

AMuST: Accessible Music Studio - Feasibility Report

Metatla, Oussama; Stockman, Tony; Bryan-Kinns, Nick

For additional information about this publication click this link.

<http://qmro.qmul.ac.uk/jspui/handle/123456789/5010>

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk

ISSN 2043-0167

AMuST: Accessible Music Studio - Feasibility Report

Oussama Metatla, Tony Stockman and Nick Bryan-Kinns



EECSRR-11-03

July 2011

School of Electronic Engineering
and Computer Science

An abstract graphic featuring a detailed image of a circuit board. Overlaid on the board are glowing white lines and circles, suggesting a network or data flow. The text "Computer Science" is written in a large, white, serif font, with "Computer" on the top line and "Science" on the bottom line.

Computer
Science

Electronic
Engineering

AMuST: Accessible Music Studio - Feasibility Report

Report prepared by Oussama Metatla, Tony Stockman and Nick Bryan-Kinns, Interaction Media and Communication research group and Centre for Digital Music,
School of Electronic Engineering and Computer Science, Queen Mary University of London.
In collaboration with Peter Boshier of SoundLinks Ltd.

**43 Broadlands Avenue,
Chesham,
Bucks. HP5 1AL,**

September 2009

Table of Contents

1 INTRODUCTION	5
1.1 OVERVIEW	5
1.2 OBJECTIVES	5
1.3 DELIVERABLES	6
1.4 PROJECT ACTIVITIES & SCHEDULE	6
1.4.1 STAGE 1 - REQUIREMENTS INVESTIGATION	6
1.4.2 STAGE 2 - EVALUATION OF DESIGN SOLUTIONS	7
1.4.3 STAGE 3 - SYNTHESIS	7
1.5 REPORT STRUCTURE	7
2 PROJECT PROCESS, COSTS & INNOVATION	9
2.1 PROCESS	9
2.2 COSTS	10
2.3 INNOVATION	11
3 REVIEW	14
3.1 DIGITAL AUDIO WORKSTATIONS	14
3.2 DAWs AND ACCESSIBILITY	21
3.2.1 SCRIPTS FOR SCREEN-READERS	21
3.2.2 NOTATION-BASED SOLUTIONS	24
3.3 REFLECTIONS ON CURRENT ACCESSIBILITY SOLUTIONS	26
4 CASE STUDIES	31
4.1 CASE 1 - PROFESSIONAL MUSICIAN	32
4.2 CASE 2 - ACCESSIBILITY TRAINER	35
4.3 CASE 3 - AUDIO PRODUCER	37
4.4 CASE 4 - EDUCATOR & HOBBYIST	40
4.5 USABILITY CONSIDERATIONS AND CATEGORISATION	43
5 INFORMING THE DESIGN - AMUST	48
5.1 AUDITORY DISPLAY DESIGN	48
5.2 HAPTIC DISPLAY DESIGN	51
5.3 SYNTHESISING THE DESIGN	53
5.3.1 PACE AND STRUCTURE OF INTERACTION	53
5.3.2 SPATIAL MANIPULATIONS THROUGH A TEMPORAL MEDIUM	55
5.3.3 INFORMATION OVERLOAD	56
5.4 SUMMARY	56
6 CONCLUSIONS	58
REFERENCES	60
APPENDIX 1: WORKING PLAN	62
APPENDIX 2 - QUESTIONNAIRES FOR CASE STUDIES	63

1 Introduction

1.1 Overview

Modern music and sound editors employ multi-function mixing and transport controls that rely heavily on visual displays for multi-track editing and production. This situation imposes severe limitations on the autonomy of visually impaired users within a field that could be a natural one for them, that of sound and music production. The reported study investigates the feasibility of designing an accessible music and audio production studio to improve access to training and jobs for visually impaired people in the creative industries.

The *Accessible Music Studio (AMuST)* is a project which will provide contextual non-visual feedback for visually impaired users by creating ways in which they can edit music and sound guided by a non-visual interface. The net result would be to enable visually impaired individuals to be at the forefront of digital music and sound production technology with tools that are designed specifically for their needs. In particular, such tools would allow visually impaired professionals and students to use music technology to better integrate into the creative industries as editors, producers and composers. The availability of a fully accessible music and audio production studio would considerably lower the skill level required of visually impaired users to obtain jobs in music and audio production. This will in turn grow the market for the services of companies such as SoundLinks Ltd., as well as opening the market to a wider range of visually impaired people who wish to work as trainers in music and audio technology.

1.2 Objectives

This project investigates the feasibility of designing accessible music studio software for visually impaired and blind musicians and sound engineers. In particular, the aim is to identify usability issues in current accessibility solutions, and ways in which they could be overcome through employment of novel interactive design techniques. This will be achieved through an examination of state-of-the-art approaches to providing accessibility to mainstream music and sound editing tools, as well as through close interaction with experienced visually impaired and blind musicians and audio

engineering practitioners.

Novel approaches to non-visual interface design will be explored in order to identify how research-based techniques could be combined and synthesised to deliver more usable accessibility to music and sound editing. The measurable objectives with relation to the above aim are therefore to:

- Review existing solutions to supporting music and sound editing for visually impaired users.
- Examine current practices and user experiences with current available accessibility solutions to music and sound editing.
- Categorise usability issues encountered by visually impaired musicians and audio engineers when using available solutions.
- Design recommendations for an accessible music and sound editing studio that combine novel techniques of non-visual human-computer interaction.

1.3 Deliverables

The primary deliverable of this project is a feasibility report containing detailed reviews of current accessibility technology, practitioners' experiences with these solutions, as well as detailed recommendations for the design of non-visual interaction techniques for music and sound editing.

1.4 Project Activities & Schedule

The activities of the projects were organised into three overlapping stages; Requirements Investigation; Evaluation of Design Solutions; and Synthesis. A Gantt chart depicting the schedule of the various activities that we undertook in each stage in order to achieve the deliverables is presented in Appendix 1 of this report. In general, project activities were managed through regular meetings with key members of the project team, which were held weekly in order to assess progress and intercept problems. The following are brief descriptions of each stage and the nature of the activities it comprised.

1.4.1 Stage 1 - Requirements Investigation

The requirements for an accessible music and sound editor were investigated through

two activities; a review of existing accessibility solutions; and an examination of case studies of user experiences with these solutions. Section two and three of this report presents the details and outcomes of these activities.

1.4.2 Stage 2 - Evaluation of Design Solutions

Design options were explored through brainstorming sessions that were held between key members of the project team. Informed by the reported case studies of usage and user experiences, ideas formulation sessions were then organised in order to explore and evaluate how novel non-visual interface design techniques could be combined into a fully accessible system. Section four of this report presents the outcome of this stage.

1.4.3 Stage 3 - Synthesis

Using the reviews and the gathered information from the users case studies, we have categorised the type of issues and challenges frequently encountered by users when interacting with current accessibility solutions to music and sound editing. This categorisation was used to drive the ideas formulation sessions, where a set of interface design techniques based on research on non-visual interaction were explored, structured and combined. Details of the synthesis are also presented in Section four of the report.

1.5 Report Structure

The remaining content of this report is organised into five sections. Section two reviews the process of carrying out the project, comparing outcomes with the original project proposal. The project costs are described, the technology involved, the innovation in the project and its impact. Section three presents an overview of mainstream music and sound editing tools, with a closer look at available solutions and existing approaches for making such tools accessible to visually impaired users. Section four then reports on a number of case studies of real user experiences with current accessibility solutions. The limitations of these solutions are highlighted, and a categorisation of usability issues is produced. Section five then presents research-based techniques and guidelines for designing accessibility solutions to music and sound editing, and the report concludes with thoughts on future development.

2 Project Process, Costs & Innovation

2.1 Process

The start of the project was delayed substantially for a number of reasons, including one of the project team members sustaining a serious injury requiring hospitalisation, and a change in the domestic arrangements of another team member. However, once the project began at the start of August 2009, it ran entirely to schedule and was completed successfully by the end of September 2009.

The initial requirements investigation was successful in carrying out a wide ranging examination of the spectrum of current solutions employed by visually impaired musicians and audio producers, which is fully documented in section 3 of this report. The information was gathered through a user experience survey, detailed reviews of current products, individual discussions, examination of specialised e-mail list archives and practical computer-based evaluations examining usability of specific aspects of audio editing software. It is clear that although a number of visually impaired individuals are, through developing substantive technical skills and sheer persistent hard work, managing to operate effectively as professional, semi-professional and amateur musicians and audio producers or as teachers of these technologies, these numbers are low and reflect the high barrier to entry imposed by the inaccessible nature of current music studio technology. Reasons for the shortcomings of current technical solutions, and the consequent detailed requirements for improved solutions, are documented and analysed in section 3.3 of the report.

At the end of the requirements gathering phase, the management team felt that in order to obtain the depth of understanding that would lead to truly innovative solutions, it was necessary to undertake a detailed examination of the ways typical representative visually impaired musicians and audio producer's currently work, and examine in depth the problems they face using the technology. What we wanted to avoid above all, was falling into the trap of proposing, in the feasibility report, yet another incremental solution, which would improve accessibility at the margins, but fail to address fundamentally the gulf in usability and support for workflow that we had established exists between visually impaired users and their sighted counterparts (see section 3.3). We therefore undertook a case study approach, which examined in detail

the working methods of representative visually impaired users of audio technology. The methods employed to gather the information during the case studies were extended interviews and computer-based evaluation sessions in which the use of specific technologies for representative tasks were examined, along with alternative solutions. The results of this phase of the project are documented in section four of the report.

At the end of this stage we felt we had developed a deep understanding of the shortcomings of existing access solutions, and were able to identify a range of viable solutions that exploit state of the art interface technology to bring about a step change in the interaction paradigms and workflow processes of visually impaired musicians. These recommendations are documented in section five of the report, and essentially form the basis for future work on this project.

2.2 Costs

The costs of the project were very largely as we identified in our original proposal. The timings of activities were also largely as we anticipated, and are represented in a Gantt chart in appendix 1 of this report.

The requirements investigation cost 6,000 pounds for the time spent by Oussama Metatla in developing, disseminating and analyzing the results of the user experience survey, conducting interviews, reviewing current products and associated access solutions, examining e-mail list archives and undertaking practical computer-based evaluations.

The work on the case studies, including the extended interviews with the four participants and task-based computer evaluations of software and alternative access solutions, analysis of these findings and synthesis into a set of final recommendations, in addition to the writing up of the feasibility report came to a total of 6,500 pounds.

The costs for the audio production software, access screen-reader and associated hardware came to just under our original estimate of 2,000 pounds. The breakdown of these costs is as follows:

- Sony VGN-TT21JN laptop: £1216

- JAWS Professional upgrade from 8.0 to 10.0: £534.50
- Sonar Producer upgrade from 8.0 to 8.5: £89
- Yamaha Audiogram 6 USB sound module: £111.86
- GoldWave 5.52 Audio Studio software licence: £31.09.
- Total: £1982.45

This equipment was invaluable in the evaluation of existing products and the computer-based evaluations of specific issues in existing access solutions in the requirements investigation, and again in the task-centred evaluations of alternative access solutions during the case studies.

The original estimate for travel turned out to be rather an over estimate due to the ability for visually impaired people to travel largely free of charge on public transport. The total costs of travel have been 83.60 pounds.

2.3 Innovation

The major innovation in this feasibility study is the proposal of a radically new approach to the provision of access to music studio software for visually impaired users. Existing solutions suffer from the fact that they seek to enhance current interaction approaches such as screen-reader technology, which are inherently poorly suited to the task. The solutions we propose would provide access to objects completely ignored by current access solutions, such as audio waveforms and graphical objects within the music studio interface.

The key concepts underlying the proposed approach are 1) direct manipulation and 2) modal substitution.

Direct manipulation - By direct manipulation we mean the representation of objects in a form such that they can be directly manipulated by a visually impaired user in a way that very similar to their sighted counterparts. An example would be the direct representation of a waveform representing a given piece of music. For a sighted user, this might be represented visually on the screen, and may be manipulated typically by

the use of a mouse. In our accessible solution there would be an equivalent representation of the wave-form, made available to a visually impaired user through a haptic (touch-based) interface, where the tactile waveform could be manipulated in the same way as its on screen equivalent. This concept should be applied to numerous objects and controls found within digital music processing systems, which would help to address the overly linear nature of speech-based interaction and significantly address the problems of support for workflow documented in sections four and five of this report.

Modal substitution - By modal substitution we mean providing the ability to easily represent information in the music studio interface in an alternative form. For example, if you are currently listening to part of a recording, but at the same time need feedback from the system about your current position as you listen to the recording, it should be easy to switch the mode in which this feedback is given, such as to haptic or touch-based information, or even to keep the feedback in audio, but to locate it to the side of you so that it is not confused with the sound of the recording which is the primary source of interest.

This flexibility, particularly in the areas of haptic and advanced spatial audio interfaces, is made possible by recent developments in interfacing technology, which have largely yet to be exploited in interfaces generally, let alone specifically in the area of accessible music studio technology.

