Aalborg University Copenhagen

Thesis for the degree of M.Sc. in Interaction Design at Medialogy

Exploring Situated Sonic Sketching & Prototyping

Sketch That!

Author:
Allan Koudal

Supervisors:
Stefania Serafin
Amalia de Goetzen

May 28, 2014

Aalborg University Copenhagen

Aalborg University Copenhagen
A.C. Meyers Vænge
2450 København SV, Denmark

Semester Coordinator:
Stefania Serafin

Secretary:
Lisbeth Nykjær

Semester:
10th

Title:
Sketch That! Exploring Situated Sketching & Prototyping

Project Period:
Spring 2014

Semester Theme:
Medialogy -
Specialization in Interaction Design

Supervisor(s):
Stefania Serafin
Amalia De Goetzen

Members:
Allan Kim Koudal

Copies: 4
Pages: 107
Finished: May 28th 2014

Abstract:

In this thesis the following objectives were addressed:

1. Identify current trends in sound design and sketching
2. Increase the availability of sound design to non-sound designers
3. Encourage situated sketching and prototyping
4. Identify relevant prototyping hardware and methods

The work identified Sonic Interaction Design (SID) as an emerging trend in sound and product design, which furthermore demands new more *active* means for evaluating sound design, compared to the more *passive* methods, which have traditionally and successfully been used to evaluate the sound quality of less interactive products.
New methods have already been proposed in the disciplines of Interaction Design and Industrial Design, and the work presented in this thesis highly depend on the novel methods emerging in those disciplines.
The project resulted in a working hardware and software based interface, Sketch That! showing potential for including "solder-less" sensors and microcontrollers in an integrated, yet modular interface, which could develop into a competent sound design tool, encouraging situated sketching and sound design for non-sound designers.

Contents

1 Introduction **2**
1.1 Reader's Guide 4

2 State of the Art **5**
2.1 Sketching 5
2.1.1 Vocal Sketching 7
2.2 Prototyping 8
2.2.1 Low-Fidelity Prototyping 10
2.2.2 High Fidelity Prototyping 12
2.2.3 Sketching in Hardware 13
2.2.4 Experience Prototyping 13
2.3 Approaches to Sound Design 14
2.3.1 From Foley to Function 15
2.3.2 Product Sound Design 17
2.3.3 Sonic Interaction Design 21
2.4 Evaluating Sound Design 21
2.4.1 Heuristics 23
2.5 Conclusion 24

3 Methods **27**
3.1 Human-Centred Design 27
3.1.1 User Experience 28
3.1.2 Interaction Design 29
3.2 Usability 31
3.2.1 Heuristics 31
3.2.2 Testing 32

4 Design **33**
4.1 Sound Design Methods 33
4.2 Physical Interface 34
4.3 Product Requirements 34

4.4 Specifications . . . 35

5 Iteration 1: Basic UI **37**
5.1 Choosing a sound source . . . 37
5.2 Apply effects . . . 38
5.3 Evaluation . . . 39

6 Iteration 2: Functional Interface **41**
6.1 Back End . . . 41
6.2 Tangible User Interface . . . 42
6.3 Front End . . . 43
6.4 Implemented Features . . . 45
6.5 Evaluation . . . 47
6.6 Results . . . 47
6.6.1 Discussion . . . 49

7 Iteration 3: Design a Sound **51**
7.1 Updated Product Requirements . . . 51
7.2 Implementation . . . 52
7.2.1 Visual GUI feedback . . . 52
7.2.2 Uploading and recording sounds . . . 54
7.2.3 Recording Sounds . . . 55
7.2.4 Sketch Box v2 . . . 57
7.3 Evaluation . . . 57
7.4 Results . . . 59
7.4.1 Encouraging situated sketching . . . 59
7.4.2 Functionality . . . 62
7.4.3 User Experience . . . 63
7.5 Discussion . . . 65

8 Discussion **67**

9 Conclusion **69**

10 Future Work **71**

Acknowledgements **72**

11 Acknowledgments **72**

A Prototyping Tools **78**
A.1 HTML5 78
A.2 CSS 79
A.3 Web Audio API 80
A.4 Node.js 81
A.5 Smartphones 82

B Iteration 1: Basic UI **83**
B.1 Index Cards 83
B.2 Evaluation 87

C Iteration 2: Functional Prototype **88**
C.1 Questionnaire 88

D Iteration 3 **93**
D.1 Test 93
D.1.1 Test Guide 93
D.2 Questionnaire 94
D.3 Laser Cutting Plans 98

List of Figures

2.1 A sketch of a dialogue with a sketch (from Buxton's Sketching User Experiences [10]) 7
2.2 The Design Funnel. Sketching belongs to the early ideation-heavy part, whereas prototypes belong in the latter usability-focused part. The black arrows indicate that it is an iterative process, with the red ascending arrow indicating the increased investment over time. (From Buxton [9]) 9
2.3 Profile of a product sound designer [22] 17

3.1 The multidisciplinary field of Interaction Design. IxD has been defined as an umbrella term for design encompassing a large mixture of disciplines, methods and philosophies. [32] 30

