p = 0.0491$
Time and effort	Mixing with MaskerAid will take less time than mixing without MaskerAid	Duration as a function of trial, comparing MaskerAid to no-MaskerAid	Hypothesis rejected: MaskerAid is significantly longer than other trials. Probably novelty, consistent with qualitative data. $t = -3.50, p = 0.0011$
	Users will make fewer manipulatios to arrive at their final mixes using MaskerAid than without	Number of session changes as a function of trial, comparing MaskerAid to no-MaskerAid	NS. $t = 0.864, p = 0.393$
Learning	Mixes made without MaskerAid after using MaskerAid will be more transparent than those made before using MaskerAid	Unmasked improvement as a function of trial, comparing pre and post	NS. $t = 0.204, p = 0.839$
		Rank test checking for post being better than pre	NS. $Z = 1.76, p = 0.239$
	Mixes made without MaskerAid after using MaskerAid will take less time than those made before using MaskerAid	Duration as a function of trial, comparing pre to post	NS. $t = 1.71, p = 0.094$
Usability	N/A	Self-reported distraction as a function of trial, comparing MaskerAid to no-MaskerAid	MaskerAid is significantly more distracting. Not surprising when adding new visual stimuli on top of existing tools. $t = 2.12, p = 0.040$

Table 4.4. Summary of selected analyses and results

	Sum of concept familiarity	How long have been mixing	Comfort with REAPER
Sum of concept familiarity	1.0	-0.284	-0.0953
How long have been mixing	-0.284	1.0	-0.127
Comfort with REAPER	-0.0953	-0.127	1.0

Table 4.5. Correlation matrix of control variables

task (see Appendix B). Next was participants' rating of how long they had been mixing. And finally, I included participants' rating of how comfortable they believe they are (or would be) using REAPER. These three experience variables were not correlated, as seen in Table 4.5, so the model did not suffer from issues of collinearity. The experience measure did not have any material impact on the results, but I kept it in the models for completeness. Third, each song is unique, so I always included the 'song' variable in the models.

4.6.2. Mix quality

Research question 1 asks whether MaskerAid enables engineers to more easily fix frequency masking. Because I designed MaskerAid with an eye toward HCI theory and user-centered design with the goal of reducing semantic distance, I hypothesized that it would. Stated more specifically, mixes made with MaskerAid should have less frequency masking than those made without MaskerAid. To test this, I used the unmasked improvement measure, I, described by Equation 4.2.

The distribution of improvement across all 72 trials (24 participants × 3 trials each) is shown in Figure 4.4. Two important things to point out are that it is pretty well normally distributed ($\mu = 1.10, \sigma = 0.101$), and that most trials resulted in a value of $I > 1$. I being greater than 1 means that most trials ended with more unmasked frames than were

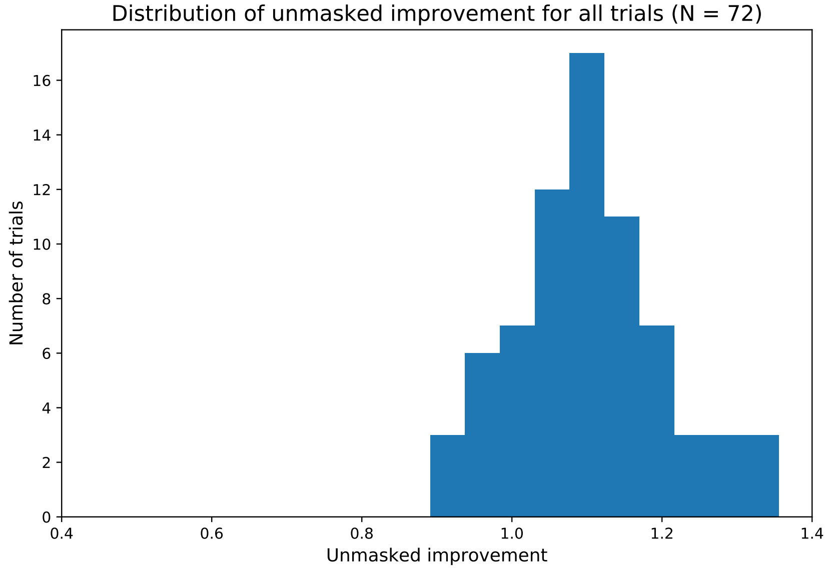


Figure 4.4. Distribution of unmasked improvement, I, for all trials

present in the unmixed songs. In other words, people successfully reduced masking, which is expected given the instruction to make each track as audible as possible.

To assess whether people reduced masking more with MaskerAid than without, I broke it out by trial (Figure 4.5). Participants did increase the number of unmasked frames using MaskerAid ($\mu = 1.13 : 1, \sigma = 0.108$) more than without ($1.09 : 1, \sigma = 0.095$ for pre-MaskerAid, and $1.09 : 1, \sigma = 0.099$ for post-MaskerAid trials). While this difference is marginally statistically detectable in this sample (see below), it is not practically meaningless: the improvement in unmasked frames with MaskerAid is 44% higher than without when compared to the no-change baseline of 1.

I tested this using an OLS regression model with I as the dependent variable, trial (pre, MaskerAid, and post) as the independent variable, and the controls described above

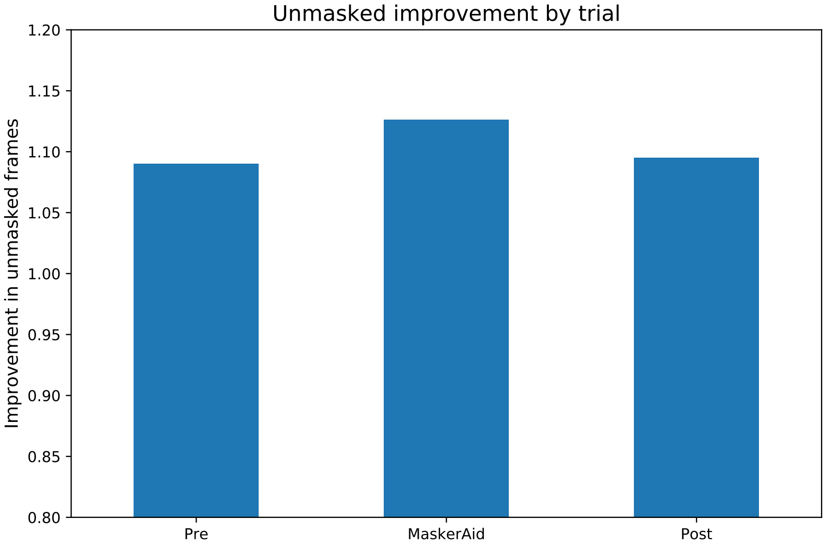


Figure 4.5. Unmasked improvement, I, by trial

(song, participant experience measures, and a random effect for participant). There was no effect of trial overall ($F(2) = 1.45$, $p = 0.246$), meaning that in no trial did unmasked improvement stand out from other trials. Using this type of omnibus analysis to compare across all three trials is not particularly revealing. This is because two of the conditions — pre and post — are the same as far as this research question is concerned, and depending on whether there is learning taking place (RQ2), there many not be much difference between them at all. This research question asked specifically about differences between trials with MaskerAid and without. Therefore, it is more appropriate to look at the contrast between MaskerAid and pre and post combined. Doing this, I find that there was greater improvement with MaskerAid than without with $I_{\text{MaskerAid}} - I_{\text{pre+post}} = 0.04$

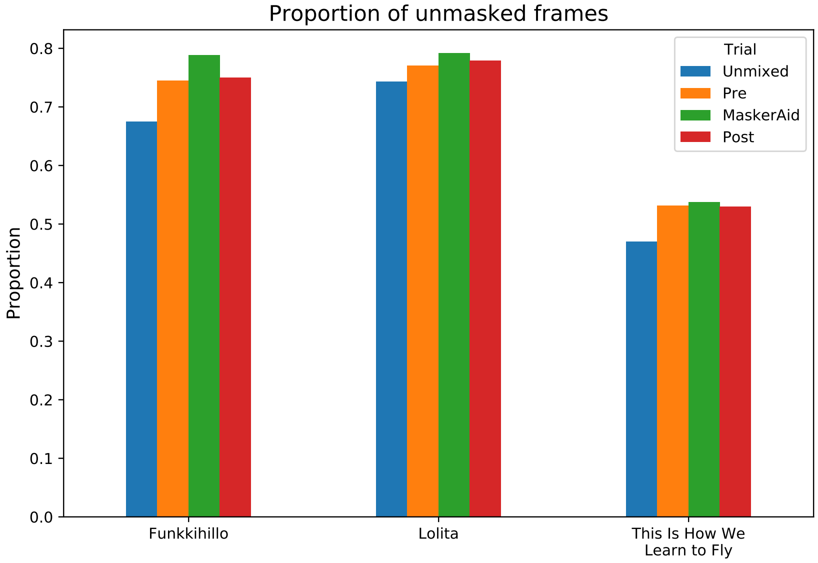


Figure 4.6. The proportion of frames that were unmasked, broken down by trial and song

$(t = 1.69, p = 0.0982)$. Alone, this is not significant, but when combined with other evidence will show a consistent trend.

The amount of masking in a mix naturally differs by song because each song is made up of different tracks with different instrumentation at different levels. Looking at the proportion of frames that are unmasked in each trial and song (Figure 4.6), there is clear evidence of the trends described above: each of the three trials is higher than the unmixed version of each song, meaning that participants successfully increased the proportion of unmasked frames by mixing. Further, for all songs, participants — on average — achieved the greatest proportion of unmasked frames using MaskerAid, with a result that is very nearly significant $(F = 3.75, p = 0.0594$ for a contrast between MaskerAid and both non-MaskerAid trials).

The differences between the mean value of unmasked improvement, I, across trials may have been too small to detect given my sample size, but the trends do suggest that mixes made with MaskerAid are consistently better than those made without. Given the possibility of learning effects, I expect that for a typical participant, MaskerAid will be their best trial, followed by post, with pre having the least improvement. To test this, I used a rank test [**28, 69**]. With higher ranks being better, pre had a median rank of 1.5, MaskerAid had a median rank of 2.5, and post was 2, which is the ordering hypotheses for RQ1 and RQ2 would predict. The difference between the pre and MaskerAid trials is statistically significant ($Z = 1.97, p = 0.0491$), indicating that there is significant evidence that MaskerAid generally results in greater improvement than is achieved in the pre-MaskerAid trial.

Overall, it does seem that MaskerAid helps people create mixes with less masking. Although the statistical effect was marginal with a regression approach, a ranked nonparametric analysis did find evidence that MaskerAid helps people perform better. The regression approach may not have shown an effect because of the relatively small sample of 24 participants, which provides lower power to detect smaller effect sizes. On the other hand, the nonparametric rank analysis is less impacted by individual differences, making it more robust against small samples.

4.6.3. Time and effort

In addition to enabling people to create mixes with less masking, RQ1 also asks about the effort required to achieve such a mix. For the same reasons I hypothesized it would reduce frequency masking — namely, that it is designed to reduce semantic distance —

I hypothesize that MaskerAid should reduce the amount of effort needed to mix. The notion of “effort” can be operationalized in many ways. In this section, I consider it largely in terms of time. MaskerAid is designed to show loudness and masking across time, so people should be able to identify and fix areas with problematic masking more quickly. Therefore, I propose the specific hypothesis that trials using MaskerAid have lower duration than those without MaskerAid.

Looking at the distribution of duration in seconds for each trial (Figure 4.7), there are a couple of things to notice. One is that the non-MaskerAid trials (pre and post) appear to be bimodally distributed, with a seemingly normal distribution near the faster end, with a bump toward the longer end. This is expected because anyone who did not finish on their own within 15 minutes was cut off. MaskerAid still has the bump toward the long end, but has fewer trials taking less time. In other words, it seems as if people are spending as much time as possible with MaskerAid.

Looking at the mean duration for each trial (Figure 4.8), MaskerAid (769 seconds) is indeed longer than the other two (692 seconds for pre; 631 seconds for post), contrary to what was hypothesized. Testing this statistically, I do find a significant effect of trial on duration ($F(2) = 7.61, p = 0.0015$), but again, the important comparison is the contrast between the MaskerAid trial and the two trials without it, which is also highly significant ($t = -3.50, p = 0.0011$).

Based on the data at hand, I can roundly reject the hypothesis that MaskerAid enabled people to mix more quickly. While this is certainly the case in the context of this experiment, novelty effects may have played an important role in this result.

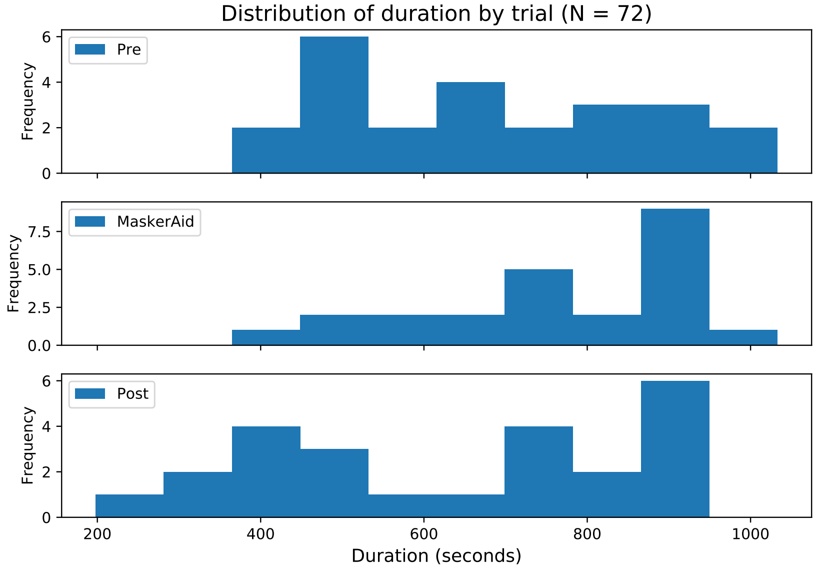


Figure 4.7. The distribution of trial duration (in seconds) broken out by trial

In addition to duration, I also operationalized the notion of effort as the number of changes made to a session over the course of a trial (Figure 4.9), and the density of changes (the number of changes per unit duration) in a trial (Figure 4.10). I found no effect of trial on number of changes, either overall ($F(2) = 1.66, p = 0.202$) or between MaskerAid and non-MaskerAid trials ($t = 0.864, p = 0.393$). I did find an effect of trial on change density ($F(2) = 11.4, p = .0001$), but that is driven by the post trial being so much shorter than the other two, thereby driving density up. Neither of these analyses give any insight into whether MaskerAid helped people mix more efficiently.

Another way of thinking about time and effort is that people may improve the quality of a mix (i.e., reduce the amount of masking) more quickly using MaskerAid even if they spend more time using it overall. As people mix, each change to the session should be

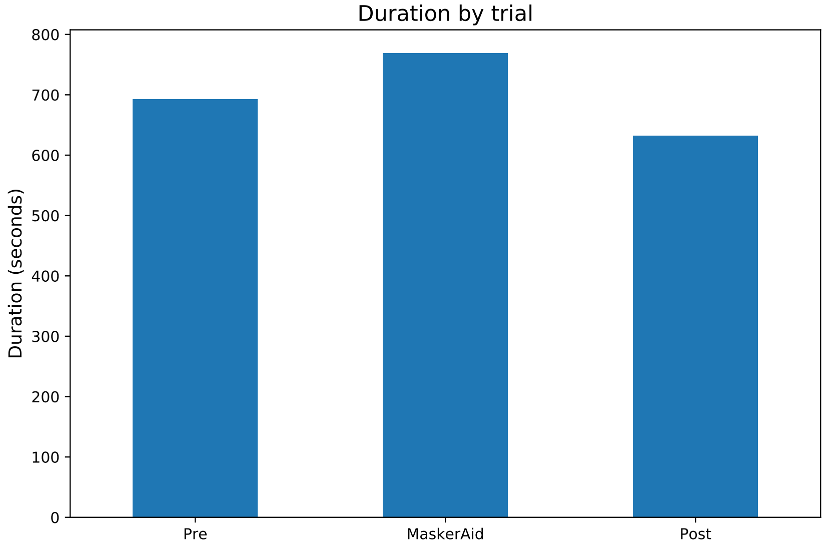


Figure 4.8. The mean duration of each trial. Participants spent more time with MaskerAid than in the other two conditions, contrary to expectations.

in service of arriving at a better mix. If this is true, the proportion of unmasked frames should generally increase with each change until a mix reaches the greatest proportion of unmasked frames that it will reach. Practically speaking, someone is likely to make bigger changes toward the beginning of their process, resolving more significant masking issues. Then, after some time, someone is likely to reach a point of diminishing returns where the marginal reduction in masking is minimal. In other words, it seems logical to suggest that the proportion of unmasked frames in a session will start out increasing quickly, then asymptotically approach its maximum value. I hypothesize that the proportion of unmasked frames will start leveling off near its maximum more quickly with MaskerAid than without.

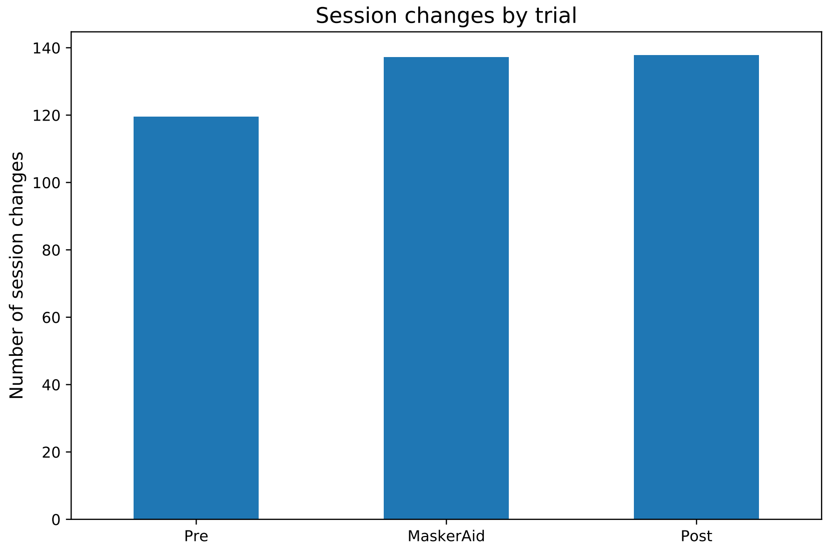


Figure 4.9. The mean number of changes to a session, by trial

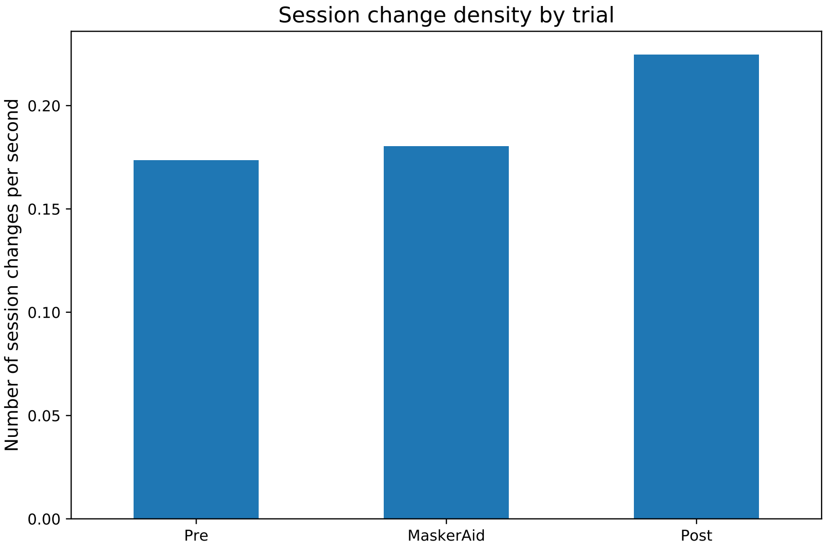
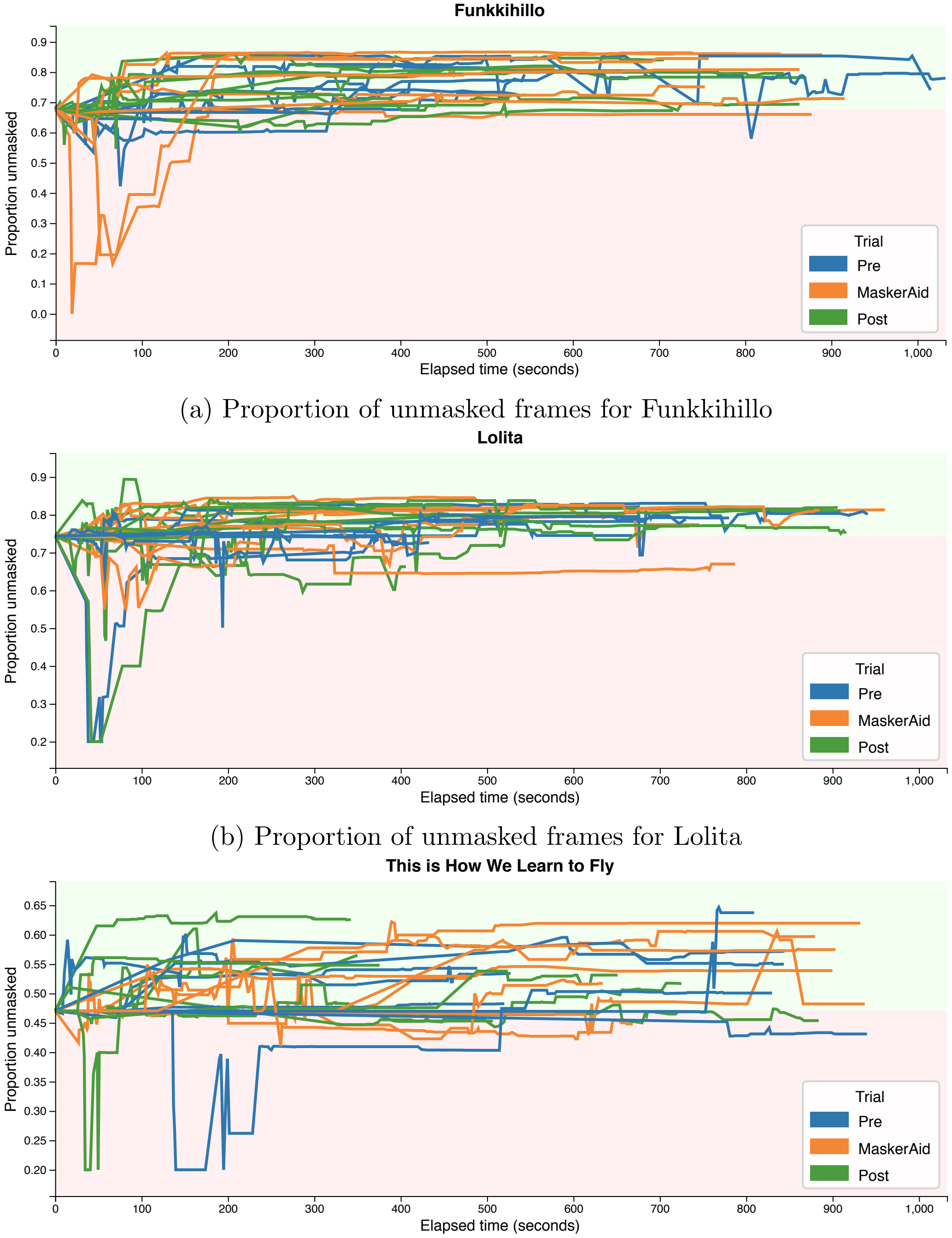


Figure 4.10. The mean number of changes to a session per second, by trial



(a) Proportion of unmasked frames for Funkkihillo

(b) Proportion of unmasked frames for Lolita

(c) Proportion of unmasked frames for This is How We Learn to Fly

Figure 4.11. Proportion of unmasked frames over time. Each line represents a trial, colored by pre, MaskerAid, or post. The background shading shows the initial proportion of unmasked frames.