3 Review

In recent years, and with the advancement in audio input/ output technology and computing devices, there has been increasing interests and successful endeavours to bring the full “traditional” physical studio to the convenience of the personal computer. Digital computer-based audio and music studios include technologies that combine features to support tasks ranging from recording, editing and mixing audio, to song writing and producing full musical compositions. These are complex activities that were traditionally carried out by distinct individuals, but it is not uncommon nowadays to find that they are carried out by a single person who acts as a sound engineer and a musician at the same time. Modern technologies supporting the modern sound and music producer open up new possibilities for music creation and constantly reshape the climate of music and digital audio manipulation. The result is a growing market boosted with competitors, with an intimidating array of choices for sighted users when it comes to obtaining an audio and music editing tool.

It is virtually impossible to provide an exhaustive review of all that is available on the market. Rather, the aim of this section is to highlight existing approaches to supporting the music making and sound editing processes. These approaches are presented together with popular software on the market as an exemplification, in an attempt to provide a comprehensive list of product types. More importantly, this review takes a closer look at ways in which such mainstream approaches to music and sound production and composition have been made accessible to visually impaired users. The section concludes with reflections on the advantages and limitations of such accessibility solutions in light of how they compare to their visual counterparts available to sighted users.

3.1 Digital Audio Workstations

Digital Audio Workstations (DAWs) are environments that combine various technologies for capturing, editing and producing digital audio. The major aim and functionality of a modern DAW is to provide a representation of the digitally coded audio, and to enable operations on such representations. In order to do that, DAWs employ highly complex visual control interfaces that typically mimic traditional

“physical” artefacts and their interaction metaphors. For example, it is very common to find such interface features as virtual knobs and sliders resembling mixing desks (e.g. Figure 1 below), score sheets, as well as virtual instruments that are controlled using the computer mouse and keyboard and other external hardware, such as MIDI keyboards.



Figure 1 A visual interface controller mimicking physical interfaces - using knobs and sliders.

In addition, modern DAWs provide representations of other aspects of the manipulated sound that were not typical or even possible in traditional analogue studio settings. For instance, the use of a visual representation of a waveform to depict audio components is now an almost indispensable feature of DAWs. Such representation allows for a finer access and a detailed visualisation of the characteristic of an audio event as well as its parameters and how these progress over time. Furthermore, waveform representations are typically overlaid with other forms of graphics, such as graphs and diagrams, to allow for direct visual manipulation and transformation of certain aspects of audio parameters (e.g. see Figure 2 below).

Another important feature of DAWs is the provision of a visual representation for multiple objects at the same time. This gives the user a powerful means for controlling and manipulating multiple audio tracks simultaneously, and is usually achieved by enabling various parts of a single project to be superimposed. This provides a musician or an audio engineer with an intuitive way to visualise and overview the overall progress of their music making and sound editing process.

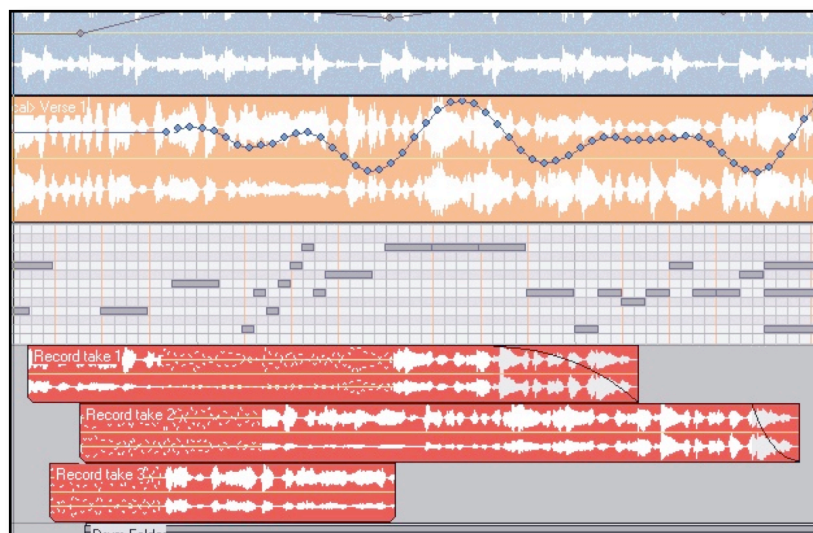


Figure 2 Waveform graphical representation typically used in DAWs, including an overlay graph like representation - on the orange waveform - to represent characteristic of the encoded audio. A user is able to drag points on the graph, which results in change of the audio output

Implementations of these features take several forms depending on the nature of a DAW and the tasks it is used for. DAWs can be dedicated to music and sound editing, live performances or both, although performance software could be considered as musical instruments themselves (e.g. Ableton **Live**¹) and are therefore not included in this review. But despite their diversity, DAWs can be broadly categorised into three major product types; Audio-intensive; MIDI-intensive; and Loop-intensive DAWs. Audio-intensive DAWs are primarily used for recording, editing and mixing audio and tend to provide sophisticated means for doing so. They can simultaneously handle a large number of stereo tracks at high resolution, and are particularly user friendly to music production. They may also include MIDI capabilities and other features of signal processing and compatibility with real-time audio transformation applications, such as Rewire. Digidesign's **Pro Tools**² (Figure 3) and Steinberg's **Nuendo**³ are good examples of this type of DAWs.

MIDI-intensive DAWs on the other hand are applications that were originally exclusively based on MIDI sequencers, but tend to include audio capabilities as well and are now further evolving towards supporting loop-oriented production. The combination of powerful MIDI sequencers and high-end audio capabilities make this

¹ <http://www.ableton.com/>

² <http://www.digidesign.com/>

³ http://www.steinberg.net/en/products/audiopostproduction_product/nuendo4.html

type of DAWs extremely powerful as they combine advantages of both worlds of audio and MIDI. Amongst popular example of MIDI-intensive DAWs are Emagic's **Logic Pro**, which is now an Apple product⁴, MOTU's **Digital Performer**⁵ and Cakewalk's **Sonar**⁶ (Figure 4).

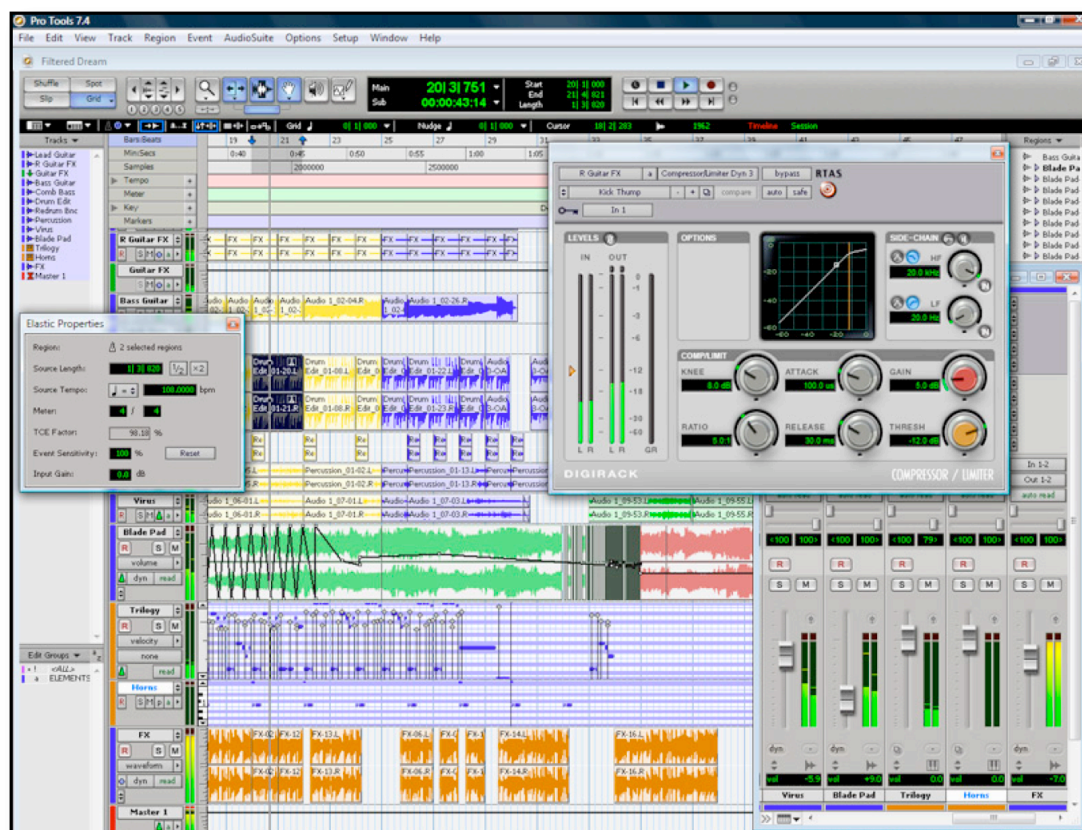


Figure 3 Screenshot of Digidesign's Pro Tools interface, an example of an Audio-intensive DAW.

And as the name implies, Loop-intensive DAWs primary handling of digital audio is supported through the form of loops. They typically include automation features for adjusting audio parameters and characteristics of sound exerts as they are added to a looping track, and are famous for making the process of music creation extremely easy. This of course contributes to their popularity on the market. The first major loop-intensive DAW product was Sony's **Acid Pro**⁷ (Figure 5), and other examples of this

⁴ <http://www.apple.com/logicstudio/>

⁵ <http://www.motu.com/products/software/dp/>

⁶ <http://www.cakewalk.com/Products/SOONAR/>

⁷ <http://www.sonycreativesoftware.com/products/acidfamily.asp>

type of application include Image-Line Software's **FL Studio**⁸ and Apple's famous **GarageBand**⁹.



Figure 4 Screen shot of Cakewalk's Sonar interface, an example of a MIDI-intensive DAW.

As the market for DAWs grows, there is growing demand for the integration of audio, MIDI and looping capabilities. The borderline between these broad categories is thus increasingly blurring, which makes their categorisation a hard task. It is much more common to find a myriad of features and functionalities straying from each of the three types of DAWs and gradually evolving or fully integrated into another.

There are however other types of applications dedicated to audio editing that can also be considered as digital audio workstations, but do not necessarily include capabilities for musical editing and production. For example, such DAWs would allow for the recording and manipulation of digital audio, but do not include MIDI or looping capabilities or any other features for musical production, such as mastering.

⁸ <http://flstudio.image-line.com/>

⁹ <http://www.apple.com/ilife/garageband/>

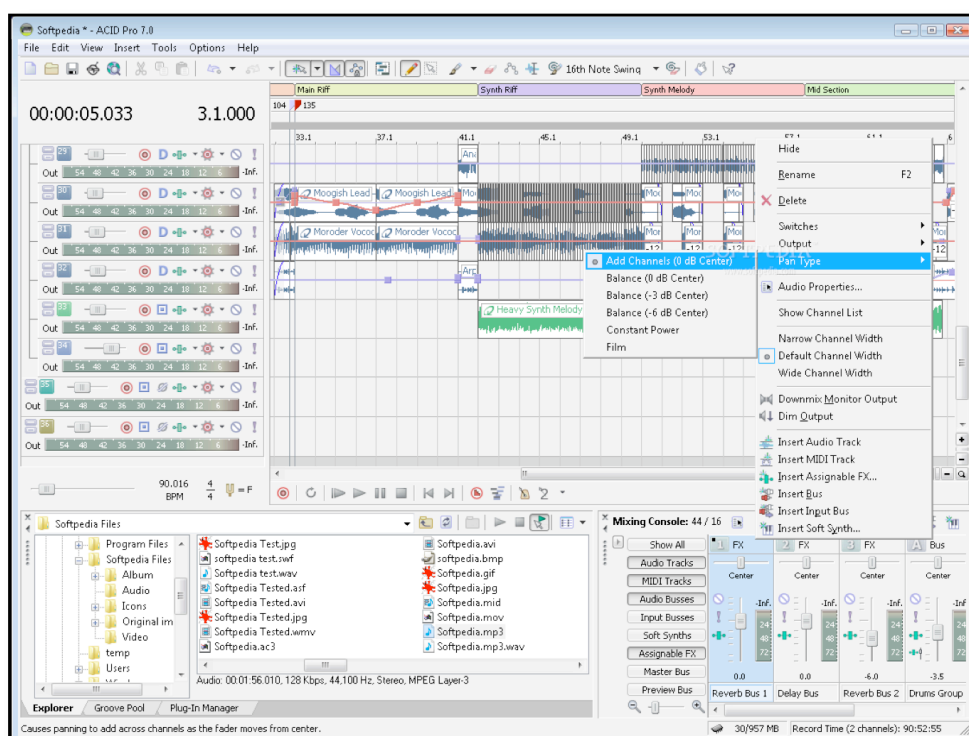


Figure 5 Screenshot of Sony's Acid Pro, an example of a Loop-intensive DAW.