5.1 The three proposed options for selecting a sound source in the application. From left to right: (1) paste URL, (2) local sound, (3) sound library. See Appendix B.1, p. 83 for larger images. . 38
5.2 By clicking the buttons in the bottom of the screen, it was chosen what effects to apply to the sound and what parameters these should be mapped to. In the picture, the gain-button has been clicked, and the "fixed" slider is chosen as the value for the gain. By clicking filters, a similar tab would replace the gain-tab, allowing to choose what inputs to map the filter frequency to. The green preview button in the bottom right corner plays back the sound with the current chosen settings. . 39

6.1 The first iteration of the Tangible User Interface. The push buttons are marked by a triangle, the potentiometer by a P and the distance sensor by a D. The aluminium object to the right is a USB battery pack, a Microduino with a wifi-shield is located below the wires. 43

6.2 The hardware enclosed in the black box TUI. It included a distance sensor, potentiometer, two push buttons (only one in the picture) and a Microduino Core with a CC3000 WiFi module. 44
6.3 The audio graph consisted of the sound source, which was connected to the gain node, the filter node and then lastly to the destination (the speakers of the system). 45
6.4 Left: The users could select a sound from a list of predefined sound sources on the server. Right: Four filters (Low Pass, High Pass, Band Pass and All Pass) were available to choose from. The sliders below P and D showed live input from the sensor reading. 46
6.5 Left: The test-stall was located in a large open area with several other students exhibiting their prototypes. Right: A subject is interacting with the distance sensor in the prototype. 48

7.1 Left: Gain is mapped to the light sensor, causing the gain (or volume) of the playback to change according to amount of light registered by the *Sketch Box*. Note the *filter* tag next to the proximity button, indicating that a proximity-filter mapping has been configured in the filter-menu. Right: The proximity sensor controls the frequency of the high pass filter. When the *manual* button is pressed (in either of the menus), the sensor-mappings are cleared and the parameters are only controlled by the GUI slider thumb. 53
7.2 Left: The movement sensor is configured to trigger the playback of the sound. The grey bar denotes that no movement is detected - when movement is detected, the entire bar turns white. Right: The native iOS shows the list presented to the user after clicking the drop-down menu to replace the current sound "Beep" with "Button2". 54
7.3 Left - Clicking the upload file button in the browser gave direct access to a dialog where the user could choose between uploading a file or launching the Android Sound Recorder application. Right - The Android Sound Recorder. Clicking done caused the file to immediately upload to the sound library. 56

7.4 Left - the redesigned *Sketch Box* cut from 4mm oil tempered hard board. The LDR and Proximity sensors are hidden behind the perforated front plate. The toggle button on the right side is an on-off switch, and the bottom-right is the potentiometer. At the very bottom part of the PIR sensor is visible. Right - *Sketch Box* on the inside with the big battery pack mounted next to the sensors and micro controllers. 58
7.5 Left: One subject used the sound recorder to directly upload his own sound. Right: *Sketch That!* "in the wild", using passers-by to trigger sounds. 60
7.6 Comparing the profile of the designers vs the medialogists showed no significant differences between the groups. 61
7.7 Each session was awarded with a score for the *situatedness* of the sketching based on how many of the situatedness criteria were met. The x-axis represents the three point options (0, 1, 2, 3) and the height of the column denotes how many sessions reach each score. There was a total of 13 sessions. The bars are furthermore divided by the respective disciplines of the subjects. 62
7.8 Comparison of the subjective usability ratings of the interface between the presumably technical group of Medialogists and less technical Designers. 64
7.9 Comparison of the subjective usability ratings of the interface between the presumably technical group of Medialogists and less technical Designers. 65

B.1 Left: The first screen presented when opening the web application. Right: The picture shown, when the paste URL button has been clicked. This screen will appear after the "new project" has been pressed. 84
B.2 Left: *Local sound* invites the user to load a sound from the device's memory. It can be dragged (in a desktop browser) or clicking the text will open a load file dialog. Right: *Sound library* allows the user to choose a sound from a predefined library on the server. 85
B.3 Left: The screen presented to the user after selecting a sound in the preceding step. Clicking the big green preview button plays the sound as it is. Right: The screen after pressing the preview button. Press the red button to stop preview. 86

D.1 Sketch Box v2 laser cutting plans. 99

Chapter 1

Introduction

Great ideas often start their lives as sketches. The purpose is often to refine and share ideas in a quick and inexpensive manner and sketching is thus an integral part of the designers tool set. It is possible to sketch graphics, physical products and even experiences. The medium and fidelity vary, going from e.g. a simple doodle on a napkin paper to more elaborate tangible cardboard versions. Video and simple animation have also been used as sketching media. At some point the best ideas might transform from a sketch into a working prototype for further development.

While pen, paper, cardboard, tape etc. might be the most well-known media when dealing with graphical or tangible products, the picture is less clear when dealing with sound. Sound is becoming increasingly present in modern products: a hearing aid might be too small to communicate through a visual display, and the refrigerator might call you from the other room, informing that the door was not closed properly. While there have been a trend to present information more visually oriented interfaces, theres nothing new in suggesting sounds for feedback in user interfaces. Buxton, Gaver and Bly [12] stressed how audio brings important and interesting properties to the design repertoire, especially the ability of most users to monitor simultaneously a number of ongoing tasks, and concluded by suggesting that the auditory channel deserves more more attention due to the inherent functionality of sound. The value and potential of sound is also stressed by Norman, together with a sense of disappointment that even though sound is widely used in product design, it is mostly limited to signal sounds, such as buzzers, bells or tones, analogous to limiting the use of visual cues to different coloured, flashing lights [27]. Norman though also stress the importance of taking great care, as sounds easily become cute rather than useful.