Figure 4.11 shows the proportion of unmasked frames over time. Each line represents a individual trial, and is colored by whether it is pre, MaskerAid or post. Songs are shown separately because each one has different initial levels of masking, and different within-song range of masking. Each data point is a change a participant made to a mix, excluding those where the session included muted or soloed tracks. Mixing does have a generally positive impact on mix transparency, because most trials end with a greater proportion of unmasked frames than they started with. (This is a more process-level look at what is shown in Figure 4.6.) It does not look like the hypothesis is borne out, though there is a lot of noise in the data stemming from individual differences between participants' approaches.

Most notable among these individual differences are two individuals who, in every trial, decrease the proportion of masked frames substantially toward the beginning of their mixing process before bringing it back up to a level more comparable to most other participants. These two participants, the log data shows, took a rather different approach to mixing: they first brought the faders for all tracks down to zero, then increased them one by one, arriving at a rough mix additively rather than by working with the whole from the start. For people who use such an approach, MaskerAid would not be useful until after all tracks are brought back in to make a rough mix.

Overall, it does not appear from this experiment that MaskerAid reduced the time or effort needed to mix. I will discuss why this may be in the Discussion section (4.7).

4.6.4. Learning

Research question two asks whether MaskerAid helps improve critical listening skills for frequency masking. By making frequency masking visually salient, MaskerAid can reinforce the concept, possibly leading to an increased ability to detect it. I can address this question using the same operationalization for improvement as in Section 4.6.2. Specifically, this question involves comparing the pre-MaskerAid trial directly to the post-MaskerAid trial, while ignoring the MaskerAid trial. Looking at Figure 4.5 and comparing the two relevant trials, it seems that the answer is no, there is no difference between them. Testing this using the model described in Section 4.6.2 but taking the contrast between the pre and post trials confirms this lack of detectable effect: $t = 0.204, p = 0.839$. In terms of rank, the post trial did typically beat pre, with median ranks of 1.5 and 2, respectively. However, the difference in rank between pre and post is similarly not significant ($Z = 1.76, p = 0.239$), indicating that post did not perform better than pre consistently enough to be statistically detectable.

Another way to operationalize RQ2's concept of learning is in conjunction with efficiency. The post trial was shorter than the pre trial with marginal significance ($t = 1.71, p = 0.094$). Whether this trend is a result of learning and practice is not clear, as it may also have been caused by fatigue and boredom setting in after mixing two other songs and having been in the study for over an hour. Figure 4.11 once again proves to be too noisy to make any claims, but if learning were an important factor in driving the shorter post trial duration, the lines for post would improve at a faster rate than those for pre.

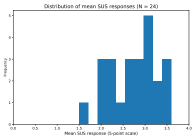
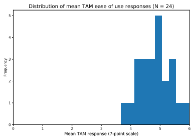
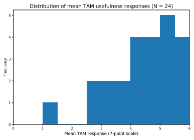
Scale	Distribution	Mean	Median	σ
SUS		2.75/4	2.8/4	0.516
TAM ease of use		4.83/6	4.83/6	0.570
TAM usefulness		4.33/6	4.67/6	1.18

Table 4.6. Distributions and summary statistics of usability scale results

4.6.5. Usability

I collected quantitative user feedback using the System Usability Scale (SUS) and Technology Acceptance Model (TAM) scales once per participant, asking about their experience with MaskerAid. Therefore, it is not possible to directly measure a causal effect of the design on perceived user experience. That said, a one-sided, absolute assessment finds that users found MaskerAid useful and easy to understand. The results summarized in Table 4.6 show that people responded positively to MaskerAid. For both the SUS and TAM, responses were consistently greater than the neutral midpoint of their five- and seven-point scales (notated as 0–4 and 0–6), respectively.

Introducing a new tool into an existing work environment can be challenging. Responses to the SUS ($\mu = 2.75/4, \sigma = 0.516$) and TAM ease-of-use subscale ($\mu = 4.83/6, \sigma = 0.570$) indicate that usability was generally perceived well, with values comfortably greater than the neutral midpoints of the scale. As a new system presenting a new type of data (perceived loudness), this may also be suppressed by novelty effects. The TAM usefulness subscale tells a more interesting story. While the mean is slightly lower than that for ease of use, there is quite a bit more disagreement between participants on how useful it is ($\mu = 4.33/6, \sigma = 1.18$). One participant felt it was "quite unlikely" (1/6) they would use MaskerAid in their work, but another rated it "extremely likely" (6/6), and eight other participants rated it between "quite likely" and "extremely likely" (5–6/6). So, thought it was not a unanimous opinion, there was consensus that MaskerAid can be very useful.

Other questionnaire-based feedback is more mixed. Participants rated their agreement with five statements for each trial (Figure 4.12, List 4.2). Most of the observed differences are not statistically significant (unless stated otherwise), but there are some interesting trends. MaskerAid seems to have left people feeling more confident that frequency masking is not a problem in their final mixes, which is good news for the design of MaskerAid. However, people reported feeling more rushed when using MaskerAid, which backs up the result that found users taking longer in the MaskerAid trial. Users also reported that MaskerAid distracted from critical listening more than traditional mixing tools ($t = 2.12, p = 0.040$), which makes sense because MaskerAid adds to the tools used in the pre and post trials.

In addition to statements about each trial, I also ask about agreement with statements specifically about MaskerAid (Figure 4.13, List 4.3). This research did not set out to validate Glasberg and Moore's perceptual models, but participants tended to "agree" or "somewhat agree" that there was a clear relationship between what they saw and what they heard. They also "somewhat agreed" that MaskerAid provided actionable information that helped them make mixing decisions. And while participants did feel

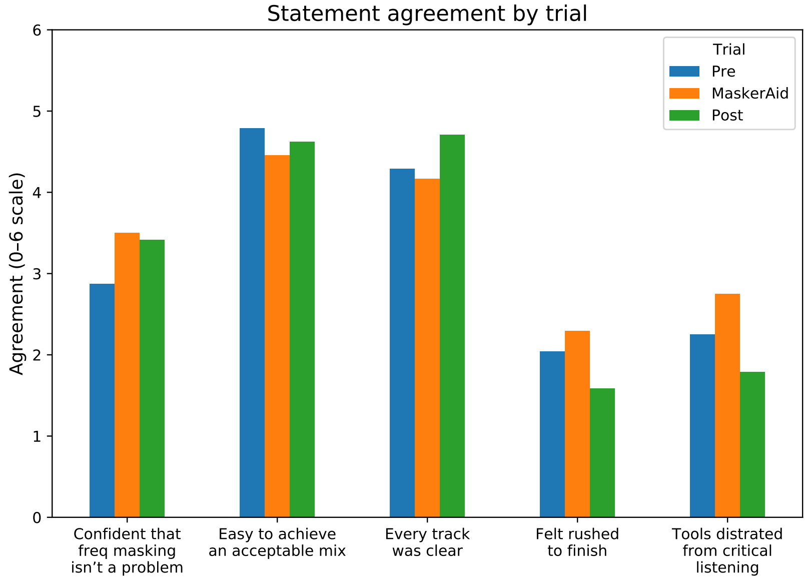


Figure 4.12. Self-reported agreement with each statement, by trial. 7-point scale ("strongly disagree"–"strongly agree").

that MaskerAid was more distracting than traditional mixing tools alone, in an absolute sense they were quite neutral on the subject ("neither agree nor disagree").

Finally, I administered the NASA Task Load Index (TLX) after each trial. Based on the results presented above, there are no surprises in those results (Figure 4.14). Once again, there are no statistically significant results here, but the trends are consistent with other results: as a new tool no one had ever used before, MaskerAid tended to have slightly higher task load than traditional tools, but not significantly so.

4.7. Discussion

This study set out to answer two research questions: Does MaskerAid enable users to achieve more transparent mixes with less effort? And, Does MaskerAid help improve

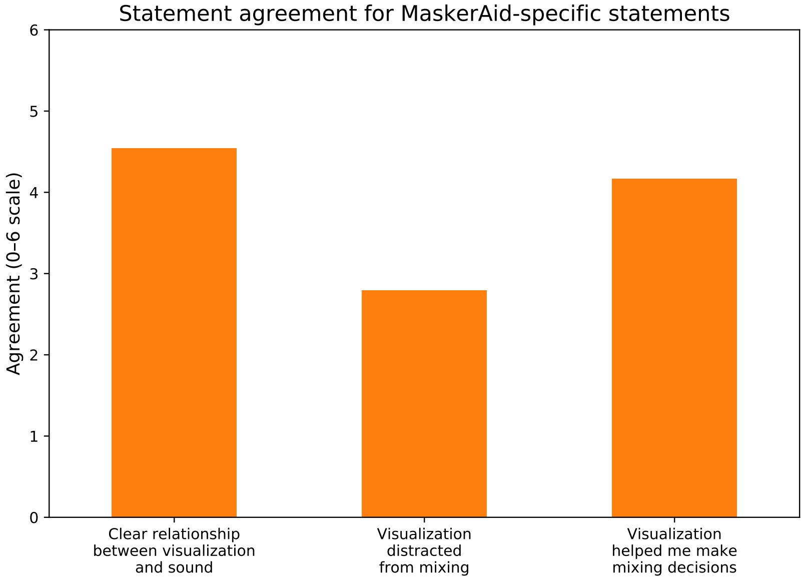


Figure 4.13. Self-reported agreement with each statement about MaskerAid. 7-point scale (“strongly disagree”–“strongly agree”).

critical listening skills for frequency masking? The answer to RQ1 is a qualified ‘yes’, at least for the mix transparency half of the question. When using MaskerAid, participants did generate more transparent mixes and may have felt more confident that their mixes were not plagued by frequency masking. However, the reduction in masking was only marginally statistically significant in the regression analysis, and the increased confidence was not significant at all. The results of the rank analysis also provide support for MaskerAid’s impact on mix transparency, with participants statistically significantly more likely to perform better with MaskerAid than in the pre trial. Despite the weakness of the results of the regression analysis, there is evidence that the increase in mix transparency is real, including the rank analysis and that the effect is consistent across songs (Figure 4.6). If the effect were the result of an interaction between song and MaskerAid,

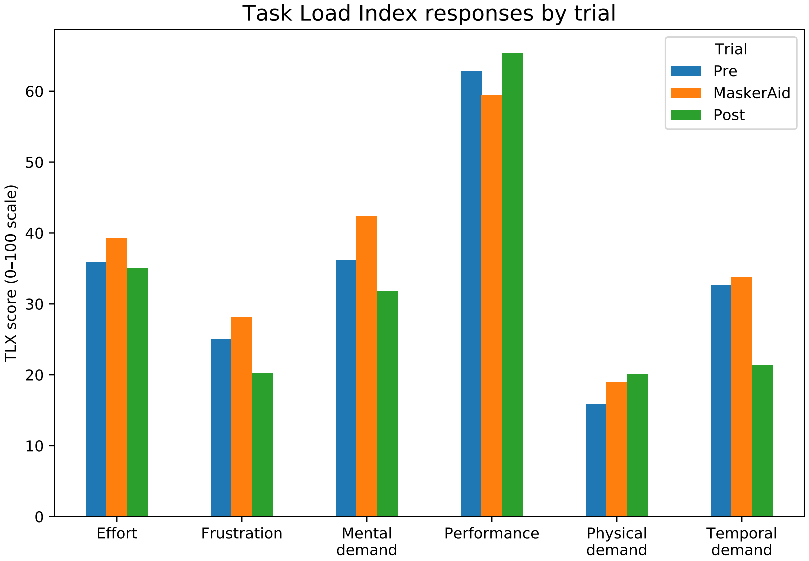


Figure 4.14. Responses to the NASA TLX, by trial

that would discredit the notion that MaskerAid itself is causing the overall increase. I also want to note the fairly small sample size for this study, limited by the difficulty of finding participants with appropriate expertise. Despite this small sample size, all evidence points in the direction of increased mix transparency with MaskerAid.

This increased mix transparency also comes in a context where, despite the design intent of MaskerAid, it was not a given that participants would perform better than the baseline unmixed levels of transparency. For example, MaskerAid could have driven down performance by being distracting and increasing cognitive load, or by the prototype putting the visualization adjacent to rather than inside the DAW. Despite these challenges, the MaskerAid trial was, on average, the trial that resulted in the most transparent mixes.

Despite the neutral questionnaire response to the question about MaskerAid being distracting, this was an issue that participants commented on:

> “I worry if I left it open all the time that I would sort of just like be looking at it and not listening as much. So I'd probably use it. . . yeah, like trouble spots. And probably like once I was pretty happy with what I was hearing, maybe like turn it on, listen through, and see, does it suggest that there are issues?”
> –Participant s_xFhWkZM

> “It's a slippery slope because you want to use it as an asset, not as a crutch, and I think it would take me an appreciable amount of time to see where that lands.”
> –Participant s_5AB6vsS

Implicit in the operationalization of mix transparency used in this study is the assumption that all instances of masking are equally problematic. In a more naturalistic environment where art and aesthetics influence mixing decisions, this would not be the case; some instances of masking are likely to be less desirable than others. One participant commented,

> “But even when I leveled his level up to the point where artistically I was happy with how it sounded, it was still showing up as red, you know? So it was like, well, I addressed it and I think it sounds good, but the system still thinks—
> So it's almost like I want a way to, like, clear it out. Like, I worked on

		Mixed		
		Masked	Unmasked	Silent
Unmixed	Masked	Uncorrected	Corrected	Uncorrected
	Unmasked	Harmed	Unharmed	Harmed
	Silent	NA	NA	NA

Table 4.7. Frame classifications based on masked state before and after being mixed

> that, you know?"
>
> –Participant s_e1n7PML

More granular ways of thinking about masking in a mix can be helpful in accounting for this. One way to approach this is by looking at which instances of masking were fixed. More specifically, depending on whether it was masked or unmasked before being mixed, a frame in the final mix can get one of four classifications: corrected (masked → unmasked), uncorrected (masked → masked), harmed (unmasked → masked), and unharmed (unmasked → unmasked) (Table 4.7).

A future analysis could then consider which frames tend to be be corrected or left uncorrected, and under what conditions. For example, frames that are more consistently corrected without using MaskerAid may be considered more artistically important than those that tend to be corrected only with MaskerAid. While this type of analysis would be straightforward for a given song, trying to generalize it across songs would be much more challenging because the factors that determine artistic importance are unique to each song.

Unlike the results of the task success analyses, the results of the time and effort analysis found that mixing with MaskerAid took longer and required more effort than mixing without it. Participants took more time with MaskerAid, and did not make fewer changes to their sessions. Self-reported effort based on questionnaire data also did not show reduced effort with MaskerAid, and often pointed in the direction of MaskerAid taking more effort. In discussing this and feeling more confident about their mixes, a participant said,

> "I don't know if it would necessarily make me mix quicker, but I think I would have a little more confidence and it might point out some things that I didn't realize. So it might make my mixes better, but I don't know about quicker."
>
> –Participant s_bW5CLHD

On the other hand, some felt that it could help get to a rough mix more directly:

> "If I can get to something that's pretty solid right away, then I can spend more time on details."
>
> –Participant s_SYhluTE

The relatively brief exposure participants had to MaskerAid may be partially responsible for the longer trial result because people may have been interested in MaskerAid and wanted to spend more time with it. In other words, novelty may have played a role in the MaskerAid trial being longer than the others. Participants said as much:

> "If you've lived with it for a while, I think that that would become a tool in your toolbox rather than the shiny new thing."
>
> –Participant s_5AB6vsS

> "I'd love to play with something like this more."
>
> –Participant s_xFhWkZM

The tendency for the post-MaskerAid trial to be shortest also suggests one of two things. One is that participants got quicker at identifying and fixing frequency masking, possibly thanks to MaskerAid (RQ2). The other is that some combination of fatigue and boredom may have set in. After all, the experiment took about 90 minutes, and by the time people started mixing the third song, they had already been going for over an hour and had mixed two other songs.

A reason the results of this study may have been weaker than theory predicted is that the design of MaskerAid itself does not accomplish the design goal of reducing semantic distance to the extent I intended. Despite the user research in Chapter 2, it is possible that loudness over time and masking are not what would reduce the semantic distance between what DAWs show today and the information needed to accomplish the user goal of mixing. That said, some participants did appreciate being able to see loudness over time:

> “The visual part is nice because you can’t listen to the whole song at once, but you can *see* the whole song at once. So I don’t have to like sit there and go and listen to the verse for 10 seconds, then jump to the chorus and jump back and jump back and go, ‘Yeah, okay, it’s louder in the verse. I want to change that.’ I can just look and quickly go, ‘Oh, it’s louder in the verse.’ And then that would save a good amount of time.”
>
> –Participant s_bW5CLHD

It could also be that my decision to focus the design on the time domain as opposed to the frequency domain was misguided. MaskerAid does capture some important effects of frequency through the loudness models, but does not center the frequency domain.

Finally, I want to comment on RQ2, which asks whether exposure to MaskerAid can help participants learn to recognize frequency masking better. This experiment failed to find evidence for this, at least when learning is narrowly defined as increased mix transparency in the post-MaskerAid mix compared to the pre-MaskerAid mix. On the other hand, the post trial was marginally shorter than the pre trial while maintaining the same degree of improvement. The shorter duration may have been, as mentioned before, the result of learning, or may have been the result of fatigue and boredom. Given that improvement in transparency remained at the same level as in pre but was accomplished in less time, this could be evidence for learning. Yet, when looking at improvement over time (Figure 4.11) it seems that people did not improve faster; rather, they declared themselves done earlier instead of fiddling around with small tweaks. Much like RQ1, RQ2 was likely impacted by novelty effects and the short exposure to MaskerAid. A long-term deployment in an educational setting could provide more insight into the trends described here.

At least one participant felt that it could be a useful tool for ear training for frequency masking:

> “I think there’s an argument to be made for that as an educational tool, where you can really accurately show somebody like, ‘Hey, this is what you think is happening, but you don’t realize that it isn’t your snare

> drum fighting against it, it's your overheads."'
>
> –Participant s_5AB6vsS

4.7.1. Theory

MaskerAid was intended to reduce semantic distance between what is presented to users and what they need to mix successfully. Rather than making audio engineers mentally translate between pre-fader waveforms and post-processing loudness, MaskerAid does it for them. This is something that participants appreciated. When asked about how they use waveforms when mixing, many talked about using them for navigation, consistent with Chapter 2. One added,

> "Yeah, I don't. Like, what a waste of my visual—
>
> Like I'm sitting there looking. Give me something that helps me! Don't
>
> just give me garbage."
>
> –Participant s_bW5CLHD

Waveforms make up a large portion of the visual information that DAWs present. When waveforms' usefulness is limited to navigation, that means a huge portion of the visual field is taken up by minimally useful (i.e., semantically distant) information. This is in contrast to the loudness curves from MaskerAid, which provide information that is more directly applicable to the mixing process.

Hutchins, Hollan, and Norman point out that system design can reduce semantic distance, as was the goal here, but so can user adaptation. In section 2.4.1 I observed that these expert users learn mappings from physical properties like amplitude and frequency to perceptual phenomena like loudness and timbre. If expert users have adapted to existing designs, existing designs may have *functionally* shorter semantic distance even if distance as a result of the design itself is arguably greater. As Dostál demonstrated in the case of Microsoft Office introducing the ribbon UI, changing an interface on experts, even to something more usable by novices, is disruptive [**22**]. Given more time, these users may have been able to better integrate the more semantically direct information presented by MaskerAid into their work processes.

This study also revealed the challenge of trying to directly measure the concept of semantic distance. The measures described in Section 4.5 are intended to be proxies for semantic distance, but may not have actually measured it. That does not mean that those measures failed to capture anything useful; just that they may not have captured the concept I was trying to change with my design. As a framework, direct manipulation provides design guidelines and a valuable analytics frame, but does not directly testable claims, which makes evaluation challenging.