Such DAWs could be used for non musical purposes (e.g. podcasting, telephone systems, audio analysis, etc.) and/ or in conjunction with more sophisticated DAWs to manipulate audio or edit individual samples to assist with the music making process. **GoldWave**¹⁰ (Figure 6) and **Audacity**¹¹ are good examples of such applications, where the user is presented with a visual representation of the audio to be manipulated, as well as various playback, effects and audio transformations controls.

Perhaps a further characteristic that can be used to distinguish types of music and audio editing tools is the nature of the graphical representations they employ to represent the audio and its controls. We have seen in the examples mentioned above how interface features of DAWs adopt visual metaphors of traditional analogue studio equipments and their physical properties (Knobs, Sliders, etc.), and how these are coupled with new forms of representing the coded audio (e.g. waveforms, graphs). This is intended to allow for finer access and intuitive visualisations of the audio editing process.

¹⁰ <http://www.goldwave.com/>

¹¹ <http://audacity.sourceforge.net/>

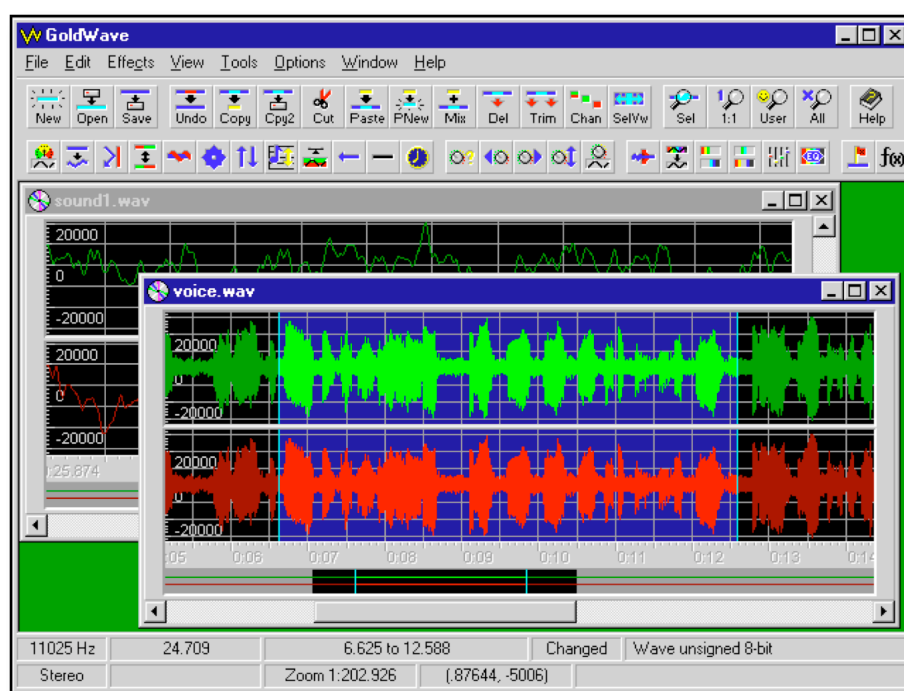


Figure 6 Screenshot of GlodWave audio editor.

However, a number of applications take a different approach to representing music and sound editing. As a music notation software, **Sibelius**¹² (Figure 7) for instance uses traditional sheet music as the main interface between the user, the sound and the music. Essentially, Sibelius provides a direct manipulation interface for creating and editing music notation scores. Users can create notes, drag them around and perform various editing actions directly on the score sheet representation. It also allows for the scores to be conducted using keyboard input with MIDI output. Sibelius can be linked to other digital audio workstations, such as Pro Tools, Steinberg CuBase or CakeWalk for audio and recording production, which makes it both attractive and powerful for complex editing and composition. Other examples of music and sound editing tools that focus on music notation include **Finale 2010**¹³ and MagicScore's **Maestro**¹⁴.

¹² <http://www.sibelius.com>

¹³ <http://www.finalemusic.com>

¹⁴ <http://www.musicaleditor.com/magicscore-music-notation-products.html>

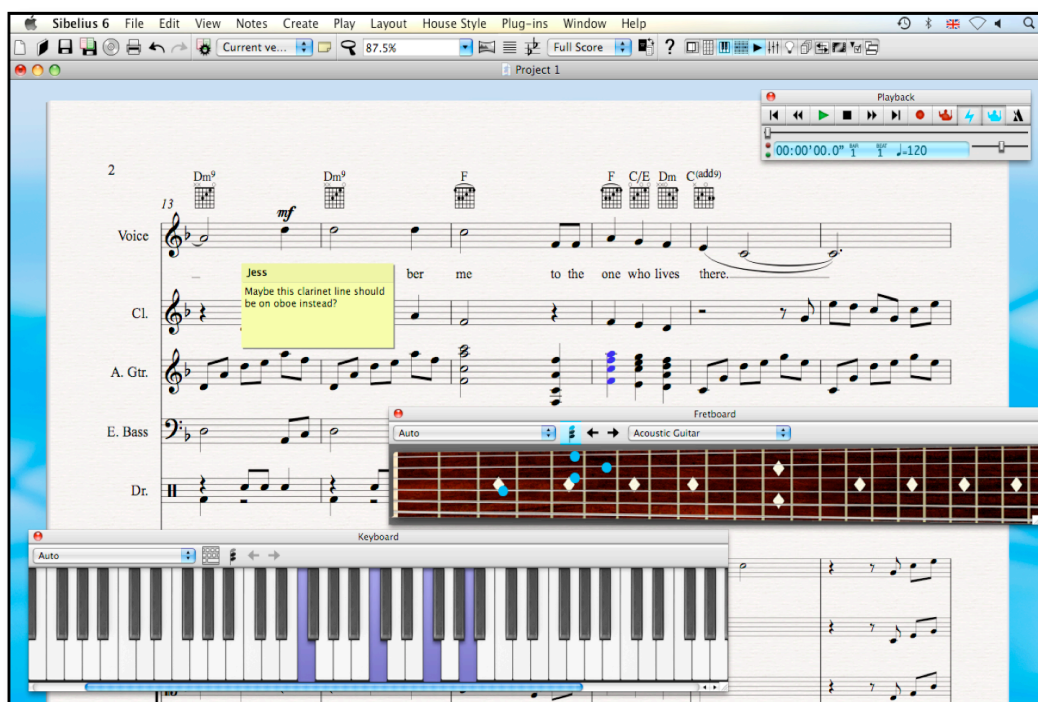


Figure 7 Screenshot of the Sibelius interface music notation software

3.2 DAWs and Accessibility

Whether it is an expressive music notation software or a powerful DAWs with analogue equipment-like representations, the heavy reliance on visual display metaphors in these applications render them essentially useless for blind and visually impaired musicians, teachers, students and audio engineers. In order to access features of mainstream DAWs, visually impaired users resort to either one of or a combination of two solutions, which are the primary accessibility solutions currently available to them:

- 1) Standard screen-reader software and/ or screen-reader scripts specially written to fit the functionalities of a given DAW.
- 2) Braille music notation and other specialised hardware.

3.2.1 Scripts for Screen-readers

One of the primary means used by visually impaired individuals to access and operate computers is screen-reader software. Essentially, a screen-reader gathers information sent to a computer's standard output and presents it to the user through synthesised

text-to-speech output or a Braille display.

Using standard screen-reader software, such as JAWS¹⁵ or Window-Eyes¹⁶, it is possible for a visually impaired user to access and operate computer-based music and sound editing tools. In general, the accessibility level of any given computer application could be determined by the amount of information that it sends to the standard output, and hence could be picked up by the screen-reader. Such amount of information differ from one application to another, and it is common to find that certain features of a given music and sound editing tool do not provide enough information about their content, and are therefore harder or impossible to operate without a screen display.

Some music and sound editing products, such as **SADiE**¹⁷ and **Sound Forge**¹⁸, explicitly enhance their accessibility by making sure that more and more relevant information is reported to the screen-reader software. But another efficient approach to overcome this issue is the use of screen-reader scripts targeting specific workflows for a given music and audio editing application. Pro Guide¹⁹ for example, has recently released a set of JAWS scripts that exploits the enhanced accessibility of SADiE to allow a visually impaired user to have improved access and control over its functions.

Screen-reader scripts are very popular amongst visually impaired computer users and, indeed, this popularity extends to music and sound editing tools. In addition to the commercially available scripts-based products such as **JSonar**²⁰ and **CakeTalking**²¹, the expert visually impaired user can use functions built in screen-readers to write their own scripts, and thus are able to tailor them to their own needs and working styles. JSonar is an open source project to provide a set of JAWS scripts for improving access to Cakewalk Sonar, and is accessible through various languages including German, Dutch, Swedish and Russian. CakeTalking, a product by Dancing Dots, is also a set of JAWS scripts for improving access to various Sonar third party plug-ins and synthesisers that are not natively accessible. In addition to spoken

¹⁵ <http://www.freedomscientific.com/products/fs/jaws-product-page.asp>

¹⁶ <http://www.gwmicro.com/Window-Eyes/>

¹⁷ <http://www.sadie.com>

¹⁸ <http://www.sonycreativesoftware.com/products/soundforgefamily.asp>

¹⁹ <http://www.proguide.eu/>

²⁰ <http://www.jsonar.org/>

²¹ <http://www.dancingdots.com/prodesc/CakeTalkingForSONAR.htm>

content, as visually impaired users interact with Sonar through specialised scripts, verbal prompts provide overview information about the available functionalities that can be access from their current position on the interface.

Dancing Dots also contributes **SibeliusSpeaking**, yet another set of JAWS scripts that make Sibelius music notation software accessible. However, later versions of Sibelius (beyond version 3 released in 2002) are no longer supported, and **Lime Aloud** is currently the recommended alternative as it provides verbal and musical output to support navigation and editing of music scores. Another popular set of screen-reader scripts that is no longer available is **outSPOKEN**. This compromised the accessibility of a powerful and ubiquitous DAW; Pro Tools, thus compromising its dedicated visually impaired users' ability to keep up with advancements introduced in its latest versions on the Apple's Mac OS X.

Those still regularly updated scripts-based solutions (e.g. JSonar, Caketalking, etc.) are even more powerful when used in conjunction with yet another set of scripts dedicated to replacing the convenience of mouse navigation; namely, **HotSpot Clicker**²². These are free screen-reader scripts that allow a user to define on-screen locations where mouse clicks need to be performed or content needs to be spoken. The user can then associate such locations with keyboard hot keys for quick access. This is a powerful means for not only allowing control over content and operations that are natively inaccessible, but also for making accessible ones operate much more efficiently and conveniently.

It is also important to note that various scripts for screen-readers can be operated using a Braille display as a control surface. The Braille display for JSonar for instance, allows for various operations to be accessible; navigating and manipulating tracks (select/ unselect); viewing the status of various functions; and playback. JSonar with Braille provides two display modes, where different functions are laid out on the Braille display to correspond to each mode; a Strip Display mode for accessing and manipulating tracks; and a Time Display mode for accessing the various timing (hour/minute/second) and measurements format (measure/beat/ticks) aspects of a project.

²² <http://www.hotspotclicker.org/>

3.2.2 Notation-Based Solutions

While Braille in the accessibility solutions described above is used as a means for displaying content and controlling features of a music and sound editing tool (e.g. JSonar with Braille display), it is also exploited in the form of Braille music notation technology. Braille music is a notational system that has been developed to translate traditional music notation into an accessible format, and is particularly popular amongst professional musicians, music educators and students (see for e.g. Figure 8).

Music notation is a different way to express music and interact with sounds, and some projects have focused specifically on enabling translation of such notation into Braille music. The **FreeDots**²³ project is an example of such effort, which essentially decodes digitally coded musical notation written in MusicXML format and translates it into Braille music (see Figure 9 below). Other examples include the **Braille Music Kit**²⁴ which also supports editing of Braille music and allow for conversion of print music through plug-ins for Finale 2010.