But how do you sketch these sounds? Vocal sketching has been proposed, where a team of designers would use their voices to sketch the sounds [13];

similar techniques are often used by musicians to communicate ideas, perhaps to musicians playing another instrument or to ask a sound engineer for an effect that is not easily described with words. Describing sounds with words can indeed be challenging, and as described by Farnell [14] in his book *Designing Sound*, it often result in strange words of the like found in classic comics, such as "higgledy-piggledy", "Twunk ... Boyoyoyoing!", references Scooby Doo bongos or light sabres being used by professionals. Sounds are often described by the physical properties of the sound producing object and the process by which it occurred. Another approach could be recording and synthesising tools, which give the possibility to create most sounds; however the complexity often associated with such tools might be a limitation if used for sketching by designers with no experience in sound design. So now to the great final question: How do YOU sketch a sound? How do you incorporate sound in the prototyping process?

The idea of writing this thesis was based on the experience gathered from prototyping audio-haptic and visual concepts in previous projects. This process is hard and time consuming, often requiring significant amounts of programming. Hardware prototyping is often also needed, when dealing with interactive sounding products, in order to develop and evaluate the sonic interaction. This type of prototyping requires knowledge of hardware prototyping platforms, such as the Arduino[1] and a proficient knowledge of sensors and actuators. Moreover, knowledge of signal processing and physics-based synthesis is also often a requirement. A main assumption here is, that lack of skills in aforementioned areas is often discouraging to the point that designers without some sort of background in audio design may shy away from properly prototyping audio in their product design. The main idea is to prototype and present these sensorial concepts to the users *and the designer* much quicker than is often done, and perhaps even more importantly in a relevant context. This thesis was thus driven by the following objectives:

1. Identify current trends in sound design and sketching
2. Increase the availability of sound design to non-sound designers
3. Encourage situated sonic sketching and prototyping
4. Identify relevant prototyping hardware and methods

Relevant state of the art in the domain of sonic sketching and prototyping was investigated with the aim of proposing a hands-on interaction design driven approach to sound design. This resulted in a hardware-software based

[1] `http://www.arduino.cc/`

tool, *Sketch That!*, moving the sound design out of the recording studio towards experience-driven situated sonic prototyping.

1.1 Reader's Guide

For the sake of continuity, code examples are not provided in this manuscript. The development of Sketch That! though resulted in extensive amount of code, which can be found in full length on the accompanying cd. Observations and notes are likewise found on the cd, as well as additional image- and video documentation from the tests.

The implementation resulted in the web application, Sketch That!, which is accessible on the address: `http://87.72.109.96/sketchthat/index.html` The newest version of Google Chrome is recommended when accessing from laptop browser as well as mobile browsers.

Chapter 2

State of the Art

In this section relevant research in the areas of *sketching* and situated audio design is presented. The investigation starts out by describing one of the designers most important activities, namely sketching. This conceptual stage can take many forms, and is often the starting point and basis before the designer moves on in the design process. Another important tool for designers is *prototyping*, which is an important design activity, likewise used by designers as a communication tool. Prototyping is a great tool in the process of imagining using the product in a realistic context or for testing feasibility by e.g. asking users. These often overlapping activities, sketching and prototyping, are included as an attempt to place the traditionally very specific activity of sound design in a non-sound designer context.

The last part of the section presents current approaches to sound design in the fields of *product sound design (PSD)* and *sonic interaction design (SID)*. PSD originates from the discipline of industrial design, where the process is often characterised by analysing the sound of current products, tinkering and modifying the structure until the desired sound is acquired. SID moves away from the goal of simply reducing undesired sounds towards replacing the formal listening test with more explorative design and evaluation principles.

2.1 Sketching

Sketching and design dates back to the late medieval period, where the trend of separating design from the process of making emerged [10]. The sketch was used as an aid of thought and enabled the sketcher to work through several design drafts, compare them to each other, share ideas with others while keeping them so they could get back to them at a later point.

An industrial designer is likely to produce large amounts of sketches in the initial design period. In this process the designer can convey several meanings through the style of the sketch. In visual sketching, a rendering with lines moving past their natural endpoints may suggest that the designer produced the sketch in a manner of minutes or seconds (which is not necessarily the actual case) and thereby invite the viewer to share suggestions and not be afraid to tell what they really think about the idea, as the style signals that the designer is not even sure about the design himself. On the other contrary, accurate looking 3D renderings (which may or may not take equally long to produce) signals a more finished product, and is thus not as inviting to suggestions to changes to the design [10]. It is important that sketches are seen as distinct from other types of renderings, such as presentation drawings, as serious problems may arise if managers, customers or the marketing department start seeing the sketches as the final product. This is not unlikely to be the result of too high fidelity.