4.7.2. Limitations

MaskerAid was designed to reduce semantic distance, so this study was, theoretically speaking, designed to measure the concept of semantic distance. This study had both methodological and ecological validity limitations in its ability to measure this theoretical construct. One limitation is that participants experienced extremely short exposure to MaskerAid: an average of 12 minutes and 49 seconds. Putting a new tool in front of expert users and requiring them to use it on a complex task when they have no experience with that tool is a tall order. Not only is it more effort, but people may actually want to spend more time with it in order to learn more about it and what it can do for them. With no more than 15 minutes to use MaskerAid, it makes sense that novelty could play a significant role in how people use and think about the tool. With more time to experience MaskerAid, it is possible that these novelty effects would have worn off and people would figure out how to best integrate it into their regular workflows. A longer-term deployment would help address these issues.

Another limitation to this study is its fairly small sample size, which was because of the need for participants with domain-specific knowledge. I am not aware of any previous work that quantitatively assesses masking and changes in masking within a mix. Therefore, I did not have a sense of what the difference in means across conditions would be, nor did I have a sense of the standard deviation. Given the observed values, a power analysis for a power of 0.8 would require a sample size on the order of 100 participants. Mixing is often a subtle art, so it makes sense that effect sizes would be small. I find it encouraging that despite this subtlety, the observed trends and findings from both the rank test and the regression analysis, in addition to average counts and other descriptive analyses, point to the improvement in mix transparency with MaskerAid being a real effect despite not having the statistical power to say so with certainty.

There are also ecological validity concerns that are worth noting. Limitations of the prototype, discussed in detail in Section 3.6.3, artificially constrained how people mix. Most notable was the requirement that participants mix with level only, because MaskerAid was only able to account for changes in level, not the effects of other parameters like panning, or plugins like EQ and compression.

> "Being constrained was challenging. I'm not used to working that way."
>
> –Participant s_xFhWkZM

The instruction to make each track audible also limited the natural creative expression that is a common part of mixing practices. Performance was also a challenge for the prototype. The design intent was for MaskerAid to update fluidly as a user changes mix parameters, but in practice it took about two seconds to update after each change. As a prototype, MaskerAid also sat adjacent to a DAW rather than being directly embedded within the software audio engineers are used to working in. This meant looking back and forth between two windows, which may have added to participants' cognitive load.

> "If it was all in one, it would probably be easier for me to just accept if it was like: here's the thing you want to know, then I would accept it. But I felt like the extra work of, like, looking at the waveform and looking at the faders and then looking over here was just like one too many like cognitive steps."
> –Participant s_hFkE2Kw

There is also the question of how well the perceptual models worked in this application. As noted in the results, this study was not designed to validate Glasberg and Moore's models, but the models did perform quite well based on questionnaire results: participants tended to agree (5) that there was a clear relationship between what was on screen and what they heard (median $= 5/6, \mu = 4.54/6, \sigma = 1.20$).

Future work should refine the implementation of MaskerAid to allow for more flexible and natural mixing behavior. Future studies should also consider longer-term deployments with a larger number of people.

Overall, despite the relatively weak statistical results, people did tend to like MaskerAid, complimenting ease of use and the overall concept.

“I don’t have to sit and look and go, ‘Okay, what’s this axis? What’s this? Let me open the manual. What should I be seeing?’ It’s just like, okay, it’s just, ‘This is loud or not. And things are being masked.’ Easy. Got it. I’m ready to go.”

–Participant s_bW5CLHD

“So the tool there I found really useful with balancing the two guitar tracks. ’Cause I found that kind of challenging. I wasn’t sure how to make them both, like, kind of similarly audible. And the tool made that really easy ’cause it’s just like, Okay, if I kind of balance the levels, I could just use my eyes and not my ears at all, and then I could switch back to my ears like, Okay, that seems like it sounds pretty balanced.”

–Participant s_xFhWkZM

“That’s f***in’ rad, man. I liked that so much.”

–Participant s_SYhluTE

CHAPTER 5

Conclusion

This dissertation has shown that computational models of human perception can be a valuable design tool when visually representing nonvisual artifacts. This is because nonvisual artifacts need to be transformed into visual artifacts in order to enable display and manipulation, especially with direct manipulation interfaces. Such a transformation, or manipulable inter-medium encoding, should result in a representation with a meaning and physical form that is relevant and useful for people's goals. I found that simple manipulable inter-medium encodings sometimes fall short of the goal of being semantically useful. In other words, there may be semantic distance between the meaning of a visual representation and what is useful. When the semantic distance is the result of people's perceptual experience of the nonvisual artifact (like sound) differing from the meaning of the output of a naïve inter-medium encoding, there is opportunity for improvement. The main idea put forward here is that designers can reduce semantic distance by using manipulable inter-medium encodings that are backed by computational models of human perception to show more perceptually relevant visualizations.

The research and design of MaskerAid is a case study in this proposition. There are three main contributions.

Chapter 2 **demonstrates the importance of choosing the right manipulable inter-medium encoding when representing nonvisual artifacts**. I found that visualizations of audio tend to use simple encodings that rely only on the physical properties of the sound — frequency and amplitude — but that these fail to encode perceptually important information like loudness and masking. This mismatch between what comes from these simple encodings and what people want to know is a source of semantic distance.

Having established the problem of visualizations that are not as useful as they could be because they lack perceptually relevant information, I set out to create a design solution. Chapter 3 is **an application of computational models of human perception to create manipulable inter-medium encodings that are designed to be more relevant and meaningful to people**. This design solution, called MaskerAid, uses computational models of human auditory perception to drive visualizations of multitrack audio. MaskerAid replaces a traditional waveform display that shows sound pressure with a display that instead shows perceived loudness. In determining loudness, it takes into account the amplitude and frequency content of each track of audio, the current state of a track in a mix (i.e., its level), and the loudness of other tracks that are playing at the same time. MaskerAid is also able to show the extent to which tracks mask each other over a given time slice. For audio engineers, the meaning that is important when mixing includes interactions between amplitude, frequency content, and loudness. Using traditional interfaces, such meaning has to be inferred in people's heads based on previous knowledge about how certain instruments or frequency bands behave perceptually. By showing this meaning more directly, MaskerAid advances the state of the art of audio mixing tools, and should reduce semantic distance between what people want to know and what is shown on screen.

Chapter 4 **demonstrated that using computational models in manipulable inter-medium encodings can meaningfully improve performance toward a goal**.

The finding was not statistically strong, but, as discussed in the chapter, the difference in performance is practically meaningful. This means that MaskerAid achieved its practical goal of increasing mix transparency by helping users identify frequency masking. As discussed above, the theoretical objective of the design was to reduce semantic distance, but it is more difficult to say whether it achieved this aim. The improvement in task performance is likely because of reduced semantic distance, but the Hutchins, Hollan, and Norman model of direct manipulation does not provide much insight into measuring semantic distance.

On the theoretical side, in the discussion of their framework for direct manipulation (Figure 5.1), Hutchins, Hollan, and Normal acknowledge that the feeling of directness is relative, and that cognitive effort is inversely proportional to the feeling of directness, but offer no insight into how to directly measure the sensation of directness. Much of classic HCI that these authors were working with at the time had measurement tools (e.g., Fitts's Law [**25**], GOMS [**14**]), but there is no way to directly measure the gulf of evaluation or semantic distance. Instead, I operationalized the concept of semantic distance by making the assumption that task performance is proportional to directness, and that increased directness is a result of reduced semantic distance. However, the theory does not offer a way to say with certainty that what I measured was semantic distance.

To further exacerbate the measurement challenge, recall that expert users can adapt to a system, and that when they do, they are effectively spanning the gulf of evaluation. It would be useful for the theory to provide guidance on how to distinguish distance arising from the design of the system from distance arising from the user's understanding of a representation.

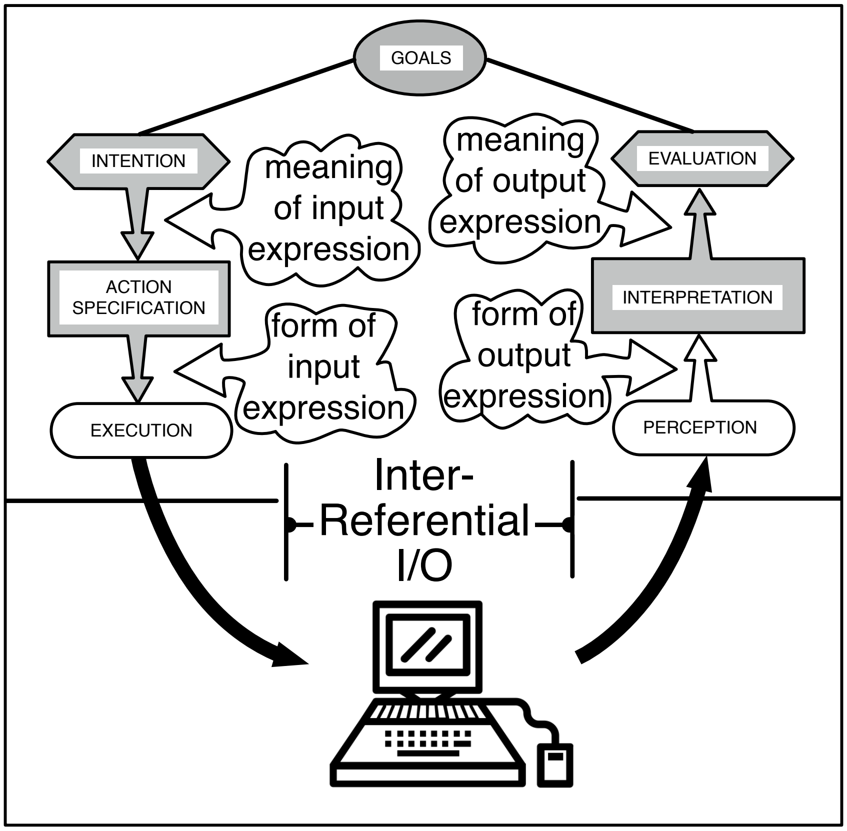


Figure 5.1. The Hutchins, Hollan, and Norman model of direct manipulation (reproduced from [**34**])

Another assumption that has been implicit throughout this dissertation is that reducing the gulf of evaluation should always be a designer's goal. I would argue that what a system displays should always be as close to the meaning a user is looking for, and that this holds true for simple consumer products and tools for professionals. The difference between these two audiences is that tools for professionals may need more representations because expert users are likely to examine multiple aspects of an artifact or system, and they may need representations that are closer to physical reality because expert users are more likely to care about physical reality (in addition to just perceived reality). Just because expert users are better able to bridge any semantic distance from representations of physical reality through learning and adaptation does not mean they should be required to do so, so offering more perceptual views in addition has few downsides.

Systems for experts are often considered difficult to use because they are “powerful”. This points to a tension between ease of use and expressiveness: the more flexible a tool is, the harder it is to use because there are more degrees of freedom. Hutchins, Hollan, and Norman describe this trade-off with the example of a piano and a violin, where playing a note on a piano is more direct (just press the appropriate button), but playing the same note on a violin involves pressing exactly the right place on a string and skillfully drawing the bow across it. The method of playing on the piano is more direct and generally considered easier, but the method of playing on the violin offers more opportunity for expressiveness through techniques such as vibrato and sliding.

An important distinction between ease of use versus expressiveness, and semantic distance in representations is that the notion of expressiveness applies strictly to the execution (user input) side of the direct manipulation model, whereas the work in this dissertation applies only to the evaluation (system output) side of the model. Because these are different, designers can design systems that are as expressive or easy to use as they want while still minimizing semantic distance on the output side. This is the approach I took when I designed MaskerAid for professionals, minimizing semantic distance by more directly showing loudness, but still leaving the repair of masking issues entirely to users, who can adjust any number of levers (EQ, level, etc.) to address issues.

Hutchins, Hollan, and Norman also provide many examples of interfaces that, in their estimation, have greater or lesser degrees of directness. Many of these examples have nonvisual output, like tones, beeps, or even musical notes, but none is about operating on or manipulating something that is not readily visualizable. Direct manipulation interfaces tend to use a model–world metaphor in which the interface is some sort of manipulable model of the world being operated on. This is not how sound works, though. For sound, modeling the sonic world has traditionally meant representations of signals (e.g., amplitude and frequency plots), signal flow (e.g., routing matrices), and occasionally spatial layouts, but that is not the sound itself. For mixed-sense interfaces such as visual interfaces for interacting with audio, it is clear that the manipulable inter-medium encoding is a very important component of the model.

The model in Figure 5.1 is a cognitive explanation for direct manipulation; the model exists entirely within the user — outside the computer, and above the horizontal line. Yet design is vitally important, and is entirely within the computer. Visualization researchers and practitioners have long known that good design is influenced by human perception. For example, differences in length are easier to detect than differences in area [**17**]. Knowing this can help designers reduce articulatory distance by choosing an appropriate physical form for a visualization. For this simple example, there is no need for sophisticated computational models of human perception. In the case of loudness and frequency masking, though, I have demonstrated the valuable role models of perception can play in determining not just how to represent something, but *what* to represent — its meaning.

As a general approach, this focus on the meaning of what is being represented, and the use of perceptual models to infer that meaning, can be summarized in three steps:

(1) Identify the *meaning* that is useful for a given user goal.
(2) Extract that meaning from the original artifact using computational models of perception for the original sense of the artifact.
(3) Represent that meaning in a useful way in the destination sense.

I have applied this approach to visualizing partial loudness and masking of audio, but it is not limited to this narrow application. Consider audio production tools for blind users, where instead of converting sound to visual, the visual display is converted to another sense, usually sound. Screen readers are typically the tool available to such users, but screen readers read all content on a screen, making it difficult to quickly find specific information. Saha et al. found that screen readers for DAWs can be particularly difficult to use because information has to be repeated for each track, making it hard to find information about a specific track, or a specific parameter across a number of tracks [**52**]. Two examples of this from their work are finding which tracks are muted and which have experienced clipping. For sighted users, finding this information is easy because it is made visually salient by being shown in red. Rather than using models of auditory perception to extract meaning to be represented visually, models of visual perception could determine which visual elements would be most salient to a sighted person (e.g., red ones) and read them first. The destination sense of an inter-medium encoding does not need to be sound or vision. Previous work [**37**] represents EQ information tactilely. One could use a similar approach but with a focus on perceived loudness rather than EQ.

Beyond just audio production, computational models of perception are able infer many types of meaning that can be relevant for people. In addition to loudness, models of auditory perception can, for example, identify concepts such as timbre [**53**] and isolate multiple sound sources into separate streams of audio [**41**]. Models like these can be used to describe an auditory scene visually, in text, or tactilely. Models of visual perception can infer depth from monocular cues, material, glossiness, and other semantically meaningful aspects of the visual environment (e.g., [**38, 68**]). It may be possible to design a system that conveys such information using sound, manual tactile feedback, or even tactile feedback on the tongue [**7**] more efficiently or accurately than current attempts (e.g., [**55**]) by focusing on more perceptually relevant information. Further afield, more sophisticated understanding of perception could perhaps even enable designed synesthetic experiences.

When interviewed for my user research in Chapter 2, P10 quoted a colleague, describing the role of audio engineers as, "the bridge between the objective — the technology — and the subjective — the art." The same is true of system designers, and of user interfaces themselves. In a visual interface focused on nonvisual artifacts, the bridge between objective and subjective — the underlying properties of what is being represented and how it is represented — is the manipulable inter-medium encoding. What the manipulable inter-medium encoding is bridging is what Hutchins et al. call the gulf of evaluation. In this thesis, I have shown that computational models of human perception can be an important building block of this bridge, particularly by helping to reduce semantic distance, bringing what is being displayed closer to the meaning people are looking for.

In closing, I have argued that many visualizations of nonvisual artifacts, like sound, inadequately represent the perceptual experience and meaning of those artifacts. User research, like that in Chapter 2, can demonstrate how these representations fall short for certain tasks, and can reveal what meaning people would find more useful for those tasks. By using computational models of human perception, it is possible to more directly represent that meaning, as I demonstrated with MaskerAid in Chapter 3. In addition to MaskerAid itself, Chapter 4 contributes initial evidence that a design approach that uses computational models of perception can result in user interfaces that enable improved user performance. Throughout this dissertation, I have introduced the use of computational models of perception as a design tool for the creation of more intelligent user interfaces, enabling designers to more effectively bridge the objective and the subjective.

References

[1] 2015. FabFilter Pro-Q 2. (June 2015). http://www.fabfilter.com/products/pro-q-2-equalizer-plug-in

[2] 2015. Har-Bal 3.0. (June 2015). http://www.har-bal.com/

[3] 2015. Mixing Console History. (June 2015). http://mixingconsole.org/web/main/history/

[4] 2019a. Neutron 3 Masking Meter. (2019). https://www.izotope.com/en/products/neutron/features/masking-meter.html

[5] 2019b. SoundGirls – F.A.Q.s. (2019). https://soundgirls.org/about-us/soundgirls-org-f-a-q-s/

[6] Philipp Aichinger, Alois Sontacchi, and Berit Schneider-Stickler. 2011. Describing the Transparency of Mixdowns: The Masked-to-Unmasked-Ratio. In *Audio Engineering Society Convention 130*. http://www.aes.org/e-lib/browse.cfm?elib=15811

[7] Paul Bach-y Rita and Kurt A Kaczmarek. 2002. Tongue placed tactile output device. US Patent and Trademark Office. (2002).

[8] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction* 24, 6 (July 2008), 574–594. DOI:http://dx.doi.org/10.1080/10447310802205776

[9] Alan F Blackwell. 2006. The reification of metaphor as a design tool. *Transactions on Computer-Human Interaction (TOCHI* 13, 4 (2006). http://dl.acm.org/citation.cfm?id=1188816.1188820&coll=DL&dl=GUIDE&CFID=511595042&CFTOKEN=65311444

[10] Mike Bostock. 2011. D3: Data-Driven Documents. (2011), BSD. http://d3js.org/

[11] ITU Radiocommunication Bureau BR. 2015. *Recommendation ITU-R BS.1770-4. Algorithms to measure audio programme loudness and true-peak audio level.* Technical Report BS.1770-4. http://search.itu.int/Pages/SearchResults.aspx?k=ALL(loudness)

[12] John Brooke. 1996. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*, P W Jordan, B Thomas, B A Weerdmeester, and I L McClelland (Eds.). Taylor & Francis, London.

[13] William Buxton and Richard Sniderman. 1980. Iteration in the Design of the Human–Computer Interface. In *Proceedings of the 13th Annual Meeting, Human Factors Association of Canada.* http://www.billbuxton.com/iteration.html

[14] Stuart K Card, Thomas P Moran, and Allen Newell. 1983. *The Psychology of Human–Computer Interaction.* Lawrence Erlbaum Associates.

[15] Juan Pablo Carrascal and Sergi Jordà. 2011. Multitouch Interface for Audio Mixing. In *NIME 2011*. Oslo, Norway, 100–103. http://www.nime.org/proceedings/2011/nime2011_100.pdf

[16] Mark Cartwright, Bryan Pardo, and Joshua D Reiss. 2014. MIXPLORATION: Rethinking the Audio Mixer Interface. *the 19th international conference* (2014), 365–370. DOI:http://dx.doi.org/10.1145/2557500.2557530

[17] William S Cleveland and Robert McGill. 1984. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *J. Amer. Statist. Assoc.* 79, 387 (Sept. 1984), 531–554. DOI:http://dx.doi.org/10.2307/2277346

[18] Fred D Davis. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly* 13, 3 (Sept. 1989), 319. DOI:http://dx.doi.org/10.2307/249008

[19] Brecht De Man and Joshua D Reiss. 2013. A Knowledge-Engineered Autonomous Mixing System. In *Audio Engineering Society Convention 135*. http://www.aes.org/e-lib/browse.cfm?elib=17011

[20] Robert Dennis. 2000. Recommended Equalization Frequencies. (March 2000). http://www.recordingeq.com/Subscribe/tip/tip11258.htm

[21] Vincent Diamante. 2007. *AWOL: Control Surfaces and Visualization for Surround Creation*. Technical Report. http://www.apocalypsewow.com/random/diamante_awol_thesispaper.pdf

[22] Martin Dostál. 2010. User Acceptance of the Microsoft Ribbon User Interface. In *DNCOCO '10: Proceedings of the 9th WSEAS international conference on Data networks, communications, computers*. 143–149.

[23] Matthew Duignan, James Noble, Pippin Barr, and Robert Biddle. 2004. Metaphors for Electronic Music Production in Reason and Live. In *APCHI 2004*. Springer Berlin Heidelberg, 111–120. DOI:http://dx.doi.org/10.1007/978-3-540-27795-8_12

[24] Jean-Daniel Fekete and Catherine Plaisant. 2002. Interactive information visualization of a million items. In *InfoViz 2002*. IEEE Comput. Soc, 117–124. DOI: http://dx.doi.org/10.1109/INFVIS.2002.1173156

[25] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (June 1954), 381–391. http://sing.stanford.edu/cs303-sp11/papers/1954-Fitts.pdf

[26] Harvey Fletcher and Wilden A Munson. 1933. Loudness, its Definition, Measurement and Calculation. *The Journal of the Acoustical Society of America* (1933), 82–108.