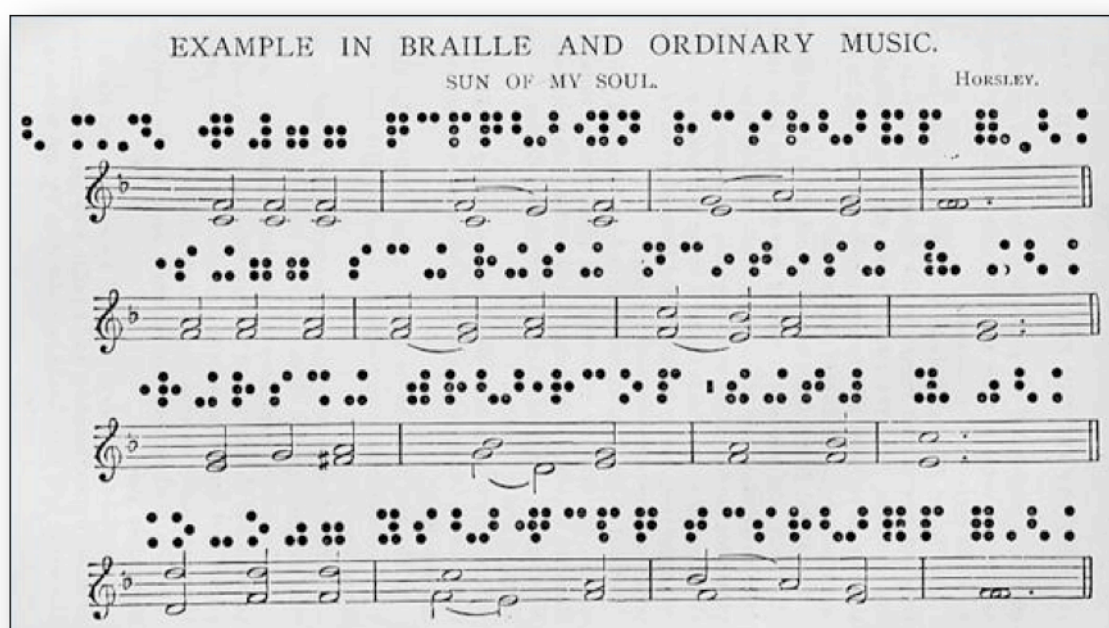


Figure 8 Traditional music notation and Braille music notation

As described in the previous section, there are various software applications available to sighted user to create and edit music score sheets on the computer (e.g. Sibelius,

²³ <http://delysid.org/freedots.html>

²⁴ http://www.dodiesis.com/index.php?q=bmk_en

Finale 2010, etc). Accessibility to such software have also been explored and developed to help blind musicians who are Braille music literate, and those who assist them through technology. Dancing Dots has developed solutions that mix mainstream software and assistive technology for scanning print music and automatically converting it into the equivalent Braille music, as well as for allowing a visually impaired musician to play and record their instrument through a microphone or a MIDI keyboard, and convert what is played into print notation for a sighted user to use. This can be done through the **GoodFeel**²⁵ Braille software, which automates the translation process that can be used in conjunction with Sibelius - through Sibelius Speaking. As mentioned above, Sibelius Speaking allows a visually impaired musician to produce music notation score sheets. GoodFeel could then be used to convert the produces score sheets to Braille music through a three steps process involving two extra software programs.

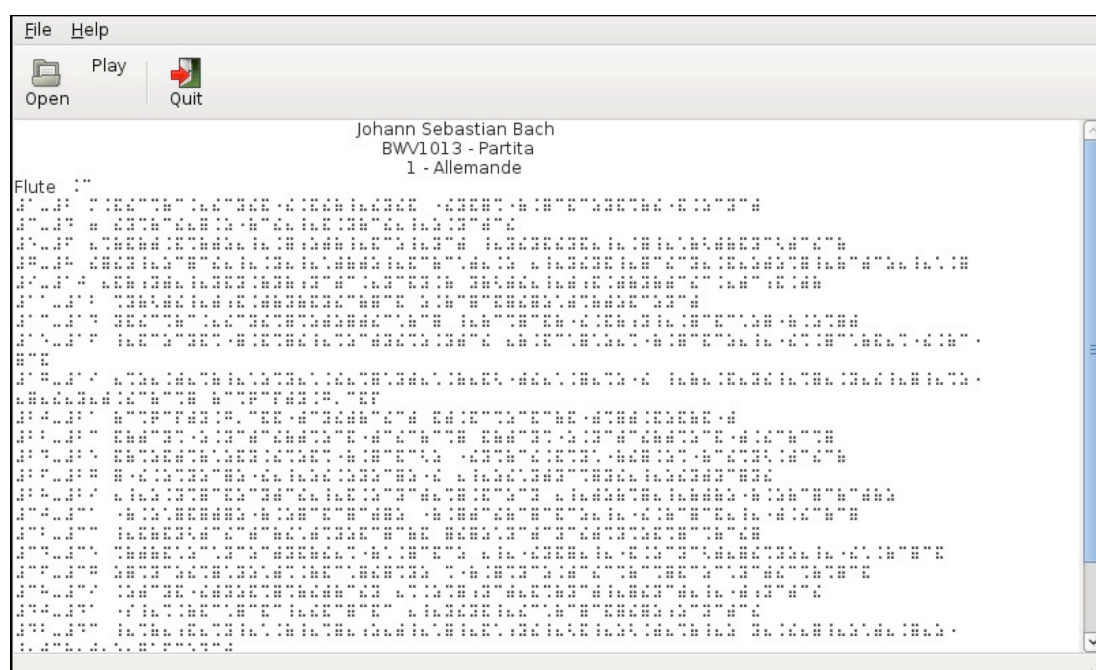


Figure 9 Screenshot of the FreeDots music notation to Braille music translation software

First, a music scanning software, called **SharpEye**, acquires and loads the image from a scanner. Notes are then recognised and an editable soft version of the printed notation is created. SharpEye recognition algorithms not only detects notes but also highlights warnings for potential problems with the recognition process and the way it

²⁵ <http://www.dancingdots.com/main/goodfeel.htm>

was interpreted from the scanned image. A sighted user can then correct all potential recognition mistakes. A playback feature is also included as a further proof reading method. The next step involves a music notation editor, called **Lime**, where the soft version of the scanned score can be loaded and annotated with appropriate labelling and formatting (such as naming the parts, right hand and left hand divisions, sheet title, etc). The next and final step involves GoodFeel which translate the edited score sheet into Braille music notation.

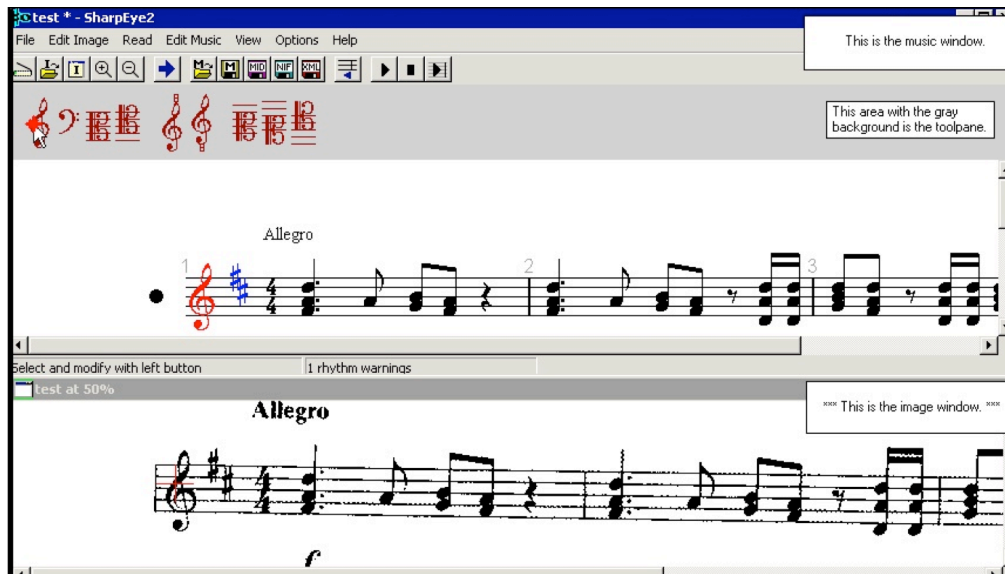


Figure 10 Screenshot of the SharpEye scanning and editing software from Dancing Dots

3.3 Reflections on Current Accessibility Solutions

It is clear from the above review that considerable efforts have been invested to make accessing and operating mainstream music and sound editing tools possible without a visual display. In particular, three distinct trends for achieving this are observed; software that work out of the box with standard screen-readers; scripted functionalities written for specific applications and packaged to address particular user needs; and two-ways conversion software to edit and translate music notation into Braille music and vice versa.

Default accessibility - For a given application to be accessible out of the box, accessibility considerations have to be taken on board during the design and implementation of its controls and functionalities. In this case, a reliance on the manufacturers themselves to account for such considerations is necessary. The

SADiE and Sound Forge software, and also GoldWave audio editor, have gained popularity amongst visually impaired users for this very reason; the ability to simply operate them out of the box. However, the reality remains the same; things do not “simply” work for all available products, and hence the rise of the need for scripting.

Script-based solutions - Products that provide sets of scripts for screen-readers make a great deal of functionality accessible and conveniently operated, but there are still problems that are encountered when this type of technology is used. The developers of JSonar for instance, have highlighted various issues related to navigation and orientation within the application, which are in part due to the windows graphical display metaphor of Sonar, and to the sequential nature of auditory presentation in another.

It is typical for computer users to have various windows open within the same project, or across different applications, switching back and forth between different windows as they execute corresponding tasks. Without a visual display, hidden parts of a project can be hard to locate with a screen-reader when they are out of focus, particularly so when focus information is not communicated to the screen-reader, thus requiring the user to manually restore focus in order to gain access to them. This can be disorienting and frustrating. Furthermore, functions such as moving a set of tracks or strips spatially, tend to have prerequisites in order to function properly, ones that the user may not necessarily be aware of. There is an evident need for alerting the user when such prerequisites are not satisfied, but surprisingly this feature is seldom available.

Locating tracks within large projects has also been reported to be sometimes problematic. The designers attribute this issue to the highly graphical nature of the Sonar interface. They have, however, developed a workaround by providing an alternative view mode, where some of the graphical clutter is removed in order to leave space for JSonar to easily locate and access parts of a project.

While some of these issues might be specific to JSonar, others are clearly a consequence of the medium of presentation (i.e. spoken output) and the model of interaction employed by this technology, and are thus experienced with other scripts-based products as well. The use of HotSpot Clicker to jump around between different actions and locations might mitigate against the issues of locating controls or content

within different parts of a project, but at the same time the sequential nature of presentation might still hinder users' ability to gain a holistic view of various aspects of a project. Another striking accessibility issue shared amongst the reviewed script-based solutions, is the complete lack of support for accessing purely graphical elements of the music and sound editing tools. We have seen how graphics, such as the waveform representation, graphs and diagrams, provide not only an intuitive overviewing feature, but also a fined vehicle for controlling audio parameters and characteristics. Yet there is no practical way for a visually impaired user to efficiently access such features using a screen-reader or a Braille display for that matter.

Braille display - Clearly, Braille software and hardware mentioned in the previous section provide certain level of autonomy for visually impaired users when it comes to handling music notation or gaining improved access to software features and controls. The ability to read, edit and convert music scores into Braille music, and vice versa, using GoodFeel for instance, supports collaboration between sighted and visually impaired musicians. When used in conjunction with script-based solutions, Braille display provides an alternative mode of interaction that might alleviate workload and improve access to particular features of a music editing tool.

Yet again we find that such technology has its limitations. For one thing, using Braille physical displays (i.e. Braille hardware devices) can consume a lot of CPU power, which is problematic when working on large projects that require considerable processing power. From a practical perspective, to be able to benefit from the functionalities of the GoodFeel approach, one has to know how to read Braille music in the first place. For the Braille music illiterate, this approach is therefore inherently useless. It is worth noting that although similar symbols are used, reading Braille does not imply ability to read Braille music. In fact, in many instances, the mappings used to express musical meaning in Braille music can be counter-intuitive. Aware of this potential barrier, Dancing Dots has developed extensive learning materials for teaching and studying Braille music, which could be obtained through their website.

A further limitation of the GoodFeel and similar Braille notation editing software is that they necessitate that a sighted user assist in the editing process. For example, when a music score sheet is scanned and loaded into SharpEye and Lime, the user is presented with a visual interface where they can edit and refine the interpreted soft version of the music sheet. This limits the autonomy of visually impaired users, who

would not only require assistance with this process from a sighted user, but from one who can actually read music notation, and so able to recognise and correct the errors detected by the software interpretation algorithms.

Usability & Accessibility - There are always going to be limitations with using interactive technology, and the challenge for interactive systems designers is to detect such limitations then to design interaction to reduce their impact or overcome them altogether. The question that poses itself with interactive accessibility solutions is whether they will ever fully compare to their visual counterparts. There is a lot of emphasis on using screen-reader technology and Braille display to make content and controls accessible and operational, but is accessibility really a matter of mere ability to read content or execute actions? Looking at the full visual interfaces, many features of the graphical display contribute to making the interaction flow and the execution of tasks usable. These features are completely lost when one moves from the visual to another modality of presentation. There is essentially a great deal of emphasis on accessibility but hardly any on the usability of the developed accessibility solutions themselves. To evaluate their products, developers usually release beta version of their product to a dedicated user list, and receive feedback on their use. This is a good way to gather knowledge about the appropriateness of a product, but might not account for long term usability.

Using audio to present information encompasses more possibilities than to simply display content in speech, and physical display techniques that are based on the sense of touch go beyond the use of pin arrays for Braille display. Combining of these techniques to provide alternative means of interaction requires a fuller understanding of user experience not just with technology but also with the objects of interaction themselves; in the context of this study, with the music and the sound itself.