Bill Buxton [10] claims that sketches can be characterised by the following qualities:

1. Quick
2. Timely
3. Inexpensive
4. Disposable
5. Plentiful
6. Clear vocabulary
7. Distinct gesture
8. Minimal detail
9. Appropriate degree of refinement
10. Suggest and explore rather than confirm
11. Ambiguity

The designer should be able to produce sketches with similar qualities in the auditory domain, in order to fully appreciate the medium of sound as a design material. Sketching as a process can be thought of as a conversation between the mind and the sketch, as illustrated in Figure 2.1, p. 7. The sketch is a representation of the current knowledge, which in turns generates new knowledge when interpreted by the designer. Sketching is therefore a way of thinking. It is a way for the designer to try out ideas quickly, reveal clues and expose problems that the designer might not encounter with the knowledge remaining in the mind of the designer. Furthermore it is stressed that there is so much more to sketching than producing drawings [10]. The ability to read sketches and contribute to them is equally important and requires specialised skills according to Buxton. As such, the activity of sketching is more a question of the mindset of the designer, than about the medium used, and if a particular external representation of a sketch, being it a drawing on a napkin or a piece of programming code, it is a sketch if it is quick, tentative and truly disposable [23] - what matters is the purpose and intention (more about this in Section 2.2, p. 8).

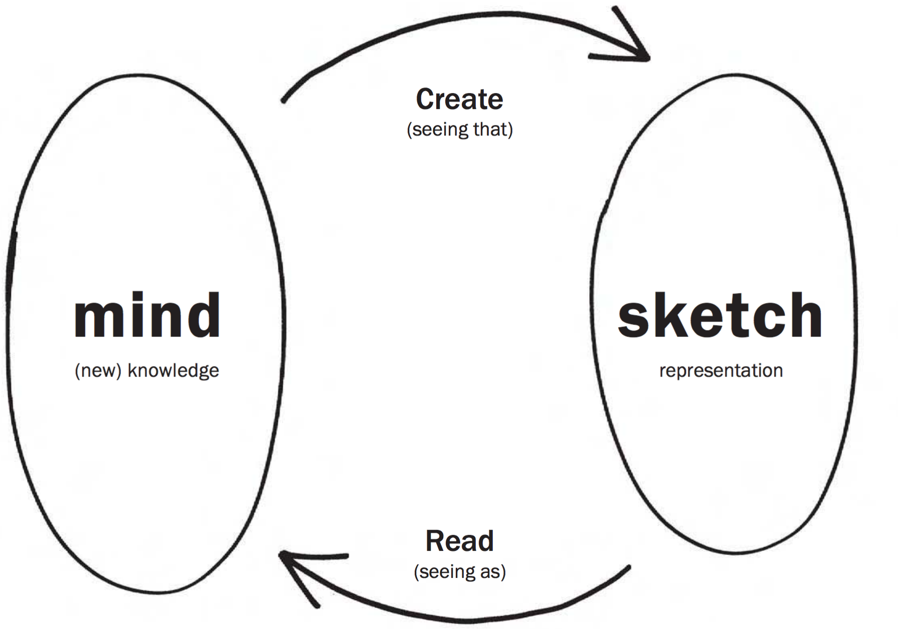

Figure 2.1: A sketch of a dialogue with a sketch (from Buxton's Sketching User Experiences [10])

2.1.1 Vocal Sketching

Vocal Sketching has been proposed as an inexpensive voice-based technique for sonic interaction design, comparable to pen and paper in the visual domain, as it is a quick, inexpensive, easy and highly communicative method to apply early in the design process [13]. While sampled sounds or found sounds might be sufficient for evaluating discrete sonic feedback in event-based interactions, this is often not sufficient when dealing with tightly coupled interactions[1], since they do not reflect the continuity of interaction. As an interesting addition, the human voice, as a sketching tool, was described as a stereotype,

[1]Embodied interaction has also been called tightly coupled interactions. In embodied interaction the user directly interacts with artefacts, embedded with technology, often benefiting from direct and highly responsive control-display loops [13]

which on one side is a powerful means of sonic drafting, but on the other hand require extensive work to defamiliarise it, to detach it from the its obvious human origins [21]. Ekman and Rinott [13] describe their findings from an initial workshop exploring *vocal sketching* as a tool for designing sonic interactions. As one may expect a major topic for observation was the social barriers to vocal interaction, and most of the participants did indeed acknowledge this as a problem; however the vast majority of participants did also report that they felt good about the sketching session and had fun. Ice-breakers, in the form of a warm up charades-like session or a brave initiator in the group was helpful (and needed) to lower the threshold of discomfort. The observations also indicated, that the use of vocal sketching tended to produce organic, complex and ecological sounds, which would have been hard to produce by synthesis, especially when using them in an interactive context. No attempts to mimic "unnatural" sounds such simple sinusoids, clicks or beeps were recorded. The following main limitations of using voice was found by the participants:

- The monopoly of the voice. One person cannot make many sounds at once(thus, sketching harmony was possible only in teams).
- The difficulty in producing specific, complex sounds.
- The lack of specific auditory control (available in sound processing software).
- Limitations due to breath cycle (e.g. long continuous sounds impossible).