[27] Jon Ford, Mark Cartwright, and Bryan Pardo. 2015. MixViz: A Tool to Visualize Masking in Audio Mixes. In *Audio Engineering Society Convention 139*. New York. http://music.cs.northwestern.edu/publications/FordCartwrightPardo_AES2015.pdf

[28] Milton Friedman. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Amer. Statist. Assoc.* 32, 200 (1937), 675–701. DOI:http://dx.doi.org/10.1080/01621459.1937.10503522

[29] Bill Gaver. 1989. The SonicFinder: An Interface That Uses Auditory Icons. *Human Communication Research* 4 (1989), 67–94. DOI:http://dx.doi.org/10.1207/s15327051hci0401_3

[30] Matthias Geier and Sascha Spors. 2012. Spatial Audio with the SoundScape Renderer. In *27th Tonmeistertagung – VDT International Convention*. http://spatialaudio.net/ssr/

[31] Brian R Glasberg and Brian C J Moore. 2002. A Model of Loudness Applicable to Time-Varying Sounds. *J. Audio Eng. Soc* 50, 5 (2002), 331–342. http://www.aes.org/e-lib/browse.cfm?elib=11081

[32] Kristian Gohlke, Michael Hlatky, Sebastian Heise, David Black, and Jörn Loviscach. 2010. Track Displays in DAW Software: Beyond Waveform Views. In *Audio Engineering Society Convention 128*. http://www.aes.org/e-lib/browse.cfm?elib=15441

[33] Sandra G Hart. 2006. *NASA-Task Load Index (NASA-TLX); 20 Years Later*. Technical Report. http://humansystems.arc.nasa.gov/groups/TLX/downloads/HFES_2006_Paper.pdf

[34] Edwin L Hutchins, James D Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Human Communication Research* 1 (1985), 311–338.

[35] International Electrotechnical Commission. 2013. Electroacoustics - Sound level meters - Part 1: Specifications. International Electrotechnical Commission. (2013).

[36] Roey Izhaki. 2011. *Mixing Audio* (2 ed.). Focal Press. https://www.safaribooksonline.com/library/view/mixing-audio-3rd/9781317508502/xhtml/Ch01.xhtml

[37] Aaron Karp and Bryan Pardo. 2017. HaptEQ: A Collaborative Tool For Visually Impaired Audio Producers. In *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*. Association for Computing Machinery, New York, NY, USA. DOI:http://dx.doi.org/10.1145/3123514.3123531

[38] Yunfeng Li, Zygmunt Pizlo, and Robert M Steinman. 2009. A computational model that recovers the 3D shape of an object from a single 2D retinal representation. *Vision Research* 49, 9 (May 2009), 979–991. DOI:http://dx.doi.org/10.1016/j.visres.2008.05.013

[39] Noah Liebman, Michael Nagara, Jacek Spiewla, and Erin Zolkosky. 2010. Cuebert: A New Mixing Board Concept for Musical Theatre. *NIME 2010* (2010), 1–6.

[40] Zheng Ma, Joshua D Reiss, and Dawn Black. 2014. Partial Loudness in Multitrack Mixing. In *Audio Engineering Society 53rd International Conference*. 1–9.

[41] Ethan Manilow, Prem Seetharaman, and Bryan Pardo. 2020. Simultaneous Separation and Transcription of Mixtures with Multiple Polyphonic and Percussive Instruments. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 771–775. DOI:http://dx.doi.org/10.1109/ICASSP40776.2020.9054340

[42] Stuart Mansbridge, Saoirse Finn, and Joshua D Reiss. 2012. Implementation and Evaluation of Autonomous Multi-Track Fader Control. *Audio Engineering Society Convention 132* (2012). http://www.aes.org/e-lib/browse.cfm?elib=16226

[43] Josh H McDermott and Andrew J Oxenham. 2008. Music perception, pitch, and the auditory system. *Curren Opinion in Neurobiology* 18, 4 (2008), 452–463. DOI:http://dx.doi.org/10.1016/j.conb.2008.09.005

[44] Brian C J Moore. 2012. *An Introduction to the Psychology of Hearing* (6 ed.). Emerald Group Publishing Limited, Cambridge, UK. http://www.worldcat.org/title/introduction-to-the-psychology-of-hearing/oclc/960182961

[45] Brian C J Moore, Brian R Glasberg, and Thomas Baer. 1997. A Model for the Prediction of Thresholds, Loudness, and Partial Loudness. *J. Audio Eng. Soc* 45, 4 (1997), 224–240. `http://www.aes.org/e-lib/browse.cfm?elib=10272`

[46] Joseph Mullenbach, Craig Schultz, J Edward Colgate, and Anne Marie Piper. 2014. Exploring Affective Communication Through Variable-Friction Surface Haptics. *CHI 2014* (2014).

[47] Josh Mycroft, Joshua D Reiss, and Tony Stockman. 2016. Visually Representing and Interpreting Multivariate Data for Audio Mixing. In *SMC 2016 - 13th Sound and Music Computing Conference*. 332–337.

[48] Francois Pachet and Olivier Delerue. 2000. On-the-Fly Multi-Track Mixing. In *Audio Engineering Society Convention 109*. `http://www.aes.org/e-lib/browse.cfm?elib=9083`

[49] Alan P Parkes. 1990. The implications of manipulable inter-medium encodings for coaching and learner modelling. In *IEEE Colloquium on Intelligent Tutoring Systems*. 8–1–8–3.

[50] Enrique Perez-Gonzalez and Joshua D Reiss. 2009. Automatic gain and fader control for live mixing. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*. 1–4. `DOI:http://dx.doi.org/10.1109/ASPAA.2009.5346498`

[51] E Perez-Gonzalez and Joshua D Reiss. 2011. Automatic Mixing. John Wiley & Sons, Ltd, Chichester, UK, 523–549. DOI:http://dx.doi.org/10.1002/9781119991298.ch13

[52] Abir Saha and Anne Marie Piper. 2020. Understanding Audio Production Practices of People with Vision Impairments. In *ASSETS '20: The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, New York, NY, USA, 1–13. DOI:http://dx.doi.org/10.1145/2700648.2809858

[53] Prem Seetharaman and Bryan Pardo. 2014. Reverbalize. In *the ACM International Conference*. ACM Press, New York, New York, USA, 739–740. DOI:http://dx.doi.org/10.1145/1640233.1640339

[54] Ben Shneiderman. 1983. Direct Manipulation: A Step Beyond Programming Languages. *Computer* 16, 8 (1983), 57–69. DOI:http://dx.doi.org/10.1109/MC.1983.1654471

[55] P M Silva, T N Pappas, J Atkins, and J E West. 2016. Perceiving Graphical and Pictorial Information via Hearing and Touch. *IEEE Transactions on Multimedia* 18, 12 (2016), 2432–2445.

[56] Andrew J R Simpson, Michael J Terrell, and Joshua D Reiss. 2013. A Practical Step-by-Step Guide to the Time- Varying Loudness Model of Moore, Glasberg and Baer (1997; 2002). In *Audio Engineering Society Convention 134*. 1–7.

[57] Spotify. 2020. Mastering & loudness – FAQ – Spotify for Artists . (2020). https://artists.spotify.com/faq/mastering-and-loudness

[58] Yôiti Suzuki and Hisashi Takeshima. 2004. Equal-loudness-level contours for pure tones. *The Journal of the Acoustical Society of America* 116, 2 (Aug. 2004), 918–933. DOI:http://dx.doi.org/10.1121/1.1763601

[59] Richard Swettenham. 1981. Console Desgin for the Eighties — Centralized Control of Outboard Electronics. *Recording Engineer/Producer* 12, 4 (Aug. 1981), 17–26.

[60] Richard Swettenham. 1991. Digitally Controlled Analogue Mixing 10 Years On. *Studio Sound and Broadcast Engineering* (April 1991), 29–34.

[61] Edward R Tufte. 1995. *Envisioning Information*. Graphics Press. https://books.google.com/books?id=eI5grgEACAAJ

[62] Bret Victor. 2013. Media for Thinking the Unthinkable. (2013). http://worrydream.com/#!/MediaForThinkingTheUnthinkable

[63] DeLiang Wang and Guy J Brown (Eds.). 2006. *Wiley: Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press. http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471741094.html

[64] Dominic Ward, Cham Athwal, and Munevver Kokuer. 2013. An efficient time-varying loudness model. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 1–4. DOI:http://dx.doi.org/10.1109/WASPAA.2013.6701884

[65] Dominic Ward, Joshua D Reiss, and Cham Athwal. 2012. Multi-track mixing using a model of loudness and partial loudness. In *Audio Engineering Society Convention 133*. 1–9.

[66] Matthew O Ward. 1994. XmdvTool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the Conference on Visualization '94*. IEEE Computer Society Press, 326–333. http://dl.acm.org/citation.cfm?id=951087.951146

[67] Gordon Wichern, Hannah Robertson, and Aaron S Wishnick. 2016. Quantitative Analysis of Masking in Multitrack Mixes using Loudness Loss. In *Audio Engineering Society 141st Convention*. Los Angeles, CA, USA, 1–9.

[68] Hugh R Wilson. 1995. Quantitative Models for Pattern Detection and Discrimination. In *Vision Models for Target Detection and Recognition*. 3–15. DOI:http://dx.doi.org/10.1142/9789812831200_0001

[69] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 143–146. DOI:http://dx.doi.org/10.1145/1978942.1978963

APPENDIX A

User Research Discussion Guide

A.1. Background

- How long have you been mixing?
- What kind of gear did you learn on? (We'll get to what gear you work on now in a bit.)
  - Do you think this has influenced your working style now?
- Do you have any thoughts about working on a large-format console as opposed to a DAW ("in the box")?
  - How about a large control surface connected to a DAW vs actually mixing on a board?

A.2. Current context

- What types of audio do you typically work with?
  - Music? Voice overs? Vocal versus instrumental? Acoustic/Electric? Genre?
  - How many tracks?
- How much time do you spend mixing in an average week? How about editing or recording/engineering a session?

A.3. Process

- Do you have a certain way that you think about mixing? For example, as a bottom-up process (starting with individual tracks and combining them) or a top-down process (starting with an idea of overall sound and adjusting the tracks till you achieve that)? Neither/both/it depends?
- Do you have any techniques for establishing an overall sound?
- What about tweaking? Is that different from getting an overall sound?
- What is different about getting into the details than a rough initial mix?
- Describe your mixing process in general.
  - Software
  - Hardware – surfaces, outboard processors
- In what ways does that process differ from project to project?

A.4. Specifics

- Talk about something you mixed recently.
- Can you walk me through how you mixed this track?
  - What was fun?
  - What was frustrating?
- When listening to the mix, what are you listening for?
  - Overall?
  - After making a change?
- Are there tools or techniques you particularly like? Dislike?

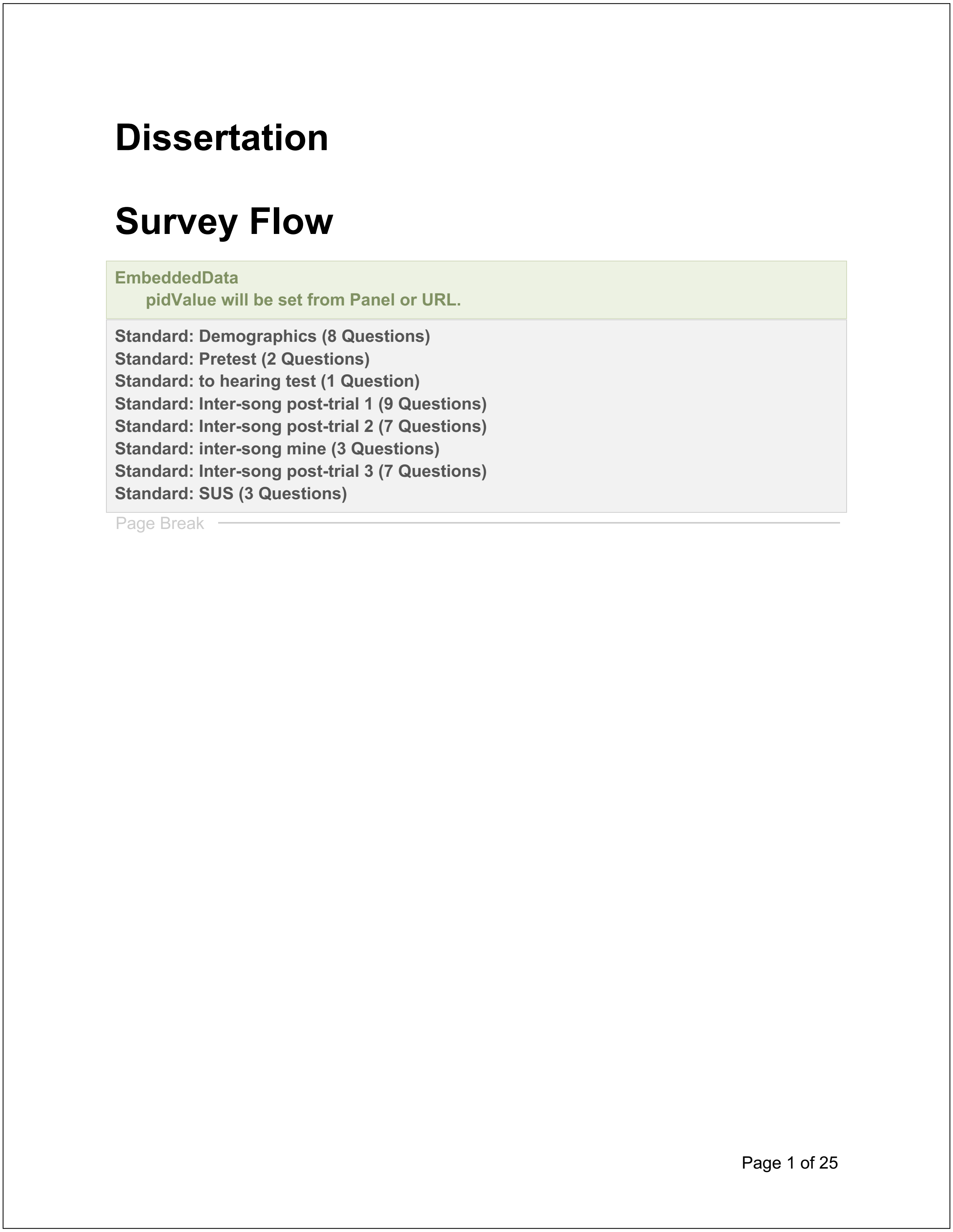
A.5. Conclusions

- What tells you when something is done?
- What issues jump out at you in a mix that tell you it's not done?
- How much do you mix on your own versus with musicians or clients or producers in the room with you? How does your process differ in each case?
- Are there ways you wish you could work, but don't have the equipment or the time to learn?

APPENDIX B

Evaluation Study Questionnaire

The questionnaire participants filled out before, between, and after completing the tasks of the evaluation study (Chapter 4) is on the following pages.



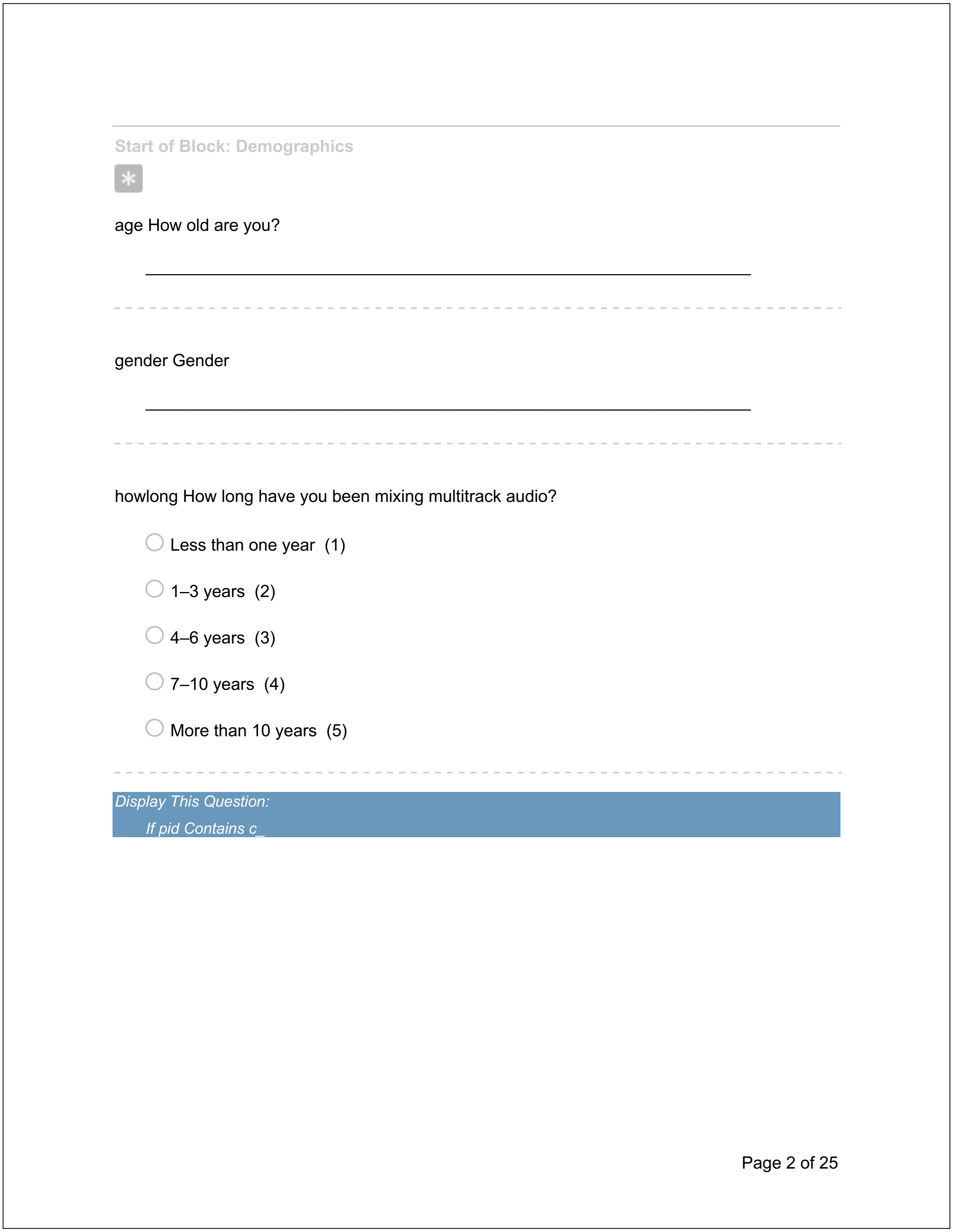
Dissertation

Survey Flow

EmbeddedData
pidValue will be set from Panel or URL.

Standard: Demographics (8 Questions)
Standard: Pretest (2 Questions)
Standard: to hearing test (1 Question)
Standard: Inter-song post-trial 1 (9 Questions)
Standard: Inter-song post-trial 2 (7 Questions)
Standard: inter-song mine (3 Questions)
Standard: Inter-song post-trial 3 (7 Questions)
Standard: SUS (3 Questions)

Page Break



Start of Block: Demographics

✱

age How old are you?

____

gender Gender

____

howlong How long have you been mixing multitrack audio?

○ Less than one year (1)

○ 1–3 years (2)

○ 4–6 years (3)

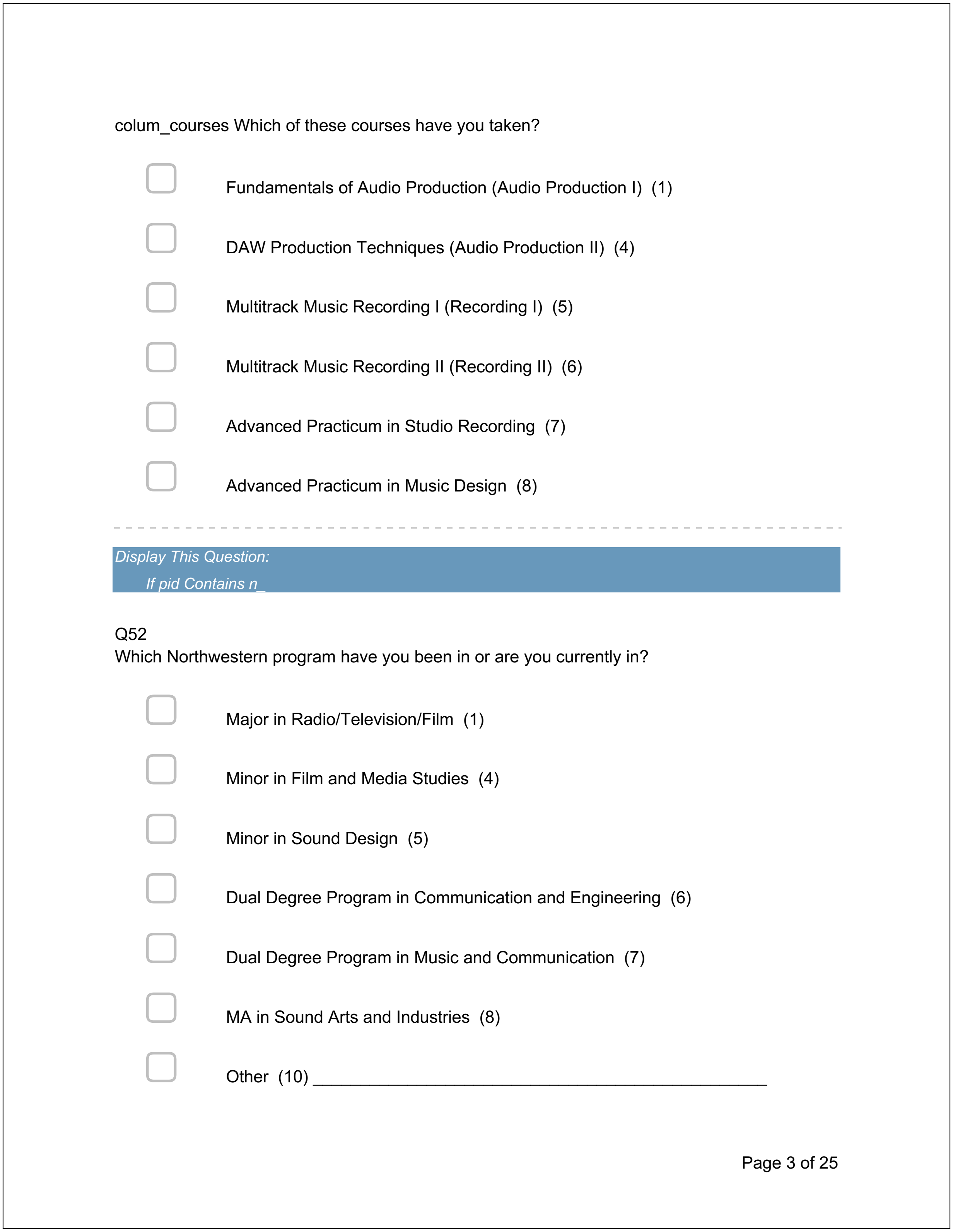
○ 7–10 years (4)

○ More than 10 years (5)

Display This Question:

*If pid Contains c_*

Page 2 of 25



colum_courses Which of these courses have you taken?