4 Case Studies

In order to learn about user experiences with current music and sound editing accessibility solutions, we conducted a number of case studies involving visually impaired users of such technologies. The case studies focused on the use of screen-readers and Braille displays reviewed in the previous section, and closely examined advantages and limitations of the ways in which they support users in real life activities and settings.

We recruited four participants with differing expertise, backgrounds and occupations in order to gain diverse perspectives on the use of accessible music technology. Case studies involved interviewing users and, where possible, observing how they go about executing music and sound editing tasks using their technologies of choice. We interviewed a professional musician, an accessibility trainer, an audio producer, and an educator and hobbyist. Where participants gave consent, interviews were recorded and data was later reviewed and analysed in order to extract issues specific to the usage of the technology. In addition, participants were encouraged to describe potential alternatives and improvements on currently available solutions. Such knowledge was then used to structure and categorise usability issues typically encountered by our volunteer visually impaired users.

We have extensively discussed as well as observed how these users go about their daily activities involving music and sound editing technology, but the amount of data gathered from each case study that is of relevance to this report varied between one participant and another. We have therefore presented each case independently to highlight these differences and make their contributions and the issues they bring out more explicit. The names of the participants were intentionally omitted from the data in order for them to remain anonymous. The details and outcome of each case study are presented in this section, together with a summary and a categorisation of the type of usability issues typically encountered with using music and sound editing accessibility technology. Appendix 2 of this report presents details about the content of the questions used during the case studies in order to structure the interviews and the observations.

4.1 Case 1 - Professional Musician

Participant - Mr. S.H.

Background - S.H. is a professional musician, having practiced music since the age of Seven, and doing so for forty-two years. He first learned classical piano, up to grade 8, and now writes both classical and pop music. He is part of a two person band, playing piano, keyboards, guitar and bass, and performing regular classical and pop concerts. He has been using recording technology, both traditional and modern, for about thirty years.

Technology - S.H. has been using Cakewalk Sonar as his main music studio software for the past ten years, in addition to using Sound Forge for audio editing purposes. He previously used a Roland VS1680 digital audio workstation as his main recording tool, but moved to software solutions as soon as he could. He is now using Window-Eyes screen-reader technology to access and operate both of Sonar and Sound Forge. He now uses the VS1680 for mixing purposes only, connecting it up to a computer, which runs the audio and music editing tools.

Activities - As a professional musician, S.H. primary use of the combination of technologies described above is for musical purposes; recording, editing and mixing audio as he writes and composes his music.

User Experience - The Ronald VS1689 is a physical device that is totally inaccessible. To use it, one has to literally remember the positions of the various sliders and buttons on the device as well as what each one is used for. As soon as software technology was available, S.H. switched to CakeWalk Sonar with Window-Eyes screen-reader. This is now his preferred technology for music and sound editing, having previously tried to use Steinberg's CuBase but found it totally inaccessible with his choice of screen-reader software.

Operating Sonar with Window-Eyes is straightforward, explains S.H. As you navigate through the Sonar interface, the screen-reader speaks out the names of the controls and the objects it encounters. It also allows access to all menu items, if for instance one is not familiar with the keyboard shortcuts that are specific to Sonar. In order to explore and navigate the content of tracks, i.e. the audio recordings themselves, S.H. resorts to listening to the audio using the playback functionalities at normal pace. He

combines this with the built-in markers facility, with which it is possible to place markers at particular points on an audio track, then jump to these parts as he requires when exploring and navigating within the audio tracks space. This is all done with the default functionality of the screen-reader and without any script-based additions, although for Window-Eyes to function properly with Sonar, some of its parameters have to be set to appropriate values in its initialisation file. Sound Forge is also very accessible by default, and allows various audio editing tasks to be performed, such as cutting gaps in audio recordings, increasing overall volume, and so on.

Typically, S.H. works on single tracks, one at a time, rather than on multiple tracks at the same time. It is common for him to record various parts of a single instrument on different tracks, and then, once edited to a desired point, merges them together to form one stereo track. This involves collecting all relevant tracks from the track space, and using the bounce function from the available menu, or through a keyboard shortcut. This is also useful when he works on large projects, which in his case sometimes include up to 80 tracks. Processing power might be jeopardised due to the number of tracks, and so he tends to merge those finalised into a single track, which frees the computer resources and allows for the addition of further tracks.

When working on projects with large numbers of tracks, S.H. makes sure that he uses a good naming convention, which helps him easily navigate and locate tracks of interest. Nonetheless, with very large projects, it would sometimes take him a while to locate tracks, particularly when going back to projects that were created sometime ago. He has not tried to use an ordering convention in conjunction with a naming convention; e.g. grouping a number of tracks together hierarchically, and this is mainly because ordering tracks usually depends on the current song or the project he has at hand.

In general S.H. is happy with using Window-Eyes with Sonar and Sound Forge and has developed his own working style with this technology, and therefore has not so far felt the need to switch to another music editing application nor another screen-reader software.

Issues - The first issue with using Window-Eyes to access Sonar and Sound Forge that S.H. described is related to navigating and exploring the content of an audio track. Currently, the way S.H. does this is through a combination of audio playback

and Sonar built-in markers facility, and although this is not causing him a huge problem, it is sometimes a time consuming process. He would prefer being able to explore an audio track more efficiently through something like a scrubbing feature, which allows a user to fast forward or rewind an audio recording while still listening to its content at different speeds. Such feature is currently not available in his choice of technology.

A more crucial issue that S.H. encounters when using Window-Eyes and Sonar to record instruments' parts, is related to what is known as "punch-in" and "punch-out" recording. Essentially, a musician would sometimes record over parts of a track, if for example they might have hit a wrong note, or decided to change a particular part. A punch-in in this case is when the musician comes in with their instrument as the track is playing to override a bar or more, and a punch-out is when the overriding halts.

In order to do this, S.H. currently uses Sonar's "Go To" function. This involves four different actions: 1) listening and navigating through the audio track to locate the exact time where he wishes to punch-in, 2) copy the time value into the clip board, 3) going to the punch-in menu item in Sonar, and 4) pasting the time value in the provided input field. The same process is needed to set the timing for a punch-out, and once both these times are set, the musician can then play the audio track again and record the desired part when prompted. This process is clearly cumbersome, and a feature to allow the punch-in and punch-out to be done through fewer steps would significantly improve on the efficiency of such activity.

Another issue that somewhat extends from the one just described, is related to recording guitar or bass parts - or any other instrument of similar physical characteristics. S.H. described how switching between controlling the Sonar interface with the computer keyboard, for example, pressing the record button, then quickly swinging back to a comfortable position to play the instrument can sometimes be awkward.

Adding alterations to audio parameters of a given track while it is playing, and how this is supported by the Sonar/Window-Eyes combination was also highlighted as problematic. When arranging instruments parts, it is common to adjust the volume parameter to, for example, gradually increase or decrease the volume of an instrument at a given moment in a song. Usually, such manipulations are executed while the

targeted track is playing in order to precisely achieve the desired effect. The ability to do this without pausing the playback is crucial to the successful execution of this task. The way S.H. does this with Window-Eyes is through Sonar's "Automation" function; to adjust an audio parameter of an instrument at a particular moment of the recording, Sonar must be set to the "auto-record" mode so as to remember the changes in parameters as they occur. S.H. then has to trigger the parameter changes at the desired point in time and adjust it as required.

Because the parameter adjustment is done and displayed through speech output, this process is not only cumbersome with regards to the amount of actions and precision by which it needs to be executed, but also poses huge burden on the musician's attention, who needs to be able to listen to both the speech and music outputs at the same time.

Summary - S.H. has been using Sonar with Window-Eyes for over ten years, and has developed an efficient working style to compose, record and arrange his music through this technology. But while able to achieve the desired product, S.H. has described four main issues related to the efficiency by which his technology of choice supports his music creation process. Three out of four issues are directly related to the temporal dynamic nature of interaction with audio, through an essentially auditory interface; being able to navigate and explore an audio track without pausing to: 1) detect parts of interest and 2) record over particular parts, and 3) alter audio parameters. The other issue was related to the nature of the arrangements between manipulating Sonar through a screen-reader and manipulating other physical instruments within the same setting.

4.2 Case 2 - Accessibility Trainer

Participant - Mr. P.B.

Background - P.B. is an accessibility trainer and a musician with extensive experience with assistive technology both as user and consultant. He runs his own company that provides accessibility solutions and training for visually impaired individuals on the use of screen-readers and Braille displays in everyday activities. He has been using a wide range of audio mixing and editing access technology for

nineteen years.

Technology - P.B. uses a combination of Sonar and SoundForge with a the Tascam FW1884 digital audio workstation control surface and audio/MIDI interface. In addition, he uses other outboard equipment, notably the Yamaha Motif ES keyboard, although he is increasingly relying on software rather than outboard hardware. He uses Caketalking screen-reader scripts for JAWS to access Sonar, and JAWS to access Sound Forge.

Activities - With an extensive experience, P.B. uses assistive technologies for a variety of purposes. Of relevance to this study are his audio production of music and speech, including audio drama activities, and musical arrangement and composition.

User experience - P.B. describes how he is able to achieve the main tasks involved in recording an audio project, i.e. recording acoustic and electronically generated music, mixing down the tracks, and then preparing a stereo master. He has previously relied on hardware equipment to carry out these tasks, such as multi-track tape and analogue mixers. But with the increasing development in accessibility technology, and particularly JAWS and scripting, he turned to using software solutions. P.B finds that JAWS scripts allow for various functionalities to be access and operated much more efficiently than the default screen-reader operations.

Issues - The main difficulties that P.B. highlighted with using JAWS scripts for Sonar and Sound Forge are related to 1) the pace of executing music and sound editing tasks, and 2) the ability to access the full available functionalities of mainstream software. Using speech to access an essentially visual interface makes for slower working. Adding to that is the fact that sometimes the access technology itself does not behave as it should, and so P.B finds him self having to invest extra time and energy to try and find solutions by re-configuring scripts, or waiting for bug-fixes to be released. Furthermore, P.B. explained that there are parts of the software still not at all accessible. For example, non-standard plug-ins to Sonar, which are useful for achieving particular effects or functions rely heavily on complex graphical interfaces, and most of these have not been scripted yet. He also highlighted issues related to studio desks hardware accessibility that also rely on visual control interfaces, for which no access solutions exist. In general terms, P.B. would like to experience interfaces with more reliability, and more intuitive means for interaction.

Summary - P.B. uses a range of accessibility solutions to mainstream music and sound editing software, and is a keen user and developer of script-based screen-reader technology. He described two important issues related to the usability of current accessibility solutions: 1) extensive reliance on speech output to access visually dense and graphically complex interfaces slow the pace of interaction, and 2) the unavailability of intuitive translations of purely graphical elements, in both software and hardware control interface.

4.3 Case 3 - Audio Producer

Participant - Mr. T.M.

Background - T.M. is an audio producer and a practicing musician. He studied music at an early age, and then went on to study sound recording and music technology at university. He has worked at the Royal National Institute for the Blind (RNIB)'s Talking Books Studios for the last nine years, and is currently an audio producer. T.M. used to have his own music recording studio, and currently plays piano and keyboards and performs regular gigs with his band.

Technology - T.M. Has used stand alone Roland VS digital audio workstation and other analogue studio equipment, but switched to using software studio recordings as soon as these were available. He now uses SADiE as his main recording and editing software at work, accessing it through the JAWS screen-reader in addition to various JAWS scripts, including ones written by himself. T.M. has also used various tactile displays, such as Opticon and haptic devices for monitoring audio parameters.

Activities - T.M.'s activities that are of relevance to this case study are recording and production of audio books. He also does additional music and sound editing activities but these are unrelated to his current job at the RNIB.

User experience - When using the physical Roland VS studio, T.M. relied on the sense of touch to remember and operate the controls that are on the surface of the device. Amongst the functions that he found most useful with this device is the scrubbing feature; this was because such function uses a buffer to break down and store the audio, which allowed T.M. access to its sound spectrum and hence to navigate the audio at finer details.

Having switched to software solutions, T.M. is currently using the SADiE audio editing environment, which in addition to the visual on-screen display, includes a physical control interface with a fully functional Jog Wheel controller. For T.m., the physical device is the easiest and most important way of interacting with the audio itself. Not only is the jog wheel used for scrubbing through an audio track in an almost similar way to the Roland VS, but it also includes other control functionalities, such as playback, and therefore feels that it presents him with an intuitive mapping relation between the physical control and the audio. He finds this particularly useful for the type of audio production work he is currently doing, which involves handling very long tracks of recorded speech.

Typically, T.M. works on single tracks, one at a time, rather than on multiple tracks at the same time. Using the physical wheel-based device in conjunction with screen-reader software, T.M. is able to locate parts of interest on an audio track, and to issue editing commands using shortcut keys that trigger SADiE functionalities for altering the audio and its parameters. This also includes adding markers at different locations on the audio track that he can jump to later on in the task. To control SADiE, T.M. uses JAWS as his primary screen-reader software with scripts, and has also written his own scripts to grab on-screen information that are particularly relevant to his production working style.