2.2 Prototyping

Prototyping, as sketching, is an important activity in the process of product design. As it remained a goal to include sound more tightly in the design process, it should also be considered how designers could bring sound into the prototyping process. This section brings a brief overview of the prototyping activity in the traditional sense, without specific focus on sound design. As a starting point it is relevant to point out similarities and differences between *sketches* and *prototypes*. Even though the concepts might appear close to each other, especially in the case of low-fidelity paper prototypes and pencil sketches, Buxton makes a fairly strict distinction by stating that "sketches are not prototypes" several places in his book, Sketching User Experiences [11]. The main distinction lies in where the sketches and prototypes respectively belong in the design process. While sketches dominate the early ideation stages, Buxton place prototypes more in the later stages of the design process, where the concepts move closer to more rigid designs, which is perhaps best illustrated with Buxton's notion of the *Design Funnel* shown in Figure 2.2, p. 9. This continuum is furthermore distinguished by the purpose or intent with which the prototyping or sketching activities are performed, as illustrated in the list in Table 2.1, p. 10.

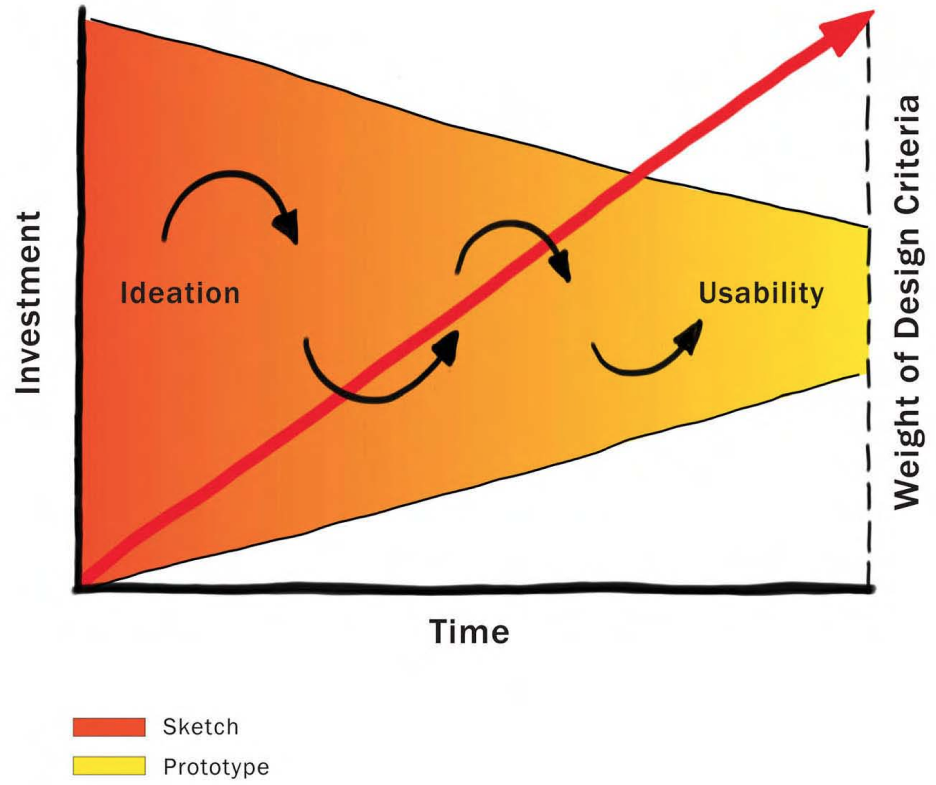

Figure 2.2: The Design Funnel. Sketching belongs to the early ideation-heavy part, whereas prototypes belong in the latter usability-focused part. The black arrows indicate that it is an iterative process, with the red ascending arrow indicating the increased investment over time. (From Buxton [9])

Prototypes range from low-fidelity prototypes made from paper or cardboard to high fidelity prototypes, much closer to the final product. A prototype allows stakeholders to interact with the envisioned product, giving a chance to experience using the envisioned product in a realistic context and explore imagined uses [34]. Prototypes serve as a communication device among team members, encourage reflection and can support designers in choosing between alternatives. They can be used to test out technical feasibility of an idea,

Sketch		**Prototype**
Evocative	$\longrightarrow$	Didactic
Suggest	$\longrightarrow$	Explore
Question	$\longrightarrow$	Answer
Propose	$\longrightarrow$	Test
Provoke	$\longrightarrow$	Resolve
Tentative	$\longrightarrow$	Specific
Noncommittal	$\longrightarrow$	Depiction

Table 2.1: *The sketch to prototype continuum.* The difference between the two lies in the intent, or purpose they serve. The arrows represent the continuum of sketches developing into prototypes. (From Buxton [9])

clarify vague requirements, in user testing and evaluation or test if the design direction is in compatible with the rest of the system development [34]. It is often recommended to prototype early and often, making each step a bit more realistic [25]. As sound is an integral part in many products (cars, computers, wearables, hearing aids etc.) it might seem obvious that sound should be considered in this prototyping process, enabling evaluation of both the aesthetic- and functional impact on the overall product design and user experience.

2.2.1 Low-Fidelity Prototyping

Low fidelity prototypes (lo-fi) do not look very similar to a final product, and are typically differentiated by choice of materials far from the final product, often inexpensive such as paper, cardboard and wood rather than electronic hardware [34]. Lo-fi prototypes are not intended for being kept and integrated in the final product, but rather serve the purpose of encouraging designers to explore, modify and discard concepts. As an example a prototype of a graphical user interface (GUI) could consist of a series of paper based sketched screens showing how a user might interact with the envisioned system; when used in conjunction with scenarios, these lo-fi prototypes offer stakeholders a chance to role-play, interact with it and step through scenarios of using the product [34]. Lo-fi prototyping can typically involve [34]:

1. *Sketching:* Simple drawings, e.g. using symbols, boxes and icons to depict intended usages of products.