□ Fundamentals of Audio Production (Audio Production I) (1)

□ DAW Production Techniques (Audio Production II) (4)

□ Multitrack Music Recording I (Recording I) (5)

□ Multitrack Music Recording II (Recording II) (6)

□ Advanced Practicum in Studio Recording (7)

□ Advanced Practicum in Music Design (8)

Display This Question:

*If pid Contains n_*

Q52
Which Northwestern program have you been in or are you currently in?

□ Major in Radio/Television/Film (1)

□ Minor in Film and Media Studies (4)

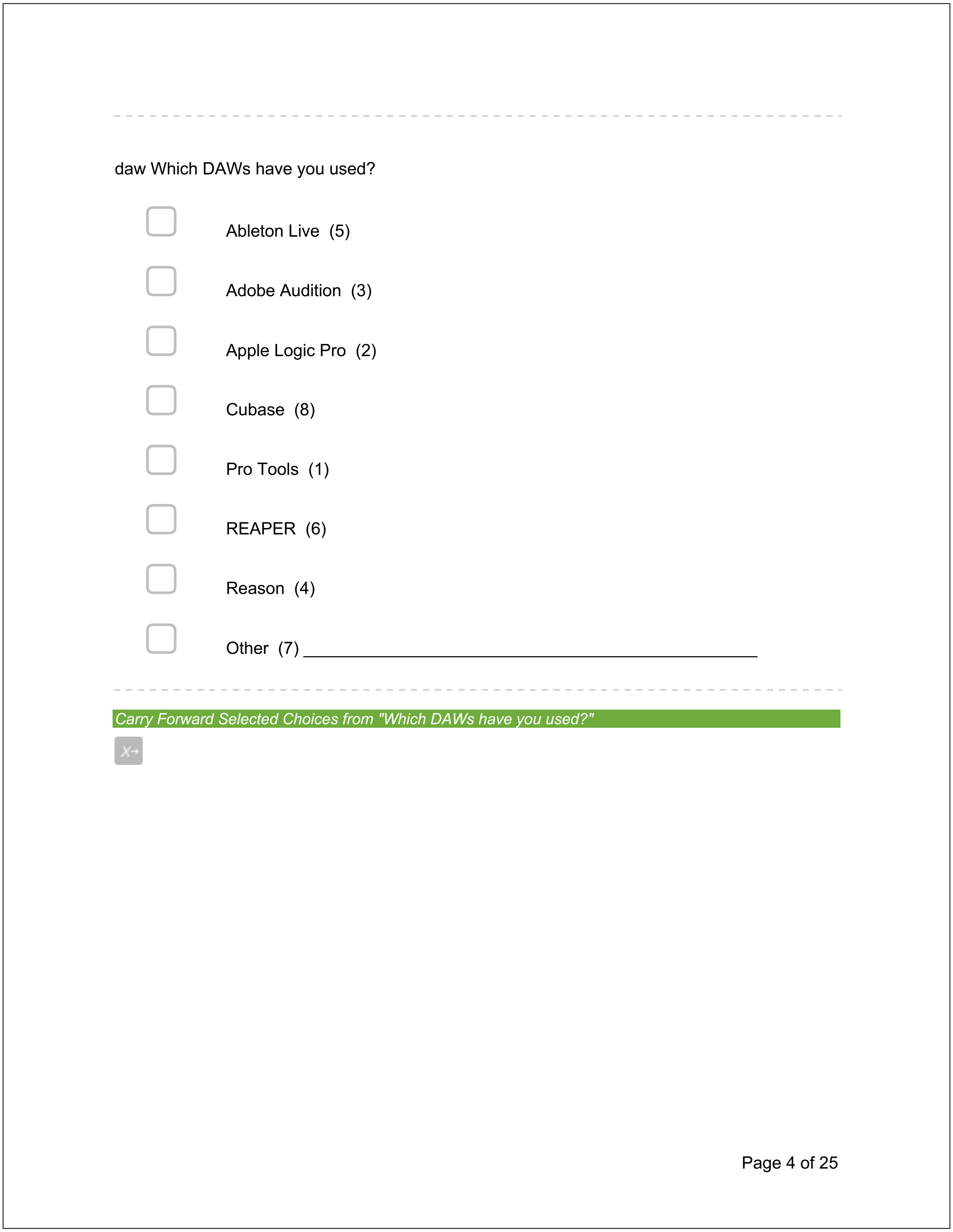
□ Minor in Sound Design (5)

□ Dual Degree Program in Communication and Engineering (6)

□ Dual Degree Program in Music and Communication (7)

□ MA in Sound Arts and Industries (8)

□ Other (10) ____



daw Which DAWs have you used?

□ Ableton Live (5)

□ Adobe Audition (3)

□ Apple Logic Pro (2)

□ Cubase (8)

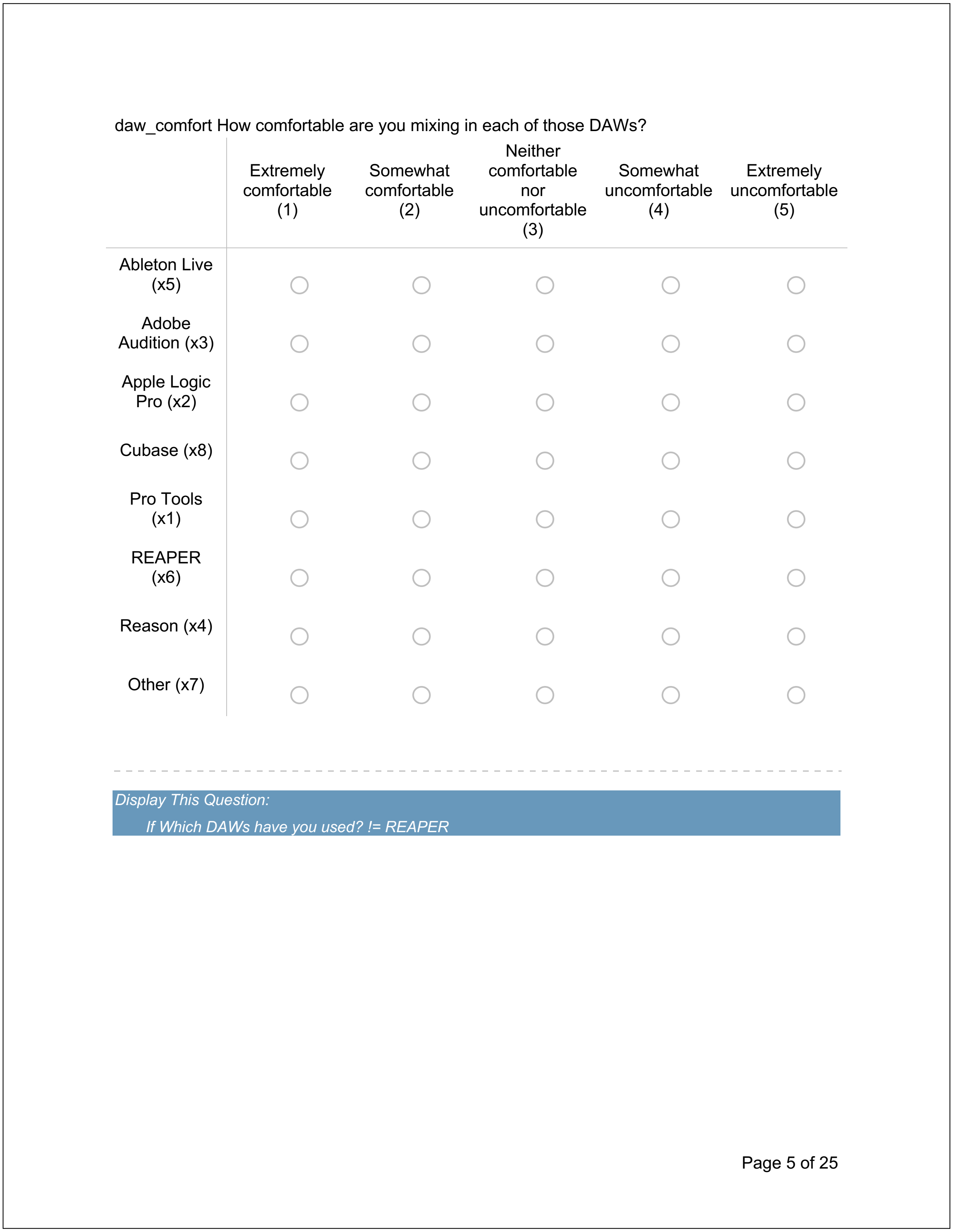
□ Pro Tools (1)

□ REAPER (6)

□ Reason (4)

□ Other (7) ____

Carry Forward Selected Choices from "Which DAWs have you used?"

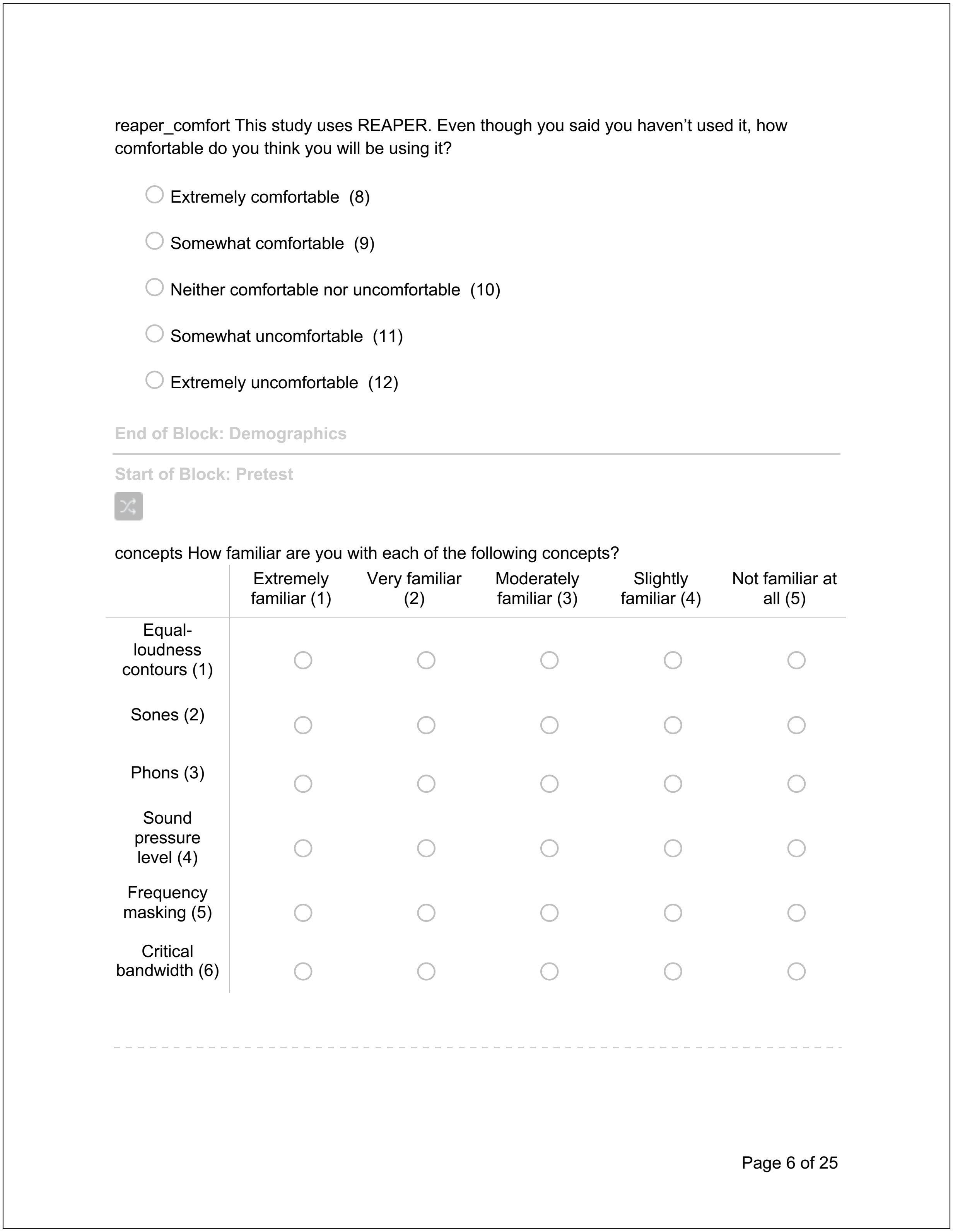


daw_comfort How comfortable are you mixing in each of those DAWs?

	Extremely comfortable (1)	Somewhat comfortable (2)	Neither comfortable nor uncomfortable (3)	Somewhat uncomfortable (4)	Extremely uncomfortable (5)
Ableton Live (x5)	○	○	○	○	○
Adobe Audition (x3)	○	○	○	○	○
Apple Logic Pro (x2)	○	○	○	○	○
Cubase (x8)	○	○	○	○	○
Pro Tools (x1)	○	○	○	○	○
REAPER (x6)	○	○	○	○	○
Reason (x4)	○	○	○	○	○
Other (x7)	○	○	○	○	○

Display This Question:

If Which DAWs have you used? != REAPER



reaper_comfort This study uses REAPER. Even though you said you haven’t used it, how comfortable do you think you will be using it?

○ Extremely comfortable (8)

○ Somewhat comfortable (9)

○ Neither comfortable nor uncomfortable (10)

○ Somewhat uncomfortable (11)

○ Extremely uncomfortable (12)

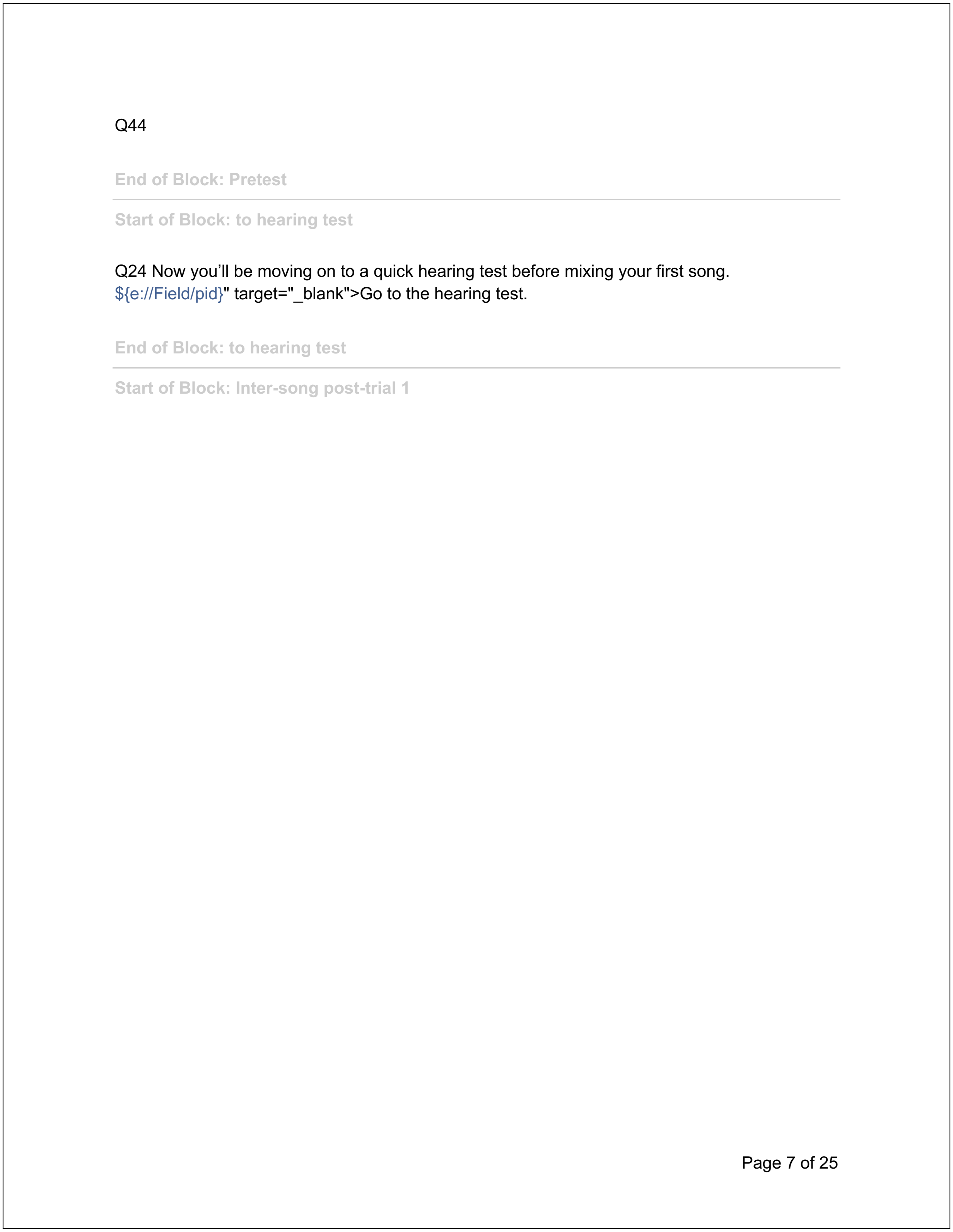
End of Block: Demographics

Start of Block: Pretest

concepts How familiar are you with each of the following concepts?

	Extremely familiar (1)	Very familiar (2)	Moderately familiar (3)	Slightly familiar (4)	Not familiar at all (5)
Equal-loudness contours (1)	○	○	○	○	○
Sones (2)	○	○	○	○	○
Phons (3)	○	○	○	○	○
Sound pressure level (4)	○	○	○	○	○
Frequency masking (5)	○	○	○	○	○
Critical bandwidth (6)	○	○	○	○	○

Page 6 of 25



Q44

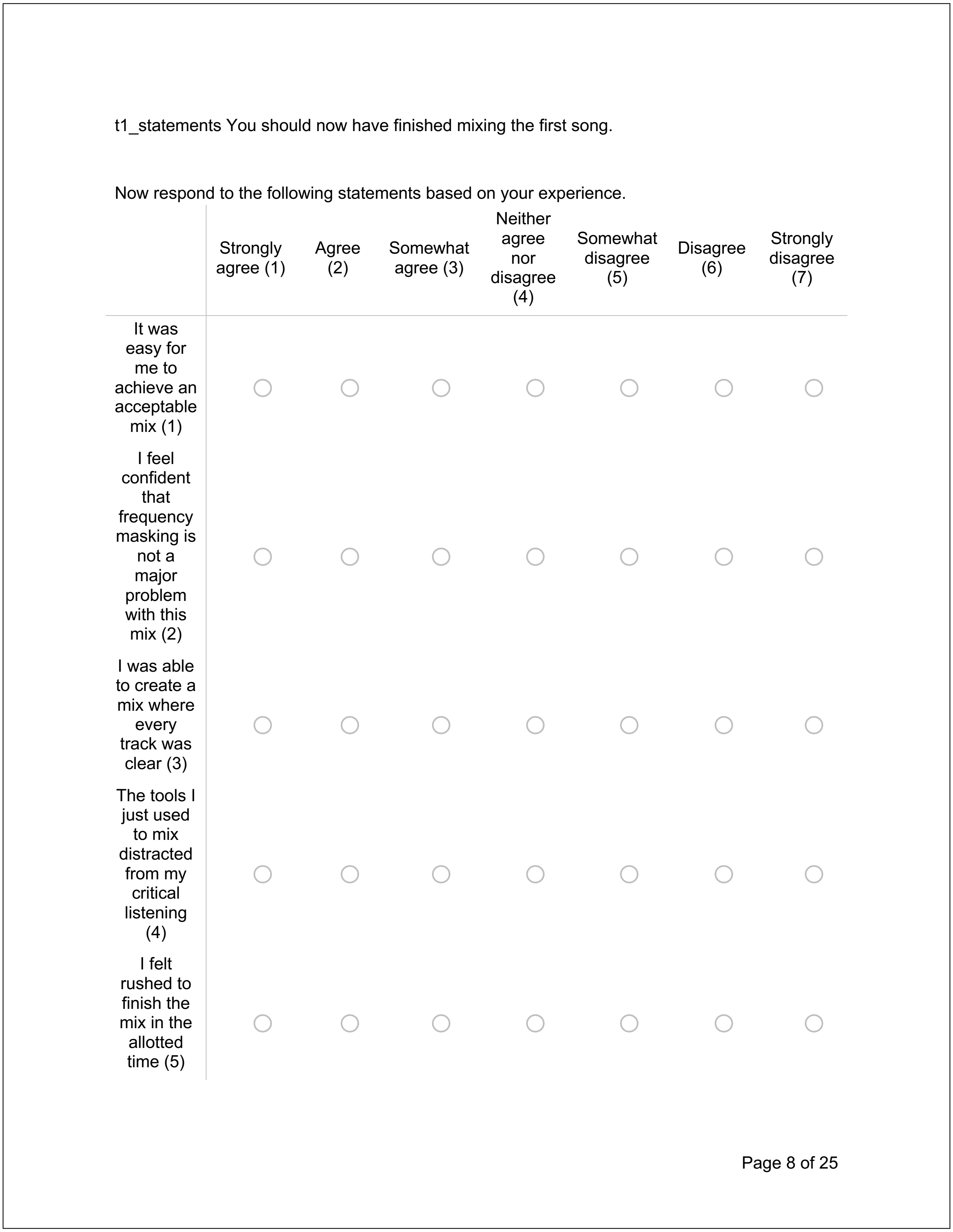
End of Block: Pretest

Start of Block: to hearing test

Q24 Now you’ll be moving on to a quick hearing test before mixing your first song.
${e://Field/pid}" target="_blank">Go to the hearing test.