In order to monitor audio parameters while recording in the studio or handling tracks, T.M. uses tactile PPMs (Peak Program Meters). For instance, to monitor amplitude levels of a part track or segment of a track, T.M. would position his fingers on four vibrating sensors, each corresponding to a particular level of amplitude. Feeling vibrations on the first finger would thus map to the audio reaching the first level of amplitude; feeling both the first and second, would correspond to the second higher level, and so on. T.M. finds this to be an intuitive way for accessing such aspects of the recorded audio.

T.M. has also used a refreshable Braille display as a control surface for a visual studio software. Through this type of Braille display, T.M. could also navigate through audio tracks as well as control the software features. He described how such displays are similar to a physical mixing desk but one that only shows one row at a time.

Issues - T.M. described various issues related to using tactile displays and screen-

readers for audio production activities. While the tactile physical displays that he uses to control sound editing software provide him with ways to input commands to the system, most of such technology currently has little to no means for providing meaningful tactile feedback to the user. T.M. highlights that the only tactile displays which do provide a form of tactile feedback are Braille displays, but describes how such technology has limited resolution with typically only 3 by 2 to 4 by 2 matrices of pins for representing information. The Opticon device is an exception to this as it displays a 4 or 6 by 25 matrix of dots, but, according to T.M., this has still its limitations, and much more expressive means are required for his audio production needs.

With regards to using screen-reader software, T.M. described how JAWS speech output often interferes with his audio production activities, particularly so while recording narrators in the studio, who usually do not like to be stopped while reading a book. Having the screen-reader output playing at the same time as the narrator readings can be overwhelming, and so providing means for off loading the pressure from the ear and into another modality would significantly improve on this activity.

It is often the case that parts of a software package do not provide enough information about their content and are hence inaccessible by a screen-reader. But T.M. described how sometimes there is rather too much information that is sent to the screen-reader, which tend to be redundant or irrelevant to the current task. This often occurs when the JAWS software refreshes its buffer and results in it “spitting” a set of spoken output, which T.M. finds very off putting, particularly if this occurs while he is performing audio editing tasks. Redundant information is also sometimes spoken if the mouse or a keyboard key is touched by mistake. To avoid these inconveniences, T.M. usually resorts to setting JAWS so that it sends its content on the Braille display instead, or to writing special scripts that would mute the speech until relevant information is encountered.

In relation to script-based solutions, T.M. finds that they sometimes limit the way certain actions are executed to specific workflows. And while they might make task execution convenient, these workflows might not necessarily match his working style. T.M. therefore resorts to writing his own JAWS scripts, or in some cases coordinate with script-based product developers to produce specialised scripts for his needs.

Summary - As an audio producer, T.M. experience with sound and music editing accessibility solutions highlights a different perspective on the usability of such technology. One of the most important aspects of his interaction is the need for multimodal means of controlling and accessing audio. According to T.M.'s account, the combination of tactile and audio input and output are particularly important for two reasons: 1) providing an intuitive mapping relation between the audio and the means by which it is manipulated to facilitate the conceptualisation of the process, and 2) minimising cognitive workload by distributing information about control and display over multiple modalities.

4.4 Case 4 - Educator & Hobbyist

Participant - Dr. A.S.

Background - A.S. rates his expertise with using music and sound editing accessibility solutions as beginner to intermediate. He has used sound editing software for about five years, but not intensively. He has over twenty five years of experience with assistive technology in general, both as a user, an educator and a consultant. He is also a keen developer of audio-based games and auditory interfaces.

Technology - A.S. uses JAWS as his primary screen-reader software to access computer based applications. In terms of music and sound editing software, he uses a combination of Sound Forge, Audacity, GoldWave and Total Recorder. In addition to software applications, A.S. uses programming languages such as Csound and SuperCollider for creating and synthesising audio.

Activities - As an educator, A.S. primary use of the above combination of technologies is for recording and editing lectures, as well as editing sound files for use in audio games or other auditory interfaces. He occasionally uses such software for music editing.

User experience - A.S. finds Sound Forge the easier to use as he is able to operate it straight out of the box. In Sound Forge, A.S. explains, navigating around sound files employs the same key combinations that are commonly used to navigate documents in word processors; control-end to go to the end of the file; and page up/down and

cursor left/right to navigate through files. In order to edit segments of a given recording, A.S. uses the “Go to time” function, which allows him to jump to a specific point in a file, he then uses available functions from the edit menu for highlighting the segments of interest. Regardless of its accuracy, the main issue that A.S. highlights with this method is the speed by which it is executed.

To synthesise audio parameters of a particular audio file, or an audio segment, A.S. uses Sound Forge’s “Synthesis” function, which he finds to be both easily accessed via pull down menus for different options, and easily operated; having selected the required option, A.S. describes that it is only a matter of tabbing through to fix specific settings to be synthesised, such as the pitch and timbre.

Using GoldWave is also easy as many of its commands are triggered using keyboard combinations. For example, GoldWave uses the function keys to simulate the typical buttons on a cassette recorder, i.e. record, play/pause, rewind, forward wind and stop. His experience with Audacity on the other hand has not been as straightforward as was the case with GoldWave and Sound Forge.

A.S. is also a keen programmer and developer, using the Csound and Supercollider programming languages to synthesise sounds for audio games or for use in auditory interfaces. He finds Csound rather easier to use because it includes intuitive programming concepts, such as the “Orchestra” and “Score” abstractions. The object-oriented style of Supercollider, which tends to structure projects over several different code fragments for different parts of a system, is on the other hand slightly harder to follow.

Issues - A.S. referred to a number of activities that he finds difficult or impossible to achieve when interacting with audio editors using a screen-reader. The first issue that he mentioned is related to executing actions or controlling features that involve non-labeled graphics, and are therefore difficult to pick up by a screen-reader. These also include situations and objects where the screen-reader cannot be scripted to recognise the graphic. An example of this, explains A.S., is the sound equaliser in Audacity, which is currently totally inaccessible due to its heavy reliance on graphical representation. A related issue is the ability to execute audio control manipulations that are usually supported through graphical means, such as aligning the end of an audio track with the beginning of another.

A.S. has also highlighted issues related to the slow pace of accessing and operating graphical displays through a screen-reader, for example, in situations where he needed to retrieve current mouse cursor position. While it is possible to identify cursor position within a given file in Sound Forge, the process of doing so is slower than it is for a sighted user, who can simply look at the screen to retrieve the value of interest. The way A.S. does this at the moment is by issuing the key combination for the “Jump to time” command, then, instead of pressing enter to execute it, he examines the default time value, which usually defaults to the current position on the file. This might be an effective workaround for the issue of retrieving one’s position on an audio track, but providing a more efficient non-visual means for acquiring such information would significantly improve the usability of this task.

Sound Forge’s “Preview” function essentially allows for automatic playback of sound files as a user browses through their directory to select a file to open. This is another feature that A.S. finds to be sometimes problematic. While useful in some instances, A.S. described how irritating it is to have to listen to the audio content of sound files in situations where he is only interested in hearing their names spoken by the screen-reader. This is because speech output in such instances is easily masked by the auditory previews. In this case, providing the user with the ability to choose the nature of the previews they desire, for example, through a toggle function that switches between spoken output or played back audio, would significantly increase a user’s sense of control over the interface.

In relation to the programming languages that A.S. is currently using with screen-reader software, SuperCollider’s style and syntax pose some issues with navigation and orientation. Although essentially text-based, the SuperCollider language relies heavily on nested brackets for structuring and organising code fragments. While the programming environment contains good visual mechanisms to assist with balancing brackets, no non-visual counter part exists. A.S. resorts to keeping track of such nestings in his own memory, which can be quite demanding, and sometimes impossible, if the level of nesting increases to large numbers.

Summary - A.S. has relatively little experience with music and sound editing software, but has extensive experience with using and developing assistive technology. He tends to switch between various applications to gain maximum accessibility by matching the task at hand with the most suitable technology. He uses

Sound Forge for detailed audio editing, GoldWave for file conversions and Total Recorder for quick recording from the internet. A.S. experience with accessibility solutions to music and sound editing has highlighted three important usability related issues: 1) executing and manipulation operations that are of spatial nature through an audio based interface; 2) availability of appropriate means for providing feedback to users while engaged in such tasks, and 3) the pace of working and speed of task execution. Clearly, all three issues are interrelated, and addressing them effectively would therefore require a design approach that closely considers their impact on one another.

4.5 Usability Considerations and Categorisation

The presented case studies have provided detailed accounts of visually impaired users experience with technologies for accessing and manipulating sound and music editing tools. Participants in the case studies used various combinations of technologies for diverse activities ranging from musical composition to training and audio production. This provided us with interesting and informative perspectives of user experiences with accessibility technology for sound and music editing, and from diverse angles. The dominant technologies used by our case studies participants were screen-reader software (e.g. Window-Eyes, JAWS), scripts for screen-readers (e.g. CakeTalking for Sonar) and devices with tactile and physical controls (e.g. Optican and SADiE's jog wheel). Additionally, three out of four participants used physical digital audio workstations (e.g. Roland VS, and a Tascam DAWs).

While these combinations of technologies supported our volunteer participants in carrying out their music and sound editing activities, each case study revealed various limitations related to their usability. We collated the outcomes of the interviews and observations across the usability issues highlighted by each case study, and organised them into a categorisation scheme. Three main categories of usability issues emerged: 1) Issues related to the pace and the structure of interaction when using screen-readers as the main means of interaction; 2) issues related to manipulating spatial elements and spatially oriented actions through a temporal medium; and 3) issues related to cognitive overload. Next, we briefly elaborate on each category and use examples from our observations to highlight it essence.

Pace and structure of interaction - Usability issues in this category are attributed to the presence of a mismatch between the nature of a given task and the means by which it is executed. In general, interacting with and through speech requires considerable time. This is partly due to the fact that spoken output requires time to be uttered. And while it would seem that speeding up the speech rate of screen-readers would simply solve this issue, there actually are more complex elements underlining this problem. We observed a more significant cause of the slow and cumbersome nature of interacting with music and sound editing tools using a screen-reader, and that is the structure of the interaction itself. Visually impaired users described having to go through an inadequate, unrelated number of interactive steps in order to execute a given action. An example of this is S.H.'s experience with setting punch-in and punch-out timings while recording over existing tracks, and A.S.' effort to retrieve on-screen cursor locations.

In such cases, the issue is not with how long it takes the screen-reader to utter speech, but a direct result of a mismatch between the task at hand and the means by which it is executed. In S.H. case, triggering the system to record and then to stop recording at various desired points on an audio track should be achieved through a different means than having to first retrieve values of interest from the track, locating input fields on the screen, then feeding this information to the system. These actions are completely unrelated to the core objective in mind; that of recording a (physical) instrument part into a song.

In the case of A.S., there is also a clear mismatch between the nature of the information that is required and the structure of the actions by which it is retrieved. The user had to trigger a sequence of actions that are unrelated to the main objective (i.e. retrieving the position of the cursor by selecting the "jump to" function). Such information could be communicated to the user through more effective non-visual means. Since the position of a cursor is essentially a spatial piece of information, and should therefore be retrieved and delivered through a means that somehow captures such spatiality.

Spatial manipulations through a temporal medium - While we attributed the previous category of usability issues to mismatches between the nature of a task and the means by which it is executed, issues in this category are attributed to a mismatch between the nature of the information that form parts of a task, and the

means by which it is delivered to the user.

Problems of accessing purely graphical elements of an interface were explicitly highlighted in all of the four case studies documented. These included both problems with accessing complex graphical parts of a tool, and executing actions that are typically designed to be achieved through graphical manipulation of information. Both P.B. and A.S. have expressed how inadequate, or completely impossible it is to access and operate areas of an interface where heavy graphics, such as graphs and diagrams, are employed. For example, when attempting to access Audacity's equaliser, no appropriate information is sent to the screen-reader, and no alternative representation is provided as means for navigating the graphically represented information. A.S. has also highlighted the lack of feedback information when aligning audio tracks, which, visually, would be a simple task to execute. Feedback information is either not delivered or provided through spoken output of absolute values.

While accessing graphically and spatially laid out information is sometimes supported through tactile devices, such as Braille display modes, T.M. has described how such technology currently provides poorly expressive means for doing so. Current tactile technologies seem to be badly designed for capturing the richness of the represented information. A simple glimpse at a waveform representation for instance, often overlaid with a graph or a diagram, would give a sighted users a rich set of information about various aspects of the audio and the parameters it encodes. This is totally lost when the same information is simply spoken out or scrubbed through using a jog wheel. Using more intuitive multimodal means for capturing and delivering representations is therefore an attractive alternative. This is particularly useful for visually impaired users who have experience manipulating physical analogue equipment, since they operated such devices through a spatial mental model of the layout of its controls. Thus focusing on allowing a transformation of such knowledge by delivering spatial feedback through tactile and spatial audio is a promising avenue for examining how to improve usability.