2. *Index cards:* Use simple index cards of cardboard of about 3x5 inches as a simple way to prototype an interaction, often used when developing

websites. With each card representing one element a screen or task, a user can step through the cards pretending how they imagine interacting with the product.

3. *Wizard of Oz:* Inspired by the American children's novel by L. Frank Baum from 1900. When prototyping a software based product, a user will sit at a computer interacting with the proposed product. However, the behaviour of the system is actually controlled by a human operating a connected computer, mimicking the way the envisioned product would behave.

4. *Wireframes:* The wireframe is a bare-bones depiction (as the name suggests) of all the components of a page and how they fit together [19]. Wireframes are typically used by both visual designers and UX designers to establish an overview of the content and navigation elements on a webpage, where they serve as a communication tool and aid of thought, though not restricted to web-based applications.

These prototyping techniques have also been investigated in the context of audio design. As mentioned earlier, vocal sketching has been proposed as the auditory equivalent to the pen and paper in the sketching process [13]. In the project, From Foley to Function by Hug and Kemper [21], Wizard of Oz was proposed in the domain of audio design, by having a test moderator trigger recorded audio while a potential user interacts with the proposed system (see Section 2.3.1, p. 15). The authors referred to this as *electroacoustic Wizard of Oz* mockups used as an intermediate step to enable interaction, albeit *fake* interaction, to to test the interaction with the prototypes. At the final step of this method, the concepts evolved into functional prototypes, a transition from lo-fi to hi-fi prototyping. Hi-fi prototyping is described in Section 2.2.2, p. 12.

Hug [20] brings another example of the use of Wizard of Oz techniques in the realm of sonic interaction design (SID) (more about SID in Section 2.3.3, p. 21). Hug describes a series of workshops with students developing sonically augmented interactive commodities for the new everyday. Categories for these everyday commodities included the smart home, wearables and tools for professionals. Students came up with concepts such as sonically augmented future assembly line control systems, hats or thermos flasks. The Wizard of Oz technique was used in all cases to demonstrate the functionality of the concepts, sometimes with users being able to interact with these *interactive* prototypes. As opposed to the *From Foley to Function* (see 2.3.1, p. 15) approach, these concepts did not evolve into functional hi-fi prototypes.

2.2.2 High Fidelity Prototyping

In high fidelity (hi-fi) prototypes materials that would be expected in the final product are used, making a more realistic look closer to that of a final product [34]. When developing physical products an arbitrary LCD screen would be a higher fidelity prototype than a laser-cut piece of wood with interchangeable paper-based GUI screens. In software, several programming languages provide the basis for a more high fidelity prototype to be developed, with popular choices being Flash[2] (ActionScript) and Qt Project[3] (C++ based), and more recently HTML5 with CSS3 (Appendix A.1, p. 78). HTML[4] was initially released in 1993, and is the main markup language for creating and structuring webpages [5]. It is an interpreted scripting language, based on tags, rendered on the fly by the web browser. Tools in the hi-fi prototyping category furthermore include micro controllers (e.g. Arduino), smartphones [3] (Appendix A.5, p. 82) and more commonly desktop or laptop computers.

An example in the area of sonic interaction can be found in the PebbleBox, a sound producing interface created with the goal of exploring the coupling between user enactment on a physical interface and the resulting feedback [28]. With reference to the musical instrument as a good example of an interface with tight coupling between the users physical manipulation and the resulting auditory feedback, the PebbleBox was designed to experiment with this action-feedback coupling. In the PebbleBox this coupling between user action and auditory response was made loose and dynamic, in the sense that the PebbleBox contained a microphone embedded in a box with a bag of pebbles, with software taking the signal from the microphone and using it to control a granular synthesiser. The sound of the colliding pebbles caused by the user's manipulation of the PebbleBox was used as input to control parameters such as amplitude, event timing and pitch of the granular synthesis algorithm. This prototype gave the user a chance to *experience* the changes in the *appearance* and *feel* of the sound based on user enactment. With basis in the concept underlying the instrumental gesture, i.e. a physical action with an associated tangible and sonic gesture, they demonstrated how these gesture-response couplings can be extended beyond musical instrument design towards a whole new class of interfaces where coupling of sound and touch is required. The high fidelity prototype was an instrument to explore gestures inspired by interaction with traditional musical instruments and verify that their working principles and resulting human perception could be extended

[2] http://www.adobe.com/products/flash.edu.html
[3] http://qt-project.org
[4] Hypertext Markup Language
[5] http://en.wikipedia.org/wiki/HTML

to other forms of human-instrument interaction.

2.2.3 Sketching in Hardware

In the field of sound and music computing, "solder-less" interfaces such as the CUI32, together with the GROVE[6] elements have been developed to encourage quick and dirty sketching in hardware [29]. The micro controller board was designed to enable students and researchers to *sketch in hardware* with the author focusing on sound an music computing, but the system open for other types fields of research as well. The solder-free approach allows users to focus on designing the interaction rather than constructing the circuits (the GROVE elements, being it distance sensors, light sensors etc. are already equipped with the proper circuits and easily connected to the CUI32 or Arduino with modular cables), which is furthermore enabled by the wireless modules (WiFi and Bluetooth) to communicate directly with e.g. mobile devices such as smartphones, avoiding the laptop as the bridge.