End of Block: to hearing test

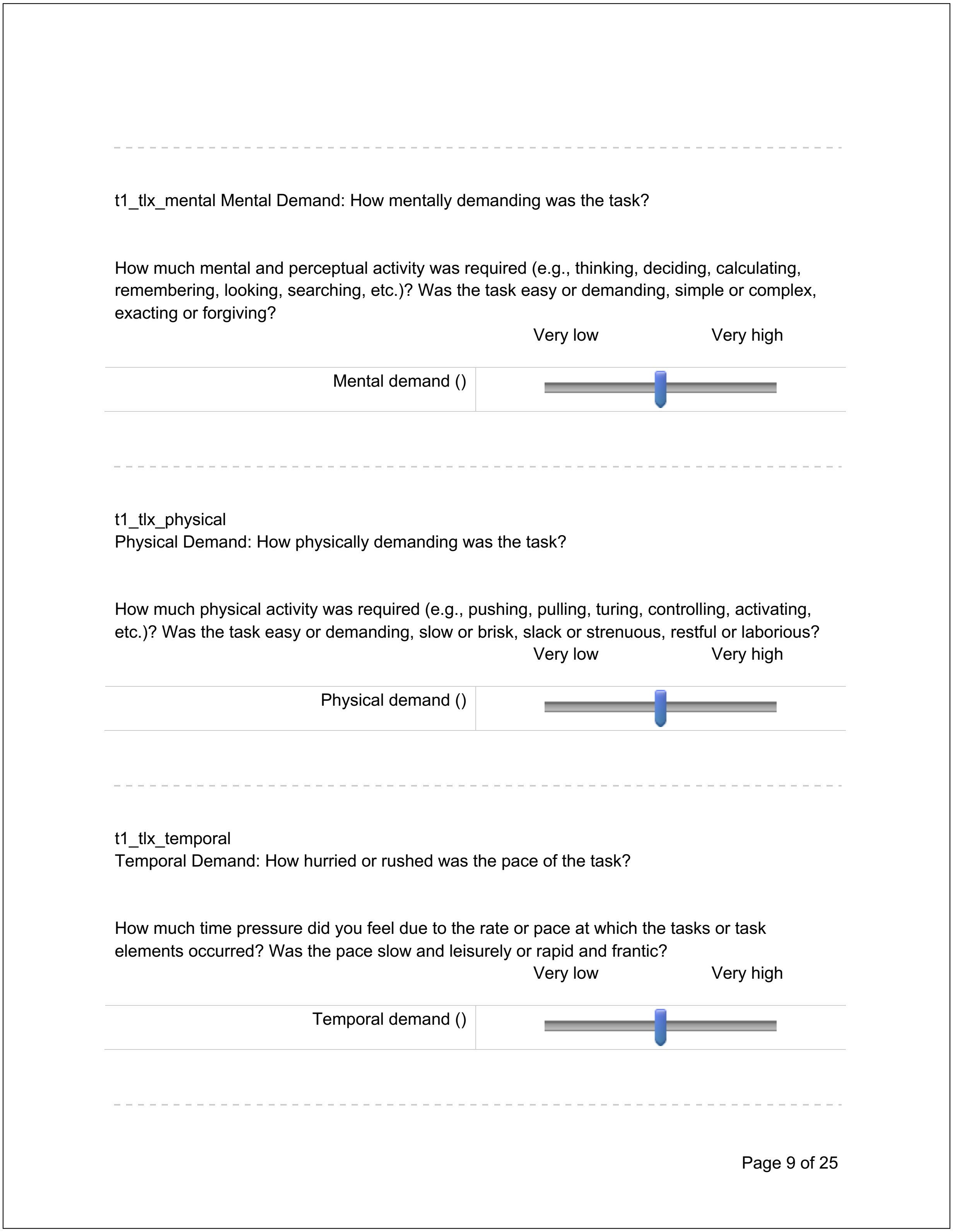
Start of Block: Inter-song post-trial 1



t1_statements You should now have finished mixing the first song.

Now respond to the following statements based on your experience.

	Strongly agree (1)	Agree (2)	Somewhat agree (3)	Neither agree nor disagree (4)	Somewhat disagree (5)	Disagree (6)	Strongly disagree (7)
It was easy for me to achieve an acceptable mix (1)	○	○	○	○	○	○	○
I feel confident that frequency masking is not a major problem with this mix (2)	○	○	○	○	○	○	○
I was able to create a mix where every track was clear (3)	○	○	○	○	○	○	○
The tools I just used to mix distracted from my critical listening (4)	○	○	○	○	○	○	○
I felt rushed to finish the mix in the allotted time (5)	○	○	○	○	○	○	○



t1_tlx_mental Mental Demand: How mentally demanding was the task?

How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?

	Very low	Very high
Mental demand ()		

t1_tlx_physical
Physical Demand: How physically demanding was the task?

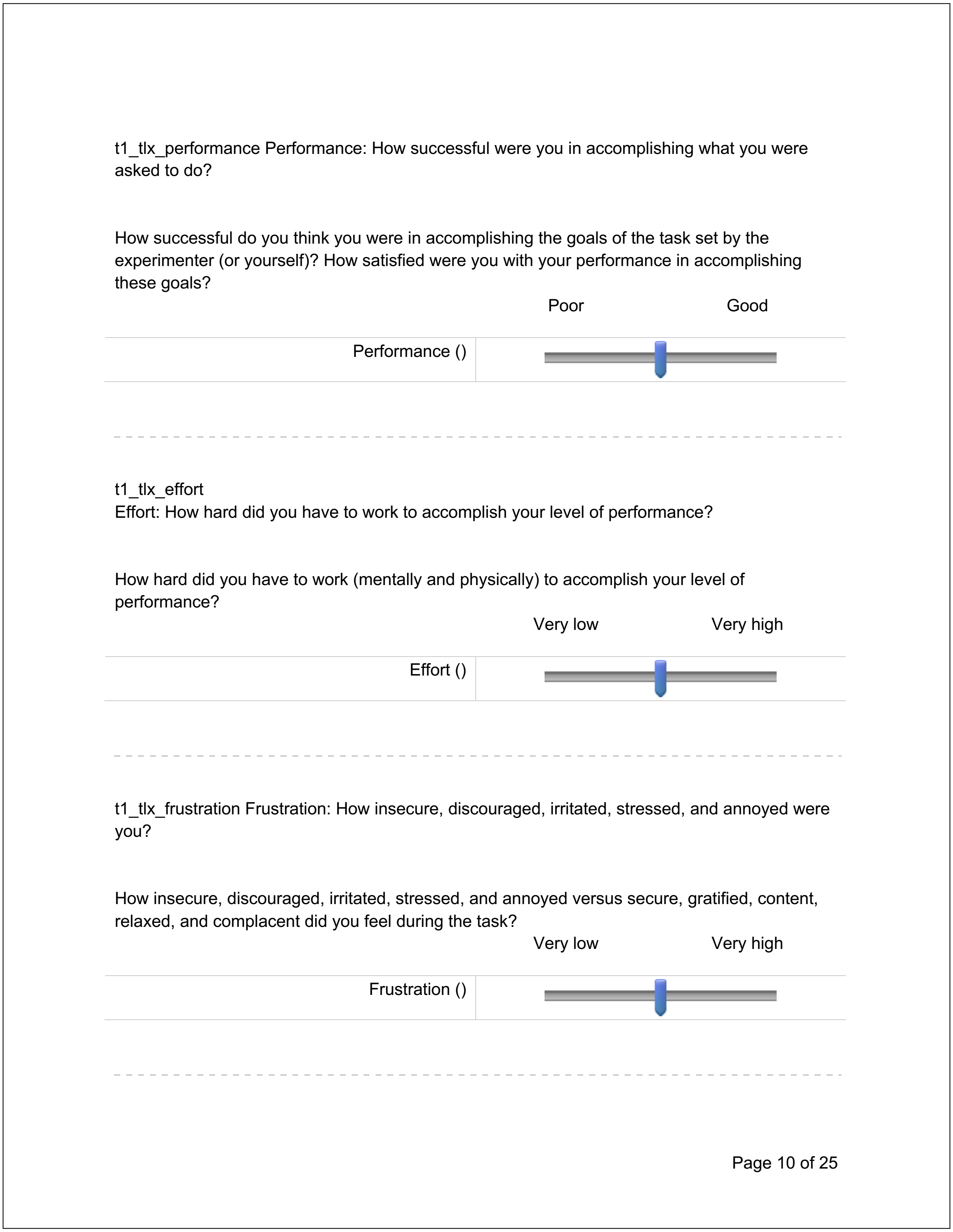
How much physical activity was required (e.g., pushing, pulling, turing, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

	Very low	Very high
Physical demand ()		

t1_tlx_temporal
Temporal Demand: How hurried or rushed was the pace of the task?

How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

	Very low	Very high
Temporal demand ()		



t1_tlx_performance Performance: How successful were you in accomplishing what you were asked to do?

How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

	Poor	Good
Performance ()		

t1_tlx_effort
Effort: How hard did you have to work to accomplish your level of performance?

How hard did you have to work (mentally and physically) to accomplish your level of performance?

	Very low	Very high
Effort ()		

t1_tlx_frustration Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?

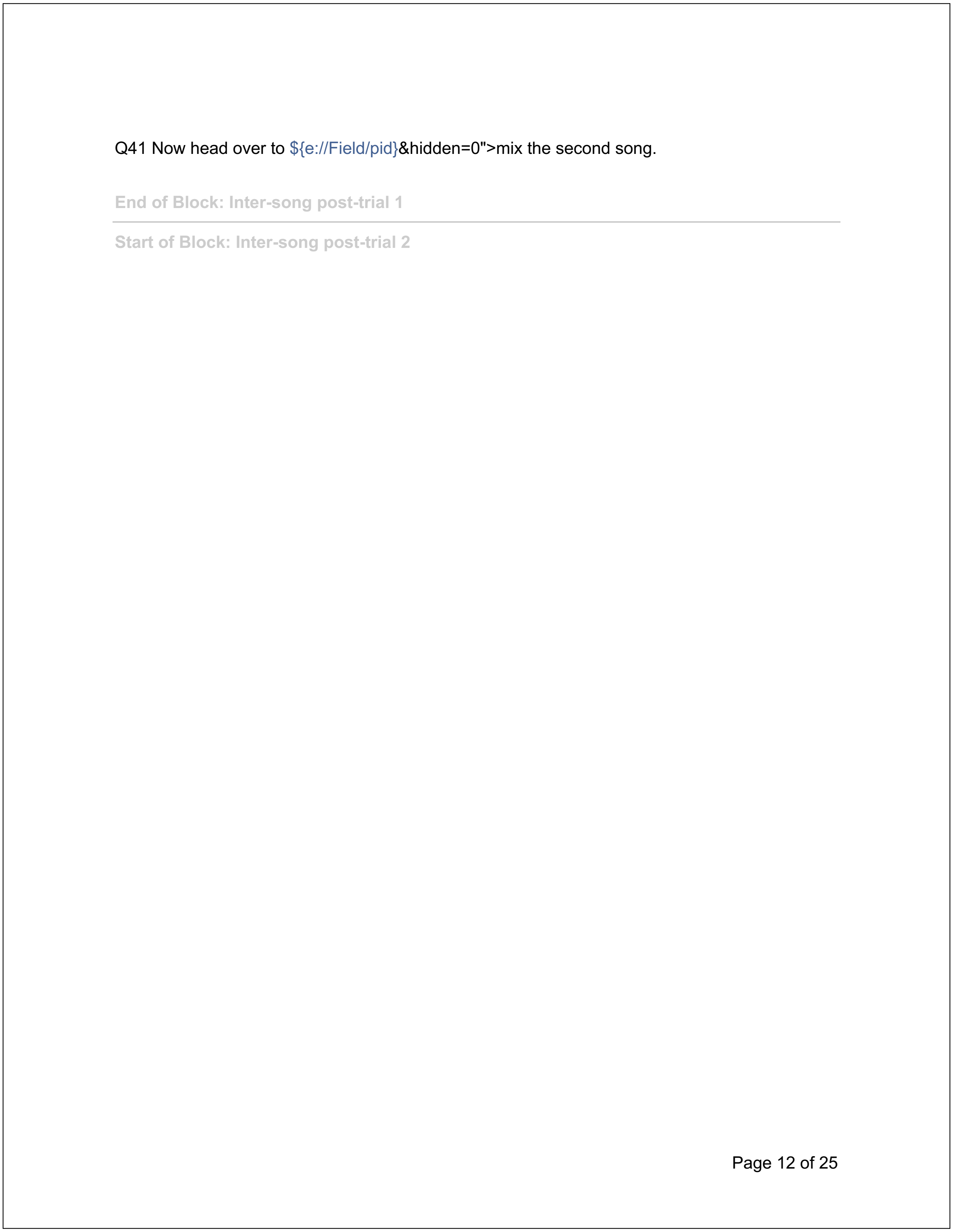
How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task?

	Very low	Very high
Frustration ()		

Q45

Page Break

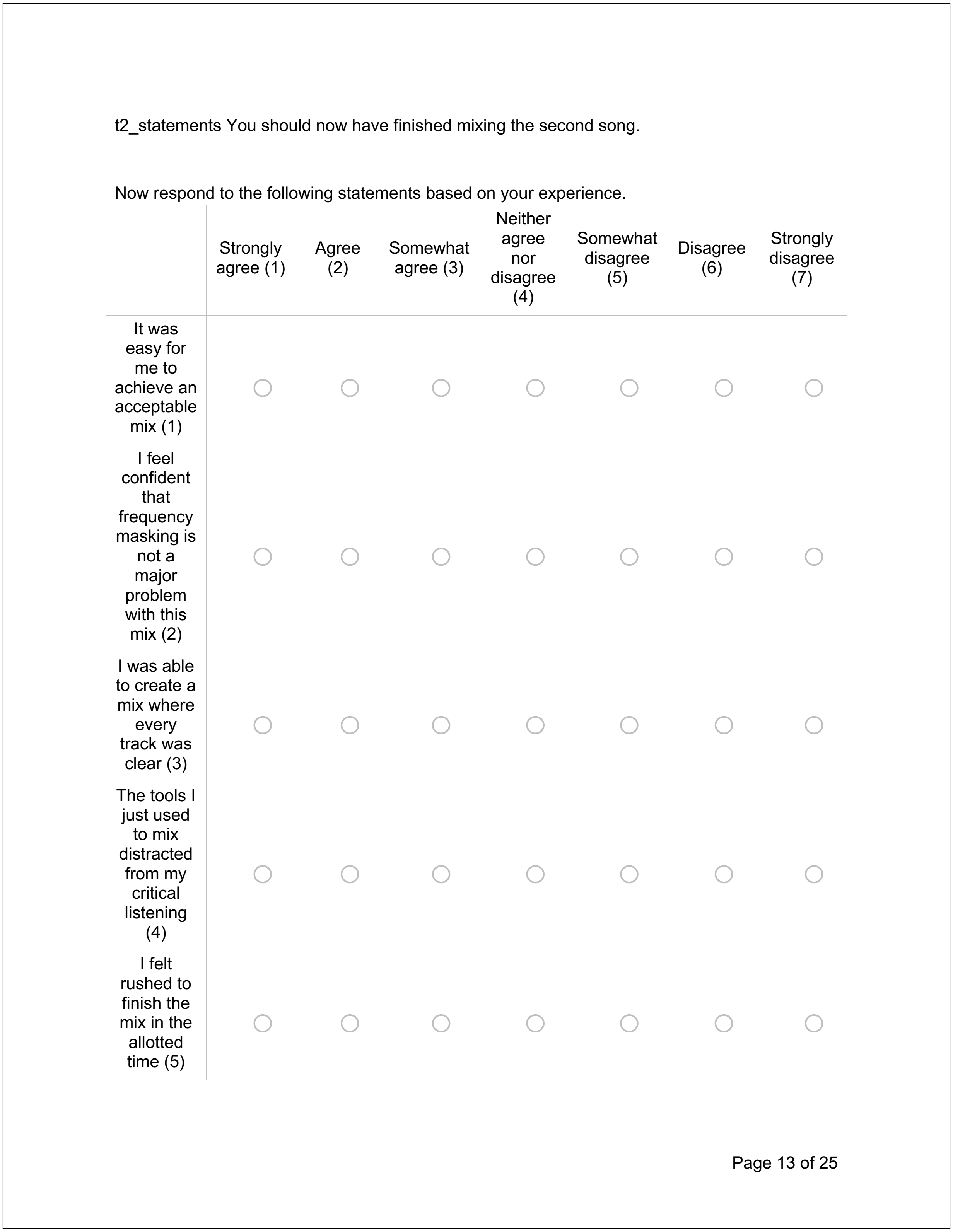
Page 11 of 25



Q41 Now head over to ${e://Field/pid}&hidden=0">mix the second song.

End of Block: Inter-song post-trial 1

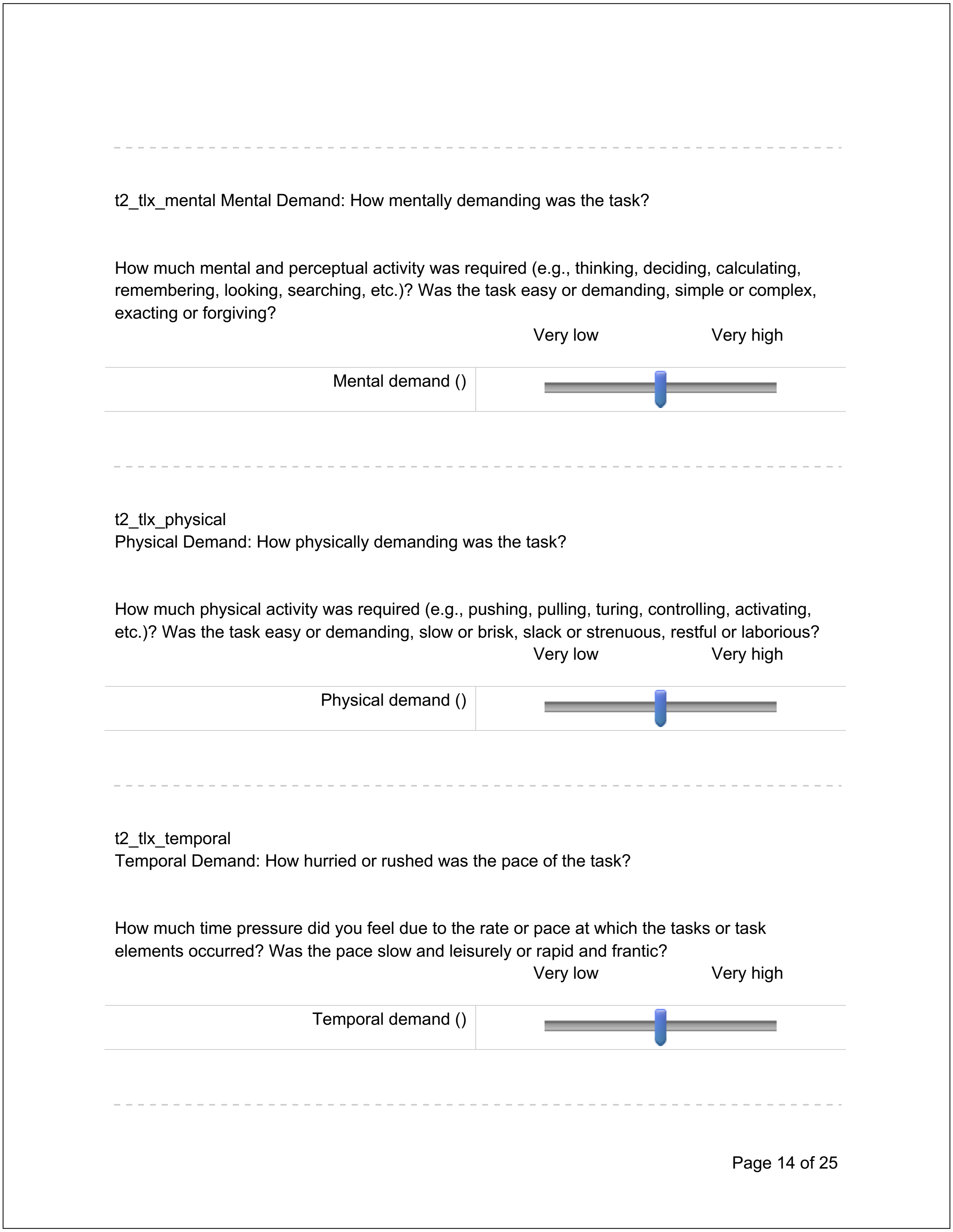
Start of Block: Inter-song post-trial 2



t2_statements You should now have finished mixing the second song.

Now respond to the following statements based on your experience.

	Strongly agree (1)	Agree (2)	Somewhat agree (3)	Neither agree nor disagree (4)	Somewhat disagree (5)	Disagree (6)	Strongly disagree (7)
It was easy for me to achieve an acceptable mix (1)	○	○	○	○	○	○	○
I feel confident that frequency masking is not a major problem with this mix (2)	○	○	○	○	○	○	○
I was able to create a mix where every track was clear (3)	○	○	○	○	○	○	○
The tools I just used to mix distracted from my critical listening (4)	○	○	○	○	○	○	○
I felt rushed to finish the mix in the allotted time (5)	○	○	○	○	○	○	○



t2_tlx_mental Mental Demand: How mentally demanding was the task?

How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?

	Very low Very high
Mental demand ()	

t2_tlx_physical
Physical Demand: How physically demanding was the task?