Information overload - Manipulating audio through auditory means is likely to result in inadequacies both in terms of perception and interaction. When users had to listen to screen-readers as they speak out the names of objects encountered on the screen, while at the same time manipulating and thinking about their sounds and music, their concentration was disturbed and exhausted. This was a recurring problem observed

across all the four case studies. A striking illustration of this is T.M.'s audio books production work, in which narrators' reading clashes with screen-reader speech during recording, causing an overwhelming feeling of information overload. This is of course also the case when recording or manipulating instrument parts, or effects, where some form of audio masking regardless of its level of impact, will inevitably occur.

While in some cases, information must be delivered through speech, T.M. has described how off loading parts of the information to another modality would significantly improve on the quality of his production and recording activity. T.M. has indeed attempted to alleviate this problem by sending screen-reader output to a Braille display rather than the text-to-speech engine. But in his case, redirecting the information to Braille came with a cost; that of losing the richness and immediacy of control.

Multimodal interaction techniques are a very promising alternative for managing interaction and capturing the richness of representation and control. However, distributing information and control across multiple modalities requires careful design considerations that must take into account both the pros and cons of each modality, and the effects of their combination.

Other issues - The categories of usability issues described above have covered recurring problems across case studies, particularly those highlighted as critical by our participants. There are a number of other issues that were also described in the case studies but they do not fit within any of the above categories, or could be solved through technical implementation and development rather than usability considerations. For example, improving the "preview" function with a toggle feature as mentioned by A.S.; managing the extra redundant information sent the screen-readers as described by T.M.; or providing a more efficient scrubbing output feature to display finer audio parameter details.

In addition, T.M. had highlighted a very important limitation of using script-based solutions for screen-readers. While they make the execution of certain tasks convenient, scripts restrict interaction possibilities to a limited set of workflows as envisaged by its developer, and such workflows may not always match the diversity of user working styles. In addition, diversity of workflows might lead to inconsistent

scripted behaviour, since most scripts are based on particular screen layout and/or screen resolution. These sorts of limitations should be addressed through technical development to improve scripting techniques and implementations.

5 Informing The Design - *AMuST*

This section explores design elements of accessible music and sound editing. In particular, non-visual interaction techniques are examined, and the ways in which they can be employed to improve the usability of operating music and sound editing tools using accessibility technology are presented. The design of these techniques is informed by 1) the categorisation of usability issues that emerged from the case studies presented in the previous section, and 2) an examination of current research-based approaches to non-visual human-computer interaction (HCI).

All four participants in the case studies use screen-reader technology as the main means to interact with visual interfaces. Braille display was also used by at least one participant. Even though a growing body of research is providing increasing evidence of the potential of using audio and haptics to convey information in a variety of ways, the explicit inclusion of such techniques in commercial assistive technologies remains limited. For instance, non-speech sound was only recently incorporated in latest screen-reader software releases, such as JAWS version 5. In practice, this means that users tend to rely on the existing default speech-only and Braille output of their screen-readers, and so continue to encounter many of the problems associated with accessing graphically dense interfaces through such displays.

We focus on presenting techniques related to the use of the auditory and haptic modalities in HCI, as the two modalities have shown great potential for augmenting and even substituting visual displays. We begin by briefly introducing a set of audio and haptic display and interaction techniques, then present suggestions of how they could be implemented to specifically address the limitations of current assistive technology as documented in the case studies.

5.1 Auditory Display Design

The research area of Auditory display (AD) is an emerging field of study that focuses on exploring how speech and non-speech sounds can be efficiently used to convey information. Auditory display techniques are typically employed in assistive technology

to present information to visually impaired people; in monitoring applications to draw attention to relevant occurrences that are outside of the field of view; and in multimodal environments to provide an additional information channel in situations where the eyes are occupied or when using devices with limited screen space. Over the past twenty years or so, research into Auditory Display and Sonification [1] has been providing a growing body of knowledge describing how to efficiently design and use auditory output in a variety of contexts. A number of techniques have been developed and thoroughly investigated to convey information using non-verbal sounds by exploiting the characteristics of human auditory perception and the nature of audio as a medium of communication and interpretation.

For example, **Earcons** [2], **Auditory Icons** [3] and **Spearcons** [4] are auditory equivalents of desktop graphical and textual icons. Earcons are symbolic musical tones that exploit auditory characteristics such as rhythm, pitch, loudness and timbre, to communicate information about objects, actions and operations. Earcons can be structured and constructed from building blocks of audio motives, and can thus be combined into larger recognisable patterns. The structural essence of earcons makes them particularly useful for conveying hierarchical organisation and supporting navigation and orientation within an audio-only environment. Unlike Earcons, Auditory Icons use combinations of natural and everyday sounds to convey information to the user about sources of data. The idea behind Auditory Icons is to relate interface sounds to their referents in the same way that natural sounds are related to their sources. The strongest feature of auditory icons is thus their reliance on analogy to convey a message, which means that their meaning is usually quickly learnt and easily interpreted. Another benefit of using Auditory Icons is allowing for data to be categorised into distinct families, each of which can be recognised through the use of a single sound.

Spearcons are also brief audio cues and are thus similar to Auditory Icons and Earcons. However, Spearcons are created from spoken text rather than abstract tones or natural non-speech sounds. To produce a Spearcon, text must be converted to speech through a text-to-speech engine, and the resulting clip sped up to a point where it is no longer comprehensible as speech [4]. The resulting auditory cue will have unique acoustic characteristics that are specific to the spoken item. Spearcons have been found to be particularly effective when scanning through menus.

Another, perhaps more complex, form of auditory display is **Sonification**. The term Sonification refers to the process of rendering a given dataset into an acoustic presentation that enables a human listener to draw conclusion about the data, such as extracting patterns, trends, etc., through listening [5]. It is the equivalent of data visualisation. As such, the design of sonifications can be tailored towards particular tasks and particular datasets, and can allow a listener to interact and manipulate the resulting sound. Data sonification has been successfully applied in a variety of contexts; to support data exploration and analysis, to translate complex visual displays, and to create momentum art. The use of sonification techniques is particularly popular for representing and allowing auditory interaction with graphs and diagrams [6, 7, 8].

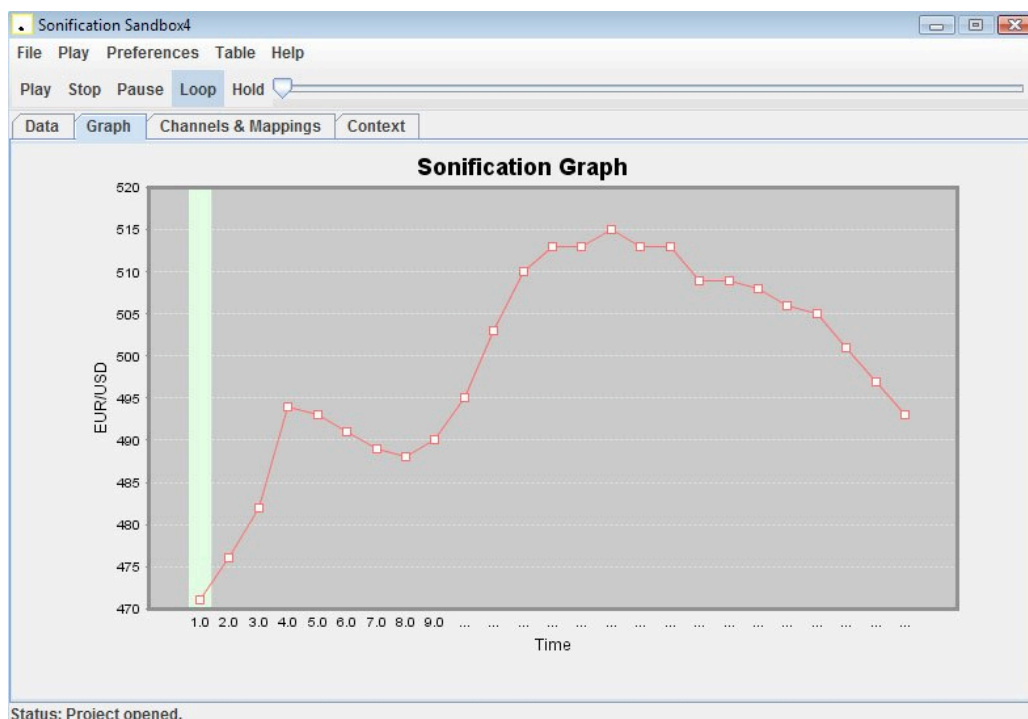


Figure 11 haptic The Sonification Sandbox, a software application for creating sonified representation of graphs [walker]

Another auditory display technique that has been increasingly investigated is the use of **3-dimensional spatial sound** to organise auditorally presented information. 3D auditory display allows for the positioning of sounding objects around a listener in a 3D space, thus conveying the illusion of spatially arranged sound sources. This is a powerful display technique and an intuitive one, since it more closely mimics the way we listen to everyday sounds in our environment and has the potential of providing

more holistic impressions of what is represented. 3D audio display can also enhance the presentation, perception and interpretation of the display techniques described above.

5.2 Haptic Display Design

Haptic displays on the other hand, are interfaces that convey information through cutaneous or kinaesthetic sensation. They allow visually represented objects to be augmented with rich physical properties, such as mass and textures, and can be used to simulate most physical sensations that can be mathematically represented, such as gravitational fields [9]. This is usually achieved by conveying haptic signals, and allowing a user to perform physical manipulations like pulling, pushing and feeling objects. Research has produced a variety of techniques for conveying information through haptic display ranging from **force feedback** technologies, to **pin arrays**, **vibrotactile** representation and tablets.



Figure 12 SensAble's PHANTOM Omni haptic device

One of the widely available desktop force feedback technologies is SensAble's PHANTOM devices (Figure 11), which allow a user to interact with virtual objects by moving a pen-stylus in a 3D virtual workspace. Attached to a mechanical arm, the stylus movements produce forces as users interact with it, stimulating kinaesthetic sensations. In conjunction with auditory display techniques, such technologies have been successfully used to support interaction with visual representations such as graphs [10], and is particularly useful for training both sighted and visually impaired users through virtual simulations.

Pin arrays are another form of tactile accessibility technology. These displays use

small pins that move up and down stimulating cutaneous sensation to represent information, and are widely used in refreshable Braille displays. To explore information, a user would use their fingertip to feel small pins that represent textures and borders of graphical information and explore such information one fraction at a time. The Opticon and VirTouch's VTPlayer mouse (Figure 12) are examples of such technologies.



Figure 13 VirTouch's VTPlayer mouse, which uses pin array display

Vibrotactile representations are another form of emerging tactile displays. **Tactons** for instance, are a form of structured tactile signals that can be used to convey abstract messages non-visually, and are equivalent to visual icons and audio Earcons [11]. Tactons have been extensively researched in order to improve their recognition and interpretation by users, and they have been successfully employed to encode abstract dynamic graphical objects, such as progress bars.

Tablet displays have also been used to present graphical information and allow for it to be explored through touch and augmented with auditory feedback to produce a Talking Tactile Tablet [12]. In some cases, graphics tablets have also been used to present tabular numeric data to visually impaired users [13]. These devices typically connect to the user's computer, who receives descriptive auditory feedback as they touch different parts of a representation that is virtually overlaid on the tablet surface.

More recently, researchers have been investigating the use of electro-rheological or magneto-rheological presentation to incorporate on tablet displays. Essentially, a layer of electricity or magnetism reactive gel is added onto a tablet-like base surface, which changes its state from fluid to solid when exposed to small electrical or magnetic

fields. Refreshable tactile displays could be created in this manner to represent shapes and diagrams that can be explored through the sense of touch. Such technology is still in its infancy however, and more development is required before it is commercially available.



Figure 14 pin Tablet displays, a diagram overlay left, and a numerical data table right, auditory feedback is display when the users touches parts of the tablets with a stylus

5.3 Synthesising the Design

The research areas of Auditory and Haptic displays are emerging fields of study that provide research-based knowledge on the use of non-visual interaction techniques in HCI. Such knowledge is seldom transferred to the real world where it can be applied in real settings and activities. We have seen how current accessibility solutions to music and sound editing tools are associated with various usability issues, which we broadly categorised in the previous section of this report. The categories are; 1) Pace of interaction, 2) Spatial manipulations through a temporal medium, and 3) information overload. There is great potential for incorporating the above research-based techniques to address some aspects of such issues. In the following, we will attempt to explore how this can be achieved.

5.3.1 Pace and structure of interaction

Usability issues in this category are attributed to the presence of a mismatch between the nature of a given task and the means by which it is executed. These issues were highlighted through various examples of interaction; the steps involved in setting up

punch-in and punch-out triggers; previewing items on the file system; and locating on-screen cursor position.