2.2.4 Experience Prototyping

Experience prototyping, as the name implies, is the process of prototyping with focus on the *experience* rather than form or specific functions of the product. It is most central to this type of prototyping, that the experience of using the imagined product is captured. As an example the increasing availability of mobile devices and pervasive computing results in users carrying increasingly many products with them in their daily life. The experience of these products are likely to change according to the situation in which the product is experienced. Are you using our portable device on a plane, in public, in the forest, while biking or snowboarding? The core is to identify obstacles and design challenges in order to steer the design towards the desired experience. The tool is as such not central, role-playing, mockups, body-storming etc. can be used, whereas other prototyping techniques such as video renderings and sketches are not as good at capturing the actual user experience. The focus is thus on *active* prototyping rather than *passive* prototyping, as in experiencing the prototypes yourself rather than watching other people experiencing the prototype. As argued by Buchenau & Suri, ,,a true experience prototype for users - providing a really relevant experience - seems to require a level of resolution and functionality such that it can be "let loose" into an everyday context and more fully integrated into people's

[6]a series of solder-free sensor- and actuator elements for quick sound and general hardware sketching. `http://www.seeedstudio.com/wiki/GROVE_System`

lives." [8]. Experience prototyping is as such not a technique based on formal rules, but rather an attitude towards solving design problems. The cases of prototyping in the examples before can therefore also fit into this description, as e.g. the Wizard of Oz technique is a way of allowing users to experience imagined use of a product, without awareness of the wizard pulling the levers behind the scenes. Experience prototyping therefore does not necessarily require a functional prototype. However, the fidelity of the *experience prototype* of course also depend on the intend behind, and an experience prototype of e.g. an auditory blind spot bike detection and warning system for a car would ideally be implemented in a real car and evaluated in real traffic, with real drivers and real cyclists in order to capture the full user experience and usability.

2.3 Approaches to Sound Design

Buxton, Gaver and Bly [12] categorise sound in the three categories, *Alarms and Warning Systems*[7], *status and monitoring messages*[8] and *encoded messages*[9]. It is has been suggested that the complexity of sensors, actuators, and control logic often required to explore multi sensory feedback and design of the continuos interaction of the devices of the future pose as tremendous challenges for designers mostly used to visual thinking an discrete interactions [17]. The main problem is, that designers unfamiliar with sound design, are lacking skills required to deal with sonic interaction projects, namely in the form of *means* to present ideas to others, *language* to discuss with others, *skill set*to prototype them and *processes* to iterate them [17].

With the aforementioned issue as a basis, Rocchesso et al [17] suggested a series of *Pedagogical Approaches and Methods* in sonic interaction design intended to teach design students about sound design and enhance their interest in exploring the possibilities in SID. The first step was to *sensitise* students to sonic interactions, that is to make aware of the importance of sound in real and mediated environments, with activities ranging from *sound walks*, listening and writing *audio dramas* and exploring *audio-tactile interaction*. Following the *sensitising* process was the sketching and prototyping of sonic interactions, where techniques such as vocal sketching and sonic overlay of video were described.

Sketching and prototyping are described as separate activities, which can

[7]signals that takes priority over others, such as the car horn or burglar alarm

[8]messages designed to take attention and then quickly fade to the background of the operators mind, often used to monitor ongoing tasks

[9]more complex messages used to present numerical (or quantitative) data, e.g. earcons

be conducted separately or in sequence, as the design moves from the conceptual early sketching stage for communication within the design team towards a functional prototype which can be evaluated in a real context, which resembles the distinction between sketching and prototyping described by Buxton (see Table 2.1, p. 10). Sketching and prototyping require different tools, due to their different placement in the design process, which is furthermore described in the *pedagogical approaches and methods*, by placing techniques such as Vocal Sketching (Section 2.1.1) and sonic overlay of videoin the early stages, whereas more advanced techniques such as physical prototyping with micro-controllers (e.g. Arduino), sensors and hardware as well as software (e.g. the visual audio programming languages MaxMSP[10] or Pure Data[11]) are placed in the latter part of the design process.

In this section, a series of approaches to sound design is described in more detail.

2.3.1 From Foley to Function

Hug & Kemper [21] proposed a pedagogical approach, From Foley to Function (FFF) to using sound in the design process. In their project, the process of sound design was divided into three connected stages:

1. Foley-based
2. Electroacoustic Wizard-of-Oz mockups
3. Functional prototypes

The main issue addressed was the lack of easy to use tools for non-sound designers. The concern was, that this lack of methods and tools discourage designers from exploiting the potential of sound in product design. The authors in fact believe, as several others, that the emergence of interactive products and ubiquitous computing demand an even stronger focus on sound in product design, as the products of tomorrow (and of today) are moving towards less visible and intrusive designs, many without a screen, however still demanding means of feedback to the user; the smartphone disappears in your pocket, while other products might be too small to be equipped with a visual display or it might be undesirable due to aesthetic reasons [21]. Furthermore, visual display of data might be undesirable for safety reasons, such as in the case of driving a car. The following five challenges were presented as central to the task of developing a pedagogical approach to sound design:

[10] `http://cycling74.com/products/max/`
[11] `http://puredata.info/`

1. The stigma of "functional" sound: Sounds should exist if and only if it offers a specific functional benefit - otherwise it is superfluous.