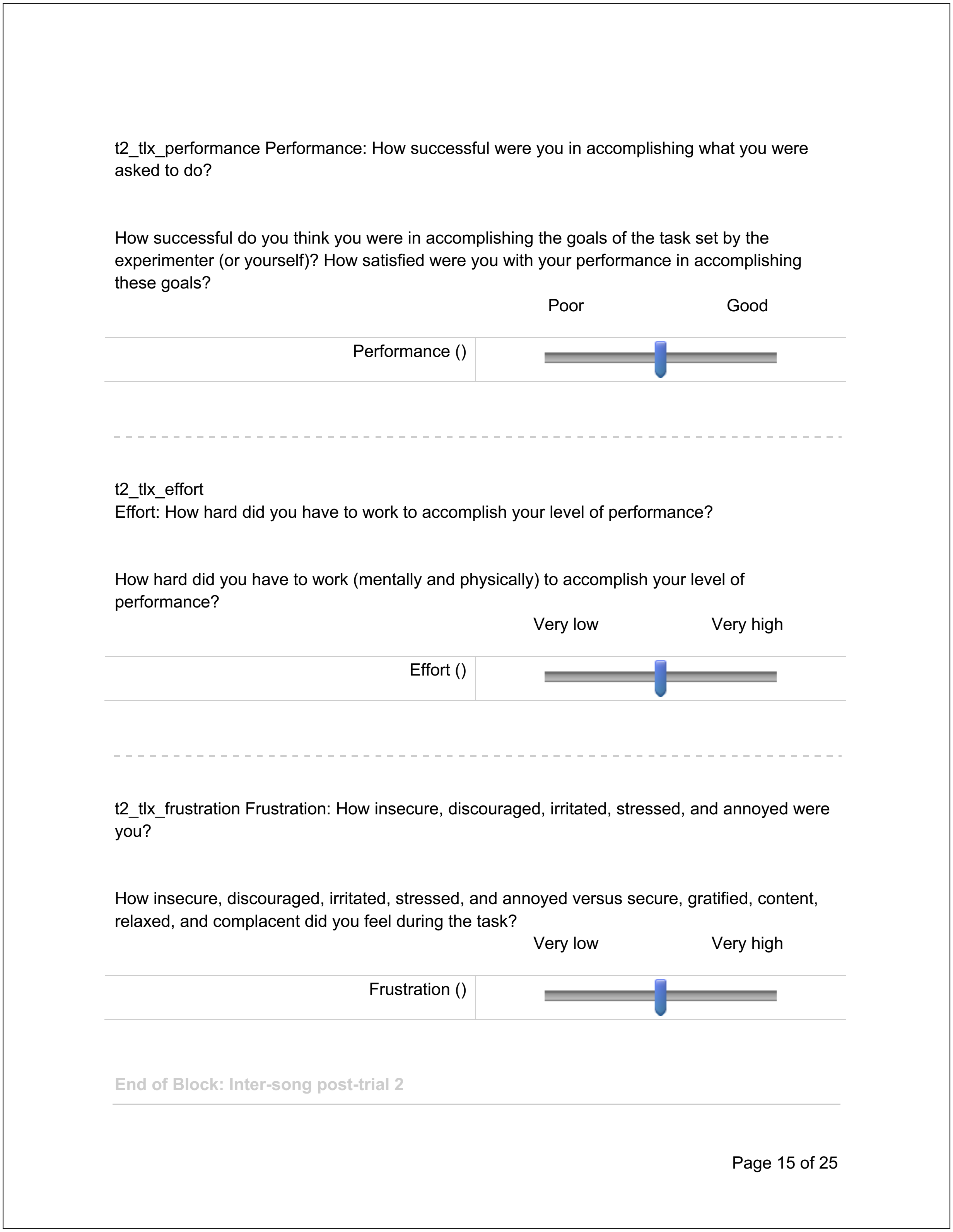
How much physical activity was required (e.g., pushing, pulling, turing, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

	Very low Very high
Physical demand ()	

t2_tlx_temporal
Temporal Demand: How hurried or rushed was the pace of the task?

How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

	Very low Very high
Temporal demand ()	



t2_tlx_performance Performance: How successful were you in accomplishing what you were asked to do?

How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

	Poor	Good
Performance ()		

t2_tlx_effort
Effort: How hard did you have to work to accomplish your level of performance?

How hard did you have to work (mentally and physically) to accomplish your level of performance?

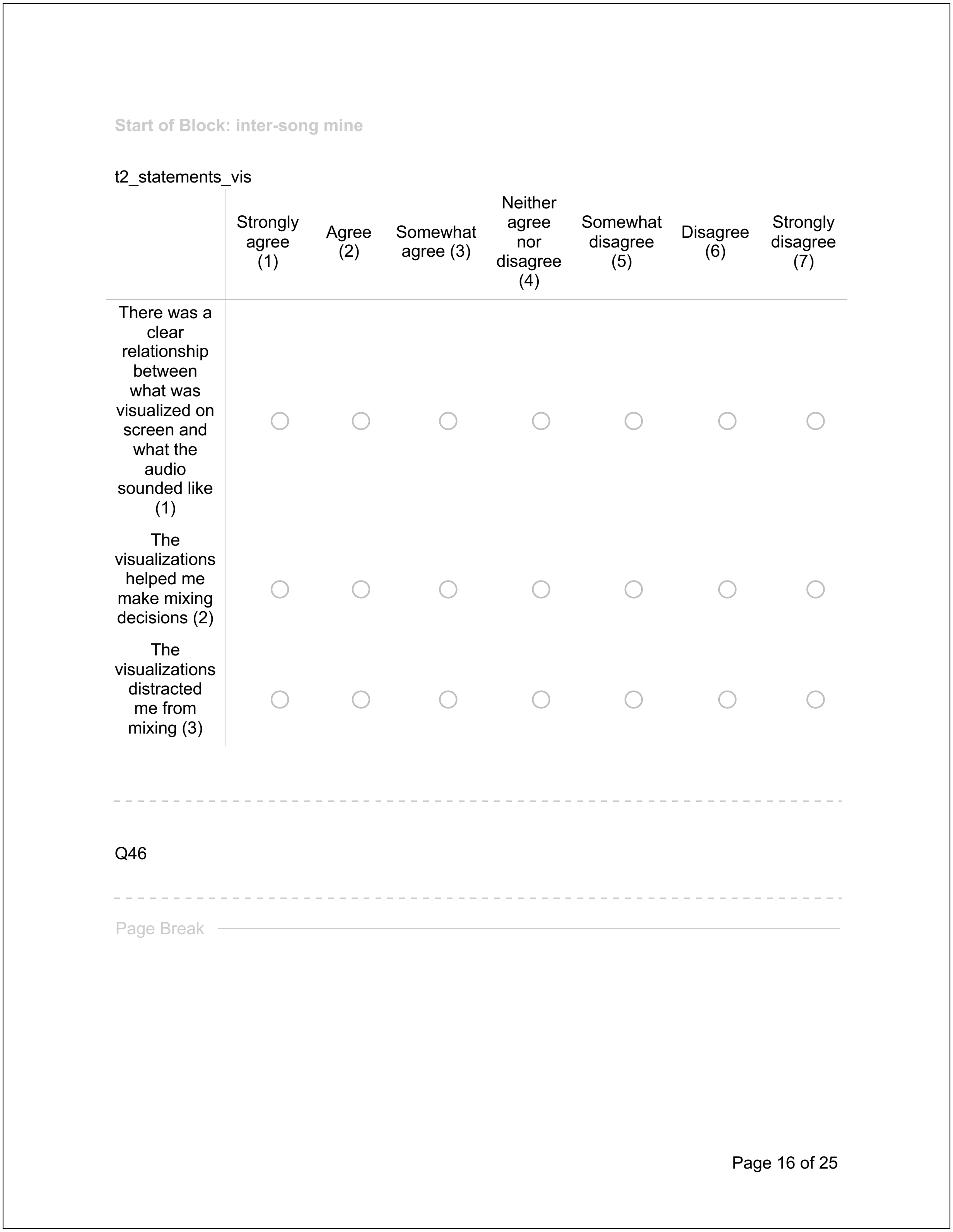
	Very low	Very high
Effort ()		

t2_tlx_frustration Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?

How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task?

	Very low	Very high
Frustration ()		

End of Block: Inter-song post-trial 2



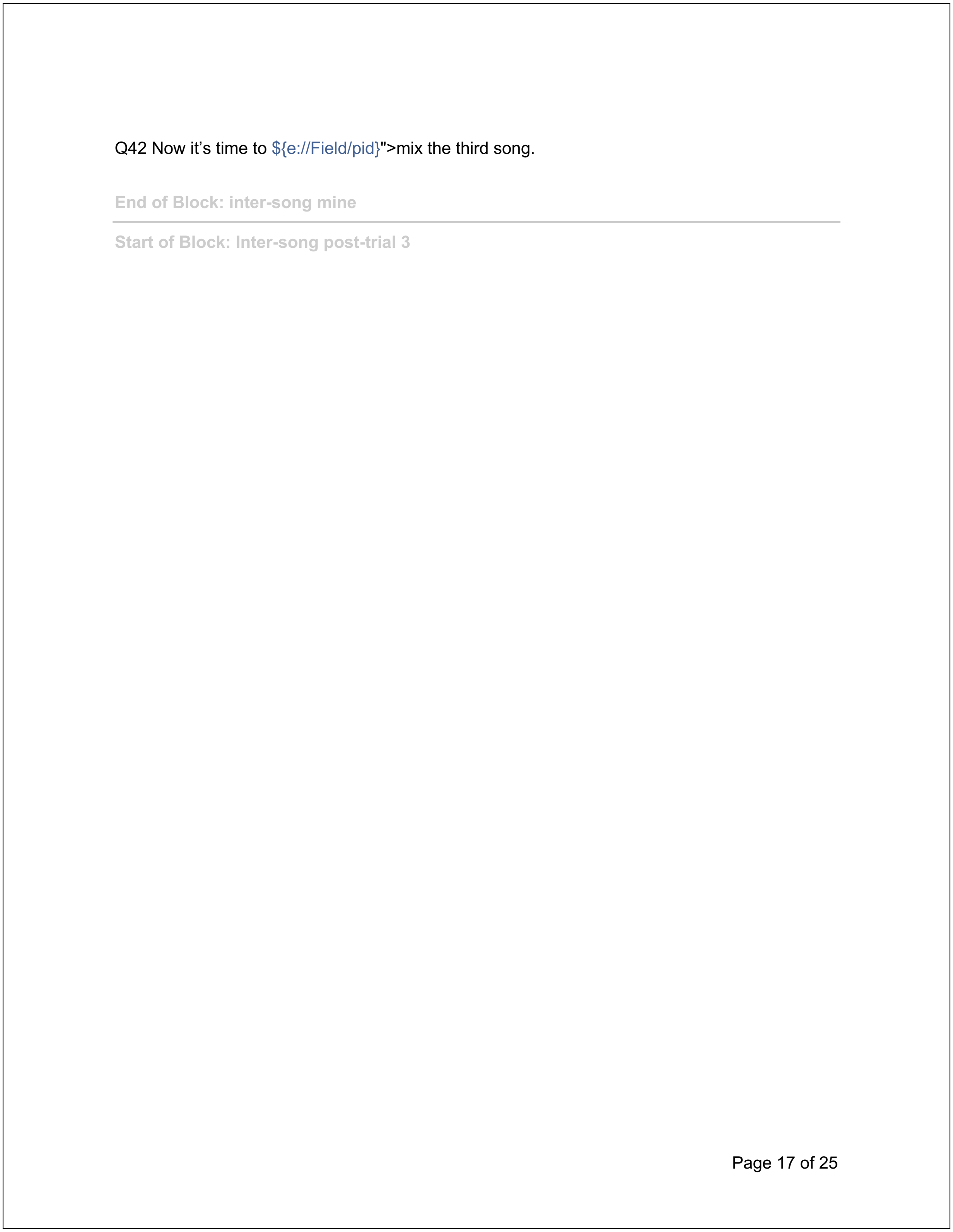
Start of Block: inter-song mine

t2_statements_vis

	Strongly agree (1)	Agree (2)	Somewhat agree (3)	Neither agree nor disagree (4)	Somewhat disagree (5)	Disagree (6)	Strongly disagree (7)
There was a clear relationship between what was visualized on screen and what the audio sounded like (1)	○	○	○	○	○	○	○
The visualizations helped me make mixing decisions (2)	○	○	○	○	○	○	○
The visualizations distracted me from mixing (3)	○	○	○	○	○	○	○

Q46

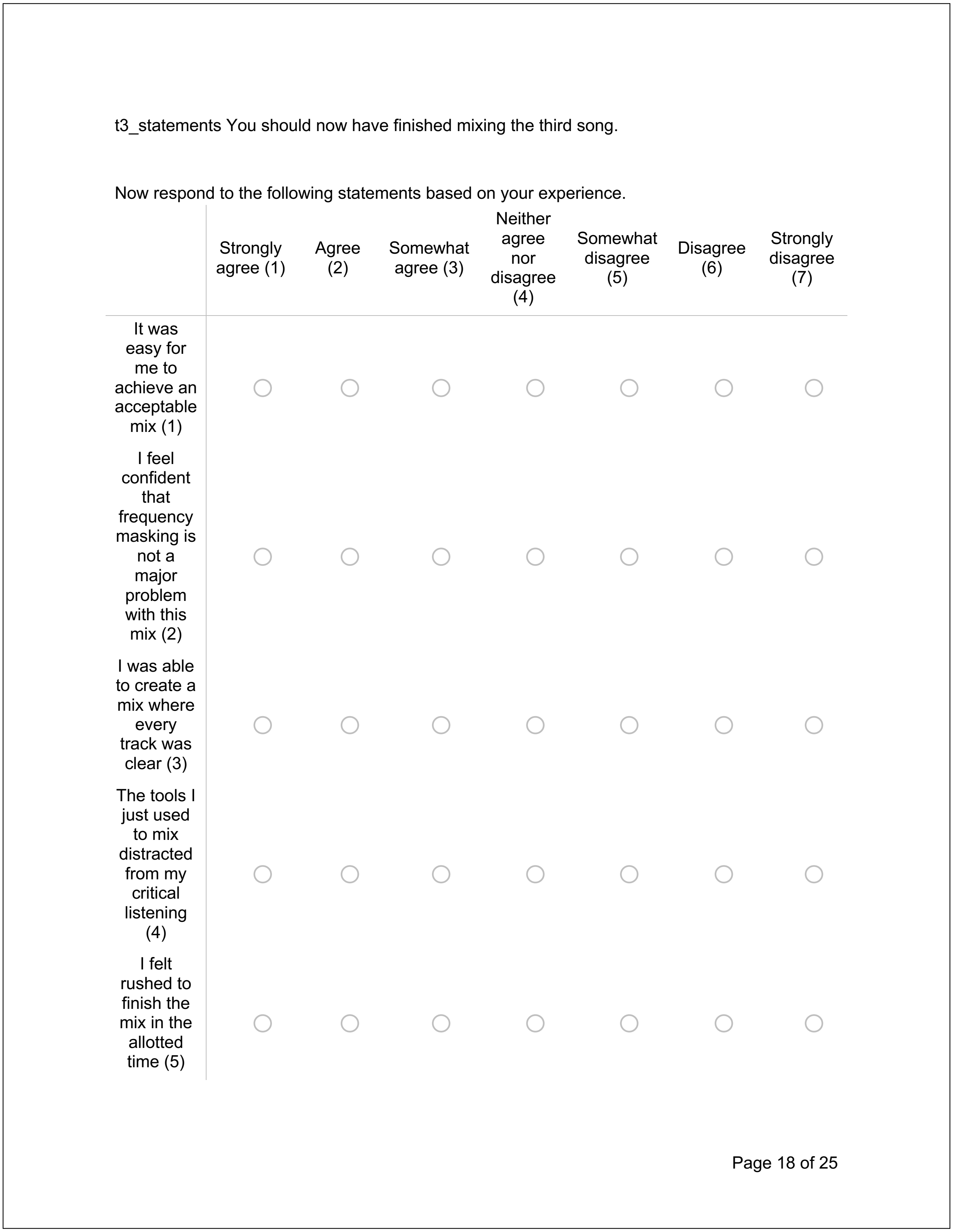
Page Break



Q42 Now it’s time to ${e://Field/pid}">mix the third song.

End of Block: inter-song mine

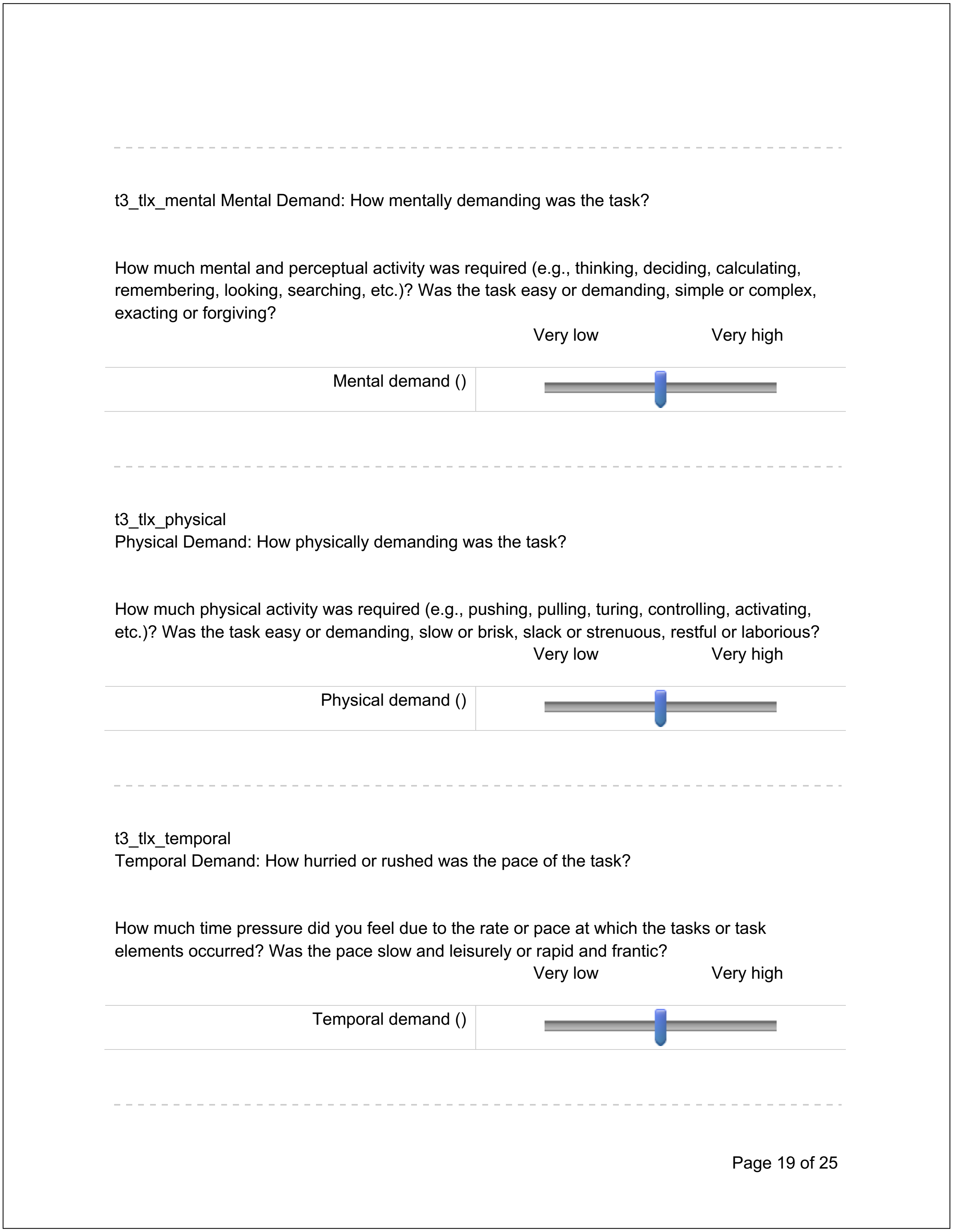
Start of Block: Inter-song post-trial 3



t3_statements You should now have finished mixing the third song.

Now respond to the following statements based on your experience.

	Strongly agree (1)	Agree (2)	Somewhat agree (3)	Neither agree nor disagree (4)	Somewhat disagree (5)	Disagree (6)	Strongly disagree (7)
It was easy for me to achieve an acceptable mix (1)	○	○	○	○	○	○	○
I feel confident that frequency masking is not a major problem with this mix (2)	○	○	○	○	○	○	○
I was able to create a mix where every track was clear (3)	○	○	○	○	○	○	○
The tools I just used to mix distracted from my critical listening (4)	○	○	○	○	○	○	○
I felt rushed to finish the mix in the allotted time (5)	○	○	○	○	○	○	○



t3_tlx_mental Mental Demand: How mentally demanding was the task?

How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?

	Very low Very high
Mental demand ()	

t3_tlx_physical
Physical Demand: How physically demanding was the task?