In the case of using Sonar with Window-Eyes, the structure of the steps and actions required to set a punch-in and punch-out value are not only complex, but also non intuitive as they break the flow of music recording. This process has to be much more intuitively supported at two levels; 1) navigating an audio track to locate parts of interest, and 2) issuing a command to the system to store such parts. We suggest that instead of manually inputting time values to the system, the process should be supported through two related and intuitive actions; 1) navigation of the track through listening; and 2) marking entry points through touch or speech input. The user could, for instance, simply scrub through a track to locate the part of interest. Once the part is reach, the user can simply issue a spoken command, or use a physical device to mark that part, for example pulling down an Omni PHANTOM stylus for punch-in, then pulling in it down again for punch-out.

In general, using Spearcons to browse through a list of items has been found to outperform other audio presentation techniques, including speech. Spearcons are therefore a clearly good display candidate to supporting quick browsing of text-based menus. Previews of sound files in Sound Forge for instance are currently unnecessarily presented through audio clip samples. These could instead be presented using Spearcons to display the names of the files. In this case, Spearcons would provide an overview presentation that is both discernable and fast at the same time, hence improving the overall pace and usability of the interaction. Of course a toggle function for switching between Spearcon and audio clip presentation should also be included to account for cases where users actually require previewing the content rather than the names of browsed files.

The process of locating the on-screen position of the mouse cursor could be improved by using a 3D auditory display. This is because such information is essentially spatial, and supplying it through spatial representation means has the potential to be much more intuitive. The mouse cursor could be represented using an auditory icon or an earcon, which on user command can be displayed in 3D sound thus conveying an overview of its location and easing the so is the process of locating it. Such presentation would be much more effectively delivered through headphones in this case, as they are more commonly used than full 6 or 8 speakers audio settings.

5.3.2 Spatial manipulations through a temporal medium

Issues in this category tie in well with those in the previous one, and are attributed to a mismatch between the nature of the information that form parts of a task, and the means by which it is delivered to the user.

This is particularly the case when graphical representations are in question, which are ubiquitously employed in modern music and sound editing tools. Graphics in music and sound editing tools are currently almost totally unusable by visually impaired users, and when accessible, speech is essentially used to convey their textual content. This approach misses out on the rich representational features of graphical representations, such as trends and patterns. At the same time, in the auditory realm, Sonification has been found to be a particularly effective means for representing graphs and diagrams, and for assisting exploration of patterns and trends in complex data sets. This is therefore a very attractive potential solution to the inaccessibility of the graphical elements of music and sound editing tools, one which to date has not been considered in current commercial solutions.

Sonifications can be designed to both present the underlying data of complex graphics, and to support interaction with such data through direct manipulation. The latter could be achieved by employing haptic devices that allow for physical manipulations, such as the Omni PHANTOM device, or overlay tablets. The mapping of physical spatial properties to virtual haptic devices in conjunction with intuitive sonifications of the represented data could provide accessibility and usability levels that are equivalent to visual interface controls.

The waveform representation for example, is one of the most used metaphors in visual music and sound editing, yet it is totally inaccessible. We suggest a tactile waveform. A tactile waveform could be overlaid on a tablet or a refreshable tactile tablet, with its audio parameters haptically accessible. Naturally, the tactile waveform can also be audible upon touch. This would give the visually impaired user an intimate access not only to the sound itself, but also to its detailed spectrum of acoustic parameters through both listening and touch.

Another issue highlighted in the case studies that fit under this category, is the lack of feedback when the user manipulates visual elements of the interface. The example of aligning two audio tracks was given. The primary barrier of the efficient execution of

this task is the use of a speech-only interface to manipulate elements that are graphically represented. Again, a potential solution to providing informative feedback during this task is to use a tactile display in conjunction with auditory feedback. In this case, the use of a pin array mouse might provide the required feedback in an unobtrusive manner, and convey both the progress and status updates on the completion of the task.

5.3.3 Information overload

Issues in this category are a direct result of the representational clash between the object and the means of interaction. In a music and sound editing process, users are essentially manipulating audio, and doing so through an auditory speech-based interface can sometimes lead to information overload. This was a recurring problem observed across all documented case studies.

A natural approach to overcoming such issues is therefore to distribute the various elements of interaction across multiple modalities that are available to the user. However, such distribution has to take into account the context of interaction and the affordances of each modality in relation to what is represented. Recording an audio take that is dominated by speech should be supported through non-visual means with minimum inclusion of speech-based input or output. For example using earcons or auditory icons instead of speech to communicate system status, and supporting more advanced controls through haptic means.

5.4 Summary

Research on Auditory and Haptic displays is providing increasing evidence of the potential of these two modalities to support efficient non-visual interaction with computer-based applications. Yet, such innovative research has not made it across to the realm of commercial accessible technology for music and sound editing.

An examination of the common problems encountered by visually impaired users with current accessibility approaches to mainstream music software has revealed various inadequacies in their design. A striking outcome of such examination is that approaching the design of accessibility should be done through a radical, but a rather natural perspective: to design for the visually impaired user experience tools that

match their experience, rather than simply making accessible those which were built for a different user population (i.e. sighted users).

The described interactive solutions to the three categories of usability issues suggest a shift in design paradigm. This requires not only considering accessibility earlier on in the design process, but making it its drive. Of course undertaking development to implement such approach is beyond the scope of this study, and further funding needs to be sought in order to undergo prototyping and evaluation of potential ways in which the report's research-based solutions can be combined into a fully accessible music and sound studio that matches the potential of visually impaired users.

6 Conclusions

Screen-readers are currently the most widely used accessibility tools for blind and visually impaired computer users. Screen-reader software gathers information about the content of the computer and converts it into synthesised speech or refreshable Braille output. Music and sound editing computer-based tools are highly complex interfaces that rely heavily on graphical displays to support sound and music editing and manipulation. Accessing and controlling such visually dominant displays with a screen-reader or a Braille display poses various usability problems for visually impaired musicians, producers and audio engineers.

This report presented a feasibility study into the design of more usable and intuitive non-visual interaction with music and sound editing, with an emphasis on accessibility as a user experience issue rather than merely a technological one. We have reviewed mainstream visual interfaces for music and sound editing, as well as current technological approaches to supporting non-visual access to such tools. Reflections on how they compare to one another in terms of usability, revealed a huge gap between the usability of the process of music and sound editing when supported through a visual interface and a non-visual interface.

In order to gain deeper understanding of visually impaired user experiences with such technology and how it affects their music and sound editing activities, we conducted four case studies involving representative users from the target population. The case studies focused on examining the way current accessibility solutions change the experience of music and sound editing; the result was a categorisation of typical usability issues associated with using screen-reader and Braille technology. These are: 1) Pace of interaction 2) Spatial manipulations through a temporal medium, and 3) Information overload.

In order to address these issues, we turned to academic research on non-visual human-computer interaction, and examined various existing techniques for employing the auditory and haptic modalities to support representation and interaction with information. Based on this knowledge, we recommend a range of design solutions to the identified categories of usability issues. Combining these design recommendations into a fully functional accessible music studio is beyond the scope of this feasibility

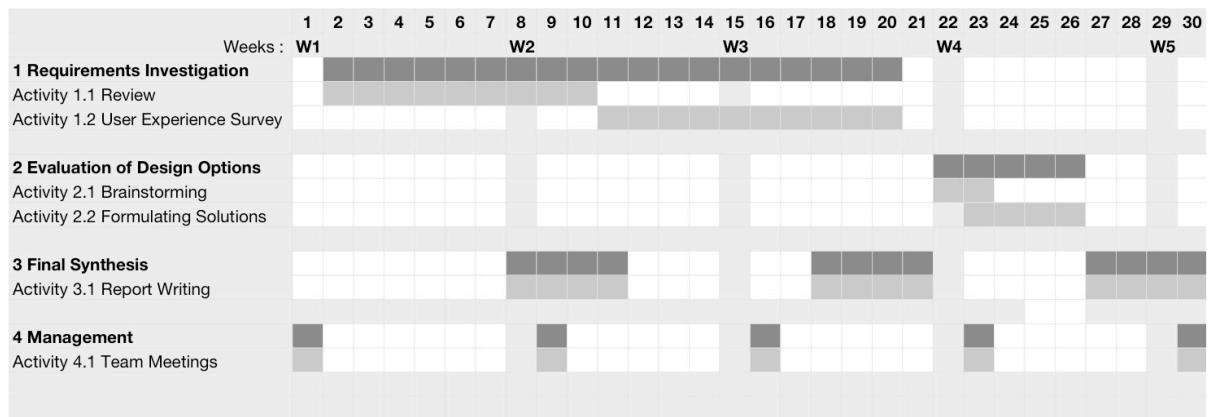
study, and a longer-term project is necessary to address the details of the implementation and evaluation of such approach to designing for accessibility. Funding will now be sought, primarily from EPSRC, to take up the recommendations of this study and develop them into an innovative solution to the problem of delivering highly accessible audio editing software which as well as addressing access, goes much nearer than current solutions to providing a user experience broadly equivalent to that of the sighted user when working in this domain.

References

- [1] Kramer, G., An Introduction to Auditory Display, in Auditory Display: Sonification, Audification, and Auditory Interfaces, G. Kramer, Editor. 1994, Addison-Wesley.
- [2] Blattner, M.M., D.A. Sumikawa & R.M. Greenberg, Earcons and Icons: Their Structure and Common Design Principles. Human-Computer Interaction, 1989. 4(1): p. 11-44.
- [3] Gaver, W.W., The SonicFinder, an Interface that Uses Auditory Icons. Human-Computer Interaction, 1989. 4(1): p. 67-94.
- [4] Walker, B. N., Nance, A., Lindsay, J., Spearcons: Speech- Based Earcons Improve Navigation Performance in Auditory Menus. In Proc. of ICAD'06, London, UK (2006).
- [5] Hermann, T. Sonification for Exploratory Data Analysis. PhD thesis, Bielefeld University, De, 2002.
- [6] Walker, B.N. & J.T. Cothran. Sonification Sandbox: a Graphical Toolkit for Auditory Graphs. in International Conference on Auditory Display (ICAD). 2003. Boston, MA, USA.
- [7] Harrar, L., Stockman, T. Designing Auditory Graph Overviews: An Examination of Discrete Vs. Continuous Sound and the Influence of Presentation Speed. In Proc. ICAD'07, 299–305, 2007.
- [8] Metatla, O., Bryan-Kinns, N., & Stockman, T. Constructing relational diagrams in audio: the multiple perspective hierarchical approach. In Proc. of Assets'08, 97–104, 2008. ACM.
- [9] Kortum, P.. HCI Beyond the GUI: Design for Haptic, Speech, Olfactory, and Other Nontraditional Interfaces. Morgan Kaufmann, 2008.
- [10] McGookin D. K., Brewster, S. A., Graph Builder: Constructing Non-Visual Visualisations. In Proc. Of BCS HCI'06 London, UK, (2006), 263-278.
- [11] Brewster, S.A. & L. Brown. Tactons: Structured Tactile Messages for Non-Visual Information Display. in AUIC 2004. 2004. Dunedin, New Zealand: Australian Computer Society.
- [12] Landau, S. & L. Wells, Merging Tactile Sensory Input and Audio Data by Means of The Talking Tactile Tablet, in EuroHaptics Conference. 2004: Trinity College Dublin.
- [13] Kildal, J. & S.A. Brewster, Exploratory Strategies and Procedures to Obtain Non-Visual Overviews Using TableVis. International Journal on Disability and Human Development, 2006. 5(3): p. 285-294.

Appendix 1: Working Plan

Gantt chart depicting the working plan project including a list and a schedule of the various activities carried out throughout its duration.



Appendix 2 - Questionnaires for case studies

The following questions were used to structure the interviews conducted in the case studies reported in this document. The questions describe the aspects of visually impaired users experience with the accessibility technology that we focused on, but do not necessary describe the order of discussions. Interviews lasted for up to 1 hour and 30 minutes, and where possible included observations of the users as they interacted with their technology of choice.

Participants in the case studies were informed that the interviews were part of a feasibility study for developing novel ways to support interaction, not with current main stream software only, but with the sounds and the music itself, and thus were encourage to not consider possible limitations with current technology when answering the questions.

Questions:

1. What is your expertise in using music and sound editing software? (e.g level of expertise, years of practice)
2. What are the primary activities for which you used music and sound editing software?(e.g. Leisure, music composition/ arrangement, audio engineering and production, etc.)
3. What technology, or combination of technologies, do you use for music composition and/ or sound editing and production (e.g. software, hardware)
4. For each technology or combination of technologies mentioned above please describe:
 - An overview of what it allows you to achieve, as well as the process by which this is achieved; e.g. does the activity you do involves more than one technology? if so, how are these combined to complete the activity?
 - Do you experience any particular problems when using these technologies? if

so, please give a brief description of these issues as well as the limitations associated with the specified technology

- Have you developed any workarounds in order to overcome the limitations of the specified technology?
- Are there problems to which you couldn't develop any workarounds to overcome?
- Are there any functionalities or features currently not present in the specified technology and that you would like to be available?

Do you have any preferences over the way certain features of the current technology support the process of sound editing, mixing, etc.? i.e. how differently would you prefer the process to be supported?