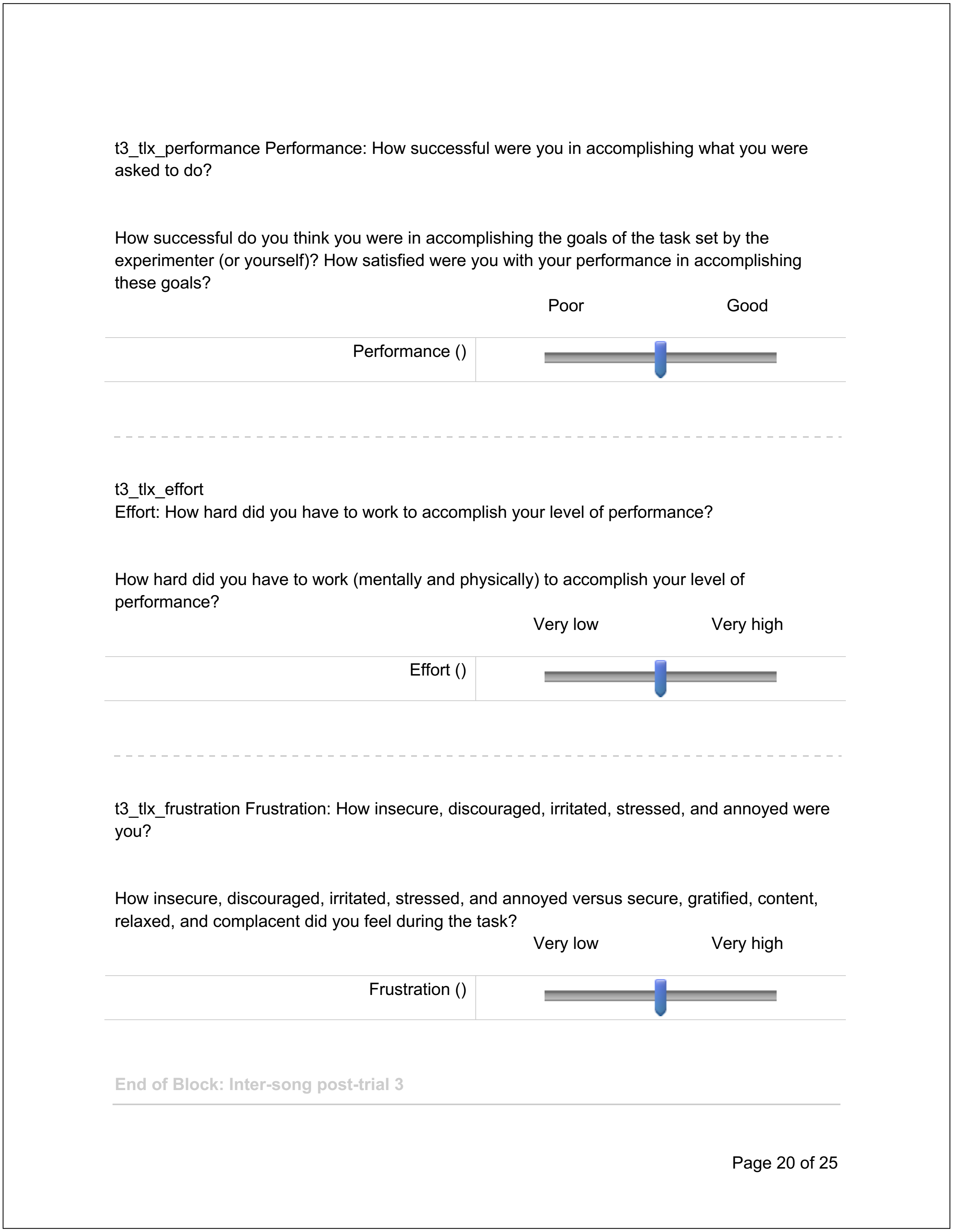
How much physical activity was required (e.g., pushing, pulling, turing, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?

	Very low Very high
Physical demand ()	

t3_tlx_temporal
Temporal Demand: How hurried or rushed was the pace of the task?

How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

	Very low Very high
Temporal demand ()	



t3_tlx_performance Performance: How successful were you in accomplishing what you were asked to do?

How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?

	Poor	Good
Performance ()		

t3_tlx_effort
Effort: How hard did you have to work to accomplish your level of performance?

How hard did you have to work (mentally and physically) to accomplish your level of performance?

	Very low	Very high
Effort ()		

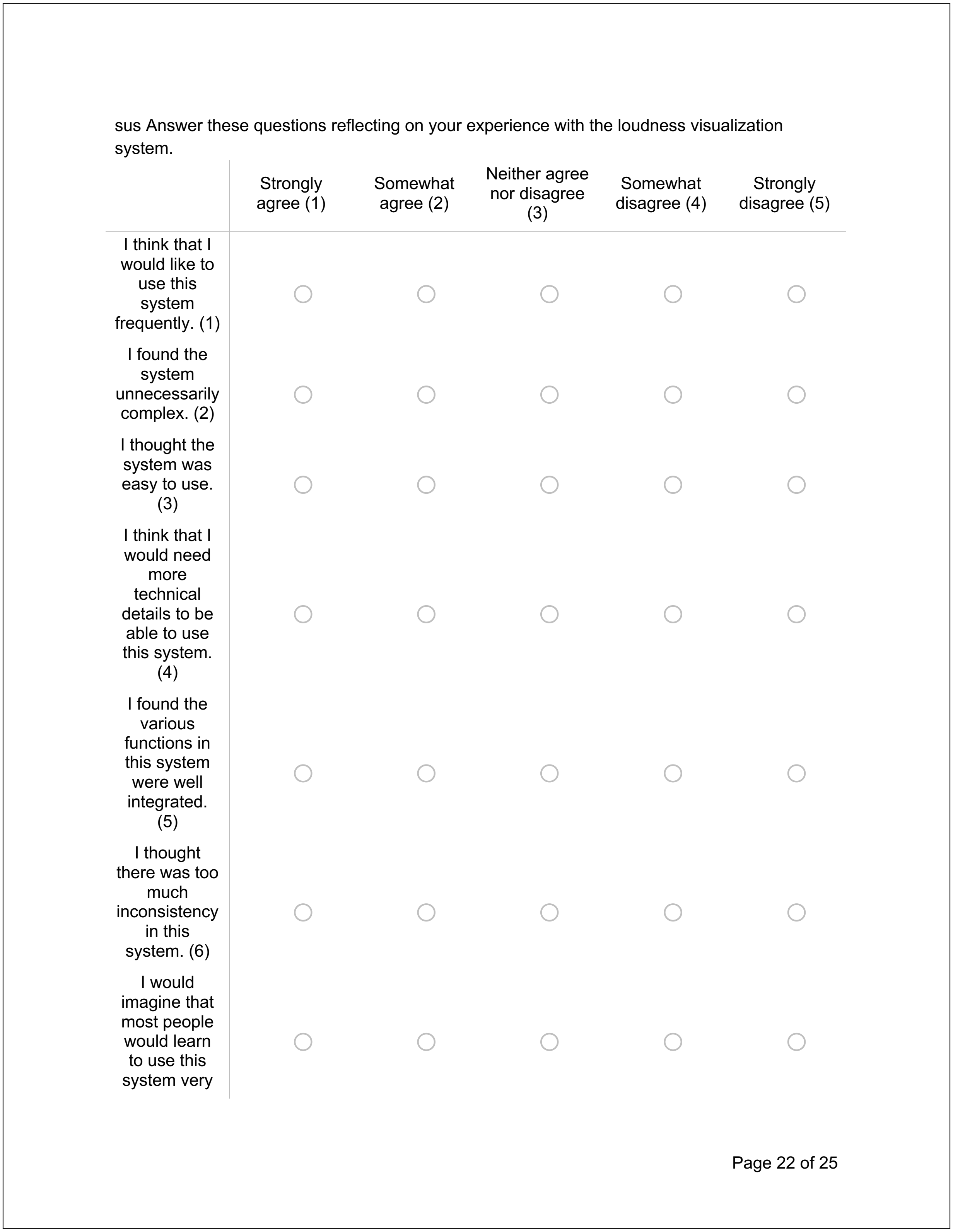
t3_tlx_frustration Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?

How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task?

	Very low	Very high
Frustration ()		

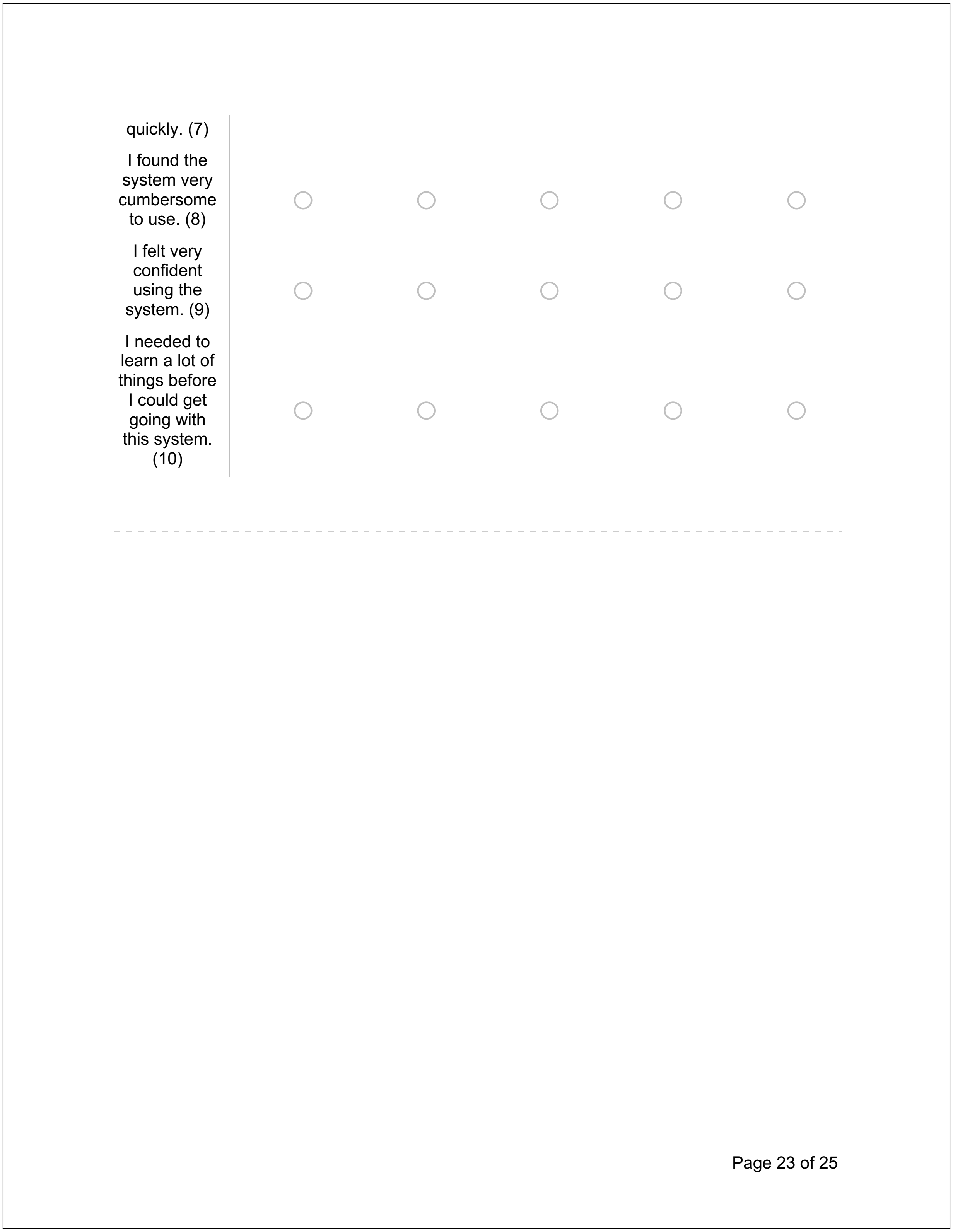
End of Block: Inter-song post-trial 3

Start of Block: SUS

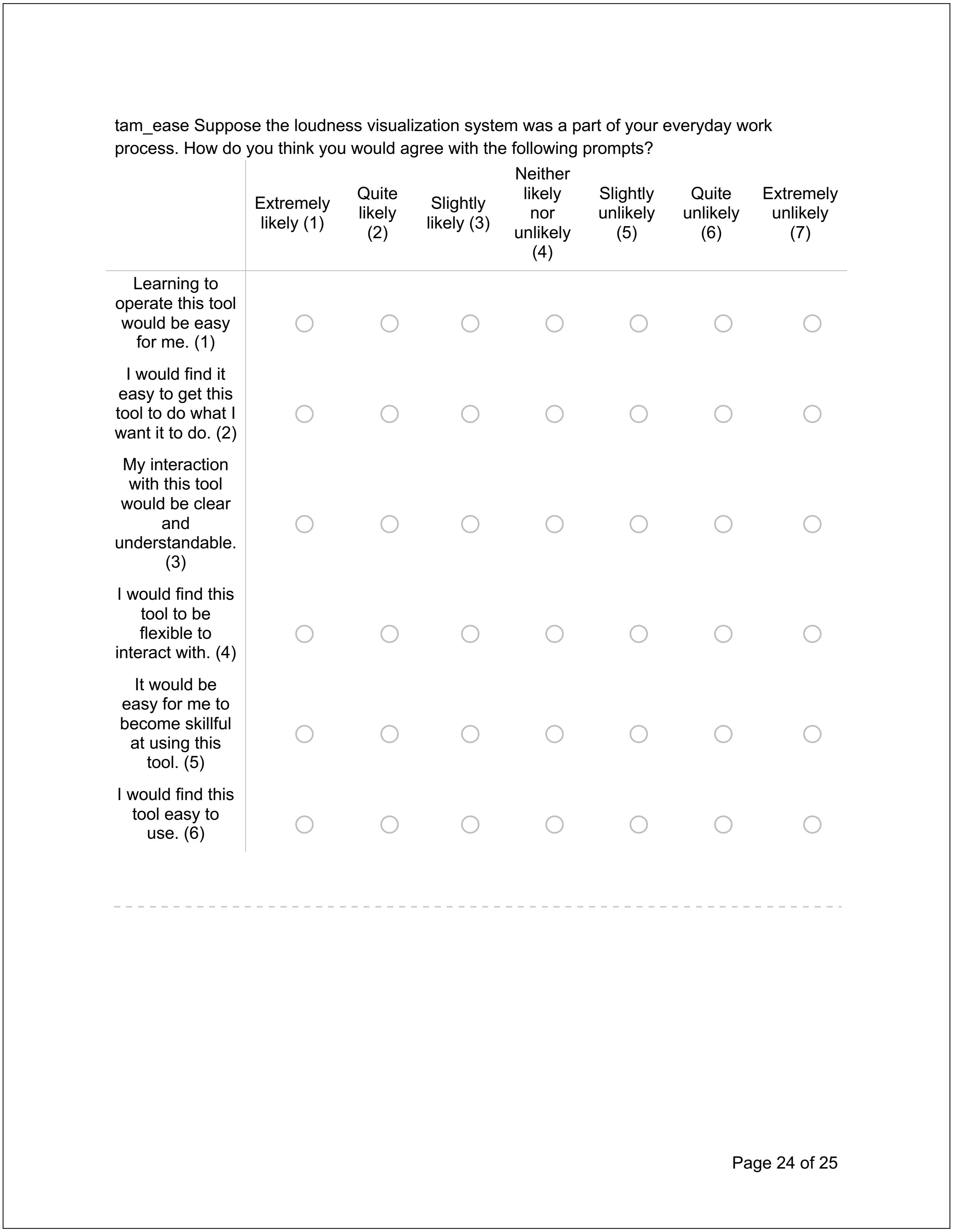


sus Answer these questions reflecting on your experience with the loudness visualization system.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
I think that I would like to use this system frequently. (1)	○	○	○	○	○
I found the system unnecessarily complex. (2)	○	○	○	○	○
I thought the system was easy to use. (3)	○	○	○	○	○
I think that I would need more technical details to be able to use this system. (4)	○	○	○	○	○
I found the various functions in this system were well integrated. (5)	○	○	○	○	○
I thought there was too much inconsistency in this system. (6)	○	○	○	○	○
I would imagine that most people would learn to use this system very	○	○	○	○	○



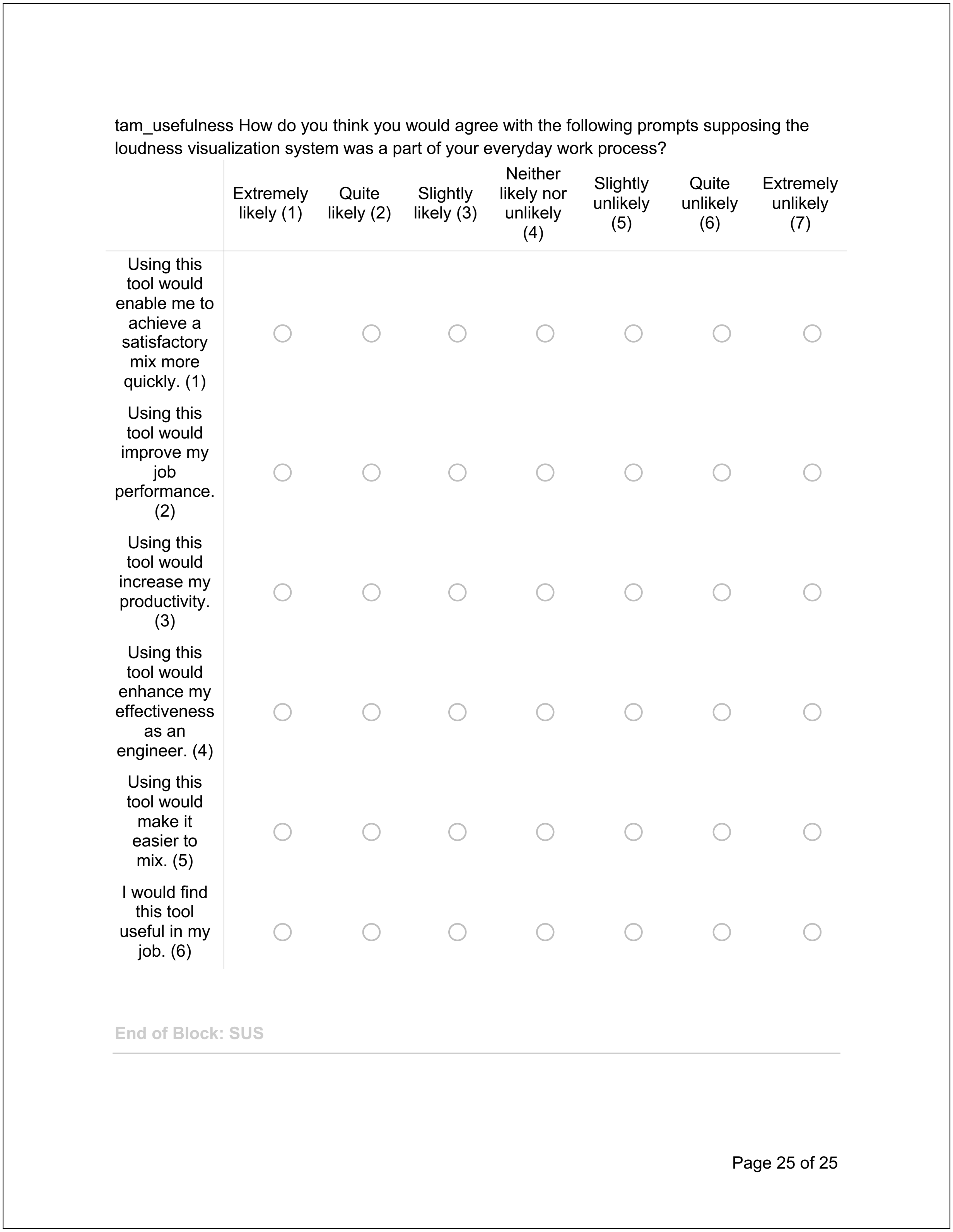
quickly. (7)					
I found the system very cumbersome to use. (8)	○	○	○	○	○
I felt very confident using the system. (9)	○	○	○	○	○
I needed to learn a lot of things before I could get going with this system. (10)	○	○	○	○	○



tam_ease Suppose the loudness visualization system was a part of your everyday work process. How do you think you would agree with the following prompts?

	Extremely likely (1)	Quite likely (2)	Slightly likely (3)	Neither likely nor unlikely (4)	Slightly unlikely (5)	Quite unlikely (6)	Extremely unlikely (7)
Learning to operate this tool would be easy for me. (1)	○	○	○	○	○	○	○
I would find it easy to get this tool to do what I want it to do. (2)	○	○	○	○	○	○	○
My interaction with this tool would be clear and understandable. (3)	○	○	○	○	○	○	○
I would find this tool to be flexible to interact with. (4)	○	○	○	○	○	○	○
It would be easy for me to become skillful at using this tool. (5)	○	○	○	○	○	○	○
I would find this tool easy to use. (6)	○	○	○	○	○	○	○

Page 24 of 25



tam_usefulness How do you think you would agree with the following prompts supposing the loudness visualization system was a part of your everyday work process?

	Extremely likely (1)	Quite likely (2)	Slightly likely (3)	Neither likely nor unlikely (4)	Slightly unlikely (5)	Quite unlikely (6)	Extremely unlikely (7)
Using this tool would enable me to achieve a satisfactory mix more quickly. (1)	○	○	○	○	○	○	○
Using this tool would improve my job performance. (2)	○	○	○	○	○	○	○
Using this tool would increase my productivity. (3)	○	○	○	○	○	○	○
Using this tool would enhance my effectiveness as an engineer. (4)	○	○	○	○	○	○	○
Using this tool would make it easier to mix. (5)	○	○	○	○	○	○	○
I would find this tool useful in my job. (6)	○	○	○	○	○	○	○

End of Block: SUS

APPENDIX C

Post-study Interview Questions

- Speaking generally, how would you describe the difference between mixing with MaskerAid and without MaskerAid?
- When you're mixing (not editing) using normal waveforms, how do you use those waveforms as part of your process?
- Was that different using MaskerAid?
- Did you have a preference for one or the other? why?
- How did you think about the concept of every track being clearly audible? did that differ by song?
- Describe the relationship between what you saw on screen in MaskerAid and what you heard.
- In addition to the loudness "waves", MaskerAid also called out potential masking. How did that feature play into your mixing?
- When MaskerAid called out potential masking, did you agree? did a track sound masked? was it helpful or distracting to see that information?
- Did selecting a track to see what masks it play into your mix at all?
- I'd like to hear about your experience using the visualization while listening intently. Did the visuals help, or were they distracting?

- How did you feel this would fit into your existing way of mixing? Not just from a workflow perspective, but also in terms of your mental process and how you think about what you're doing.