2. The (undesired) dominance of technology: The need for a functional prototype might cause designers to make do with what is technically feasible.

3. The dialectic of tools: Every tool affords a certain functional and aesthetic direction, thus biasing the design.

4. The aesthetic and technical complexity of interactive sound: Sound design require a broad skill set and knowledge, which often fill up the entire curricula of full-time educational programmes.

5. The mythical sound designer: Sound design is often thought to be some kind of obscure art form performed by highly specialised eccentrics in high-end studios.

The goal in the study was to demystify sound and encourage (non-sound) designers to treat sound as an equal material of design, aiming at a dialogical, discursive approach to sonic interaction design, rather than an "expertise driven" approach building on what (little) could be known already [21]. Their approach however required the participants, in this case interaction design students, to possess some basic knowledge of sound editing and multitrack composition with simple effects. Basic knowledge of electronics and prototyping was furthermore required for the functional prototyping part of the process.

The authors evaluated the framework by having three group of interaction design students follow the process as part of a course, resulting in three very different prototypes. It was found that both Foley and electroacoustic mockups can encourage exploration and ad-hoc ideation, and that these processes can be though of as equivalent to mood boards and sketching in visual design. However, a main observation was that the participants had to overcome inhibitions relating to being on stage, performing and making sounds in a playful way. The authors were disappointed that the performances, even in the initial Foley stage, tended to be scripted and rehearsed, which was thought to cause a too restricted and controlled performance and ideation process for the participants. Furthermore, again to the disappointment of the authors, a tendency to see a working prototype as the only valid form of prototyping was observed as an limitation to the creative and explorative phase - non functional prototyping in a quick and dirty manner is an essential part. In order to encourage a purely sound driven approach, the authors suggested that traditional methods such as storyboarding and sketching should be restricted, at least in the early stages. Regarding the multi-sampling in the electroacoustic stage, the method could in the worst case result in a focus on scripting of the interaction process and focus on triggering and sequencing of sounds, while in the positive case, the improved control over sound parameters could lead to increased expressive and narrative control. Finally the authors strongly advocate starting with and keeping open methods for sound generation and sensor technology, as an early focus on sensor technology, be it conceptual or implemented, will take time and energy away from the sound-driven exploration.

2.3.2 Product Sound Design

This section describes an approach to *product sound design* from the perspective of Industrial Design. Product sounds can thus include both digital as well as sounds resulting from the physical workings of products. Product sound design is multidisciplinary, relating to three indispensable disciplines: *engineering*, *acoustics* and *psychology* [22]. The discipline is an elective course at the Master of Industrial Design at Delft University of Technology in the Netherlands.

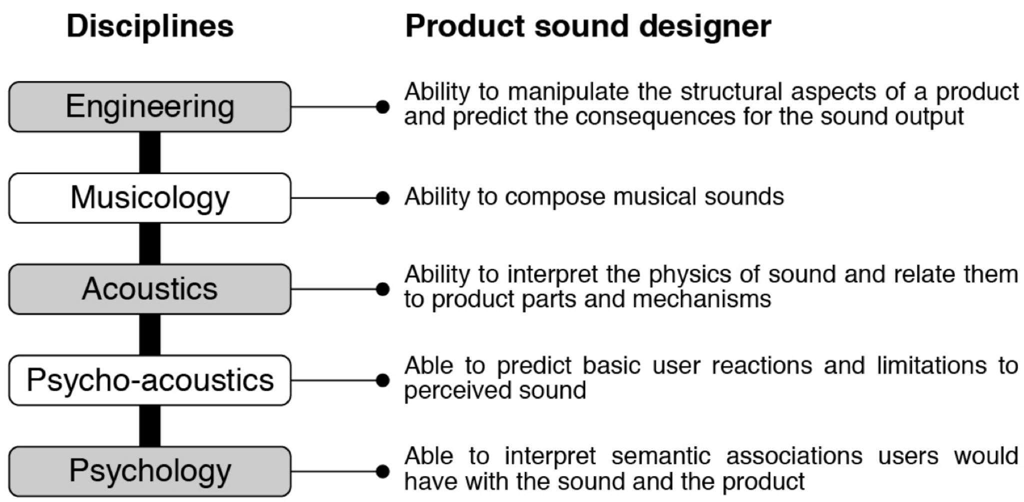

Figure 2.3: Profile of a product sound designer [22]

Intentional- and Consequential Sounds

An *intentional sound* could be an alarm. A *consequential sound* could be the the compressor of a refrigerator. Feedback can be *discrete*, in the form of e.g.

the confirmation beeps received when pressing the buttons of a microwave oven, or in the form of *continuous* monitoring that is e.g. seen in the parking assistant systems found in many modern cars. The sound feedback can furthermore be *static*, as the example of the microwave oven, where the beeps are only played when a button is pressed, or *dynamic* as in the parking assist, where the time gap between the beeps is inversely proportional to the distance to potential obstacles [22]. Intentional sounds can be placed in four categories: earcons[12][7], auditory icons[13][6], sonification and continuous sonic interaction.

Two main approaches to creating intentional sounds are claimed to exist: recording and parametric synthesis. The recording process is especially suitable for generating auditory icons, as realistic sounds are most easily obtained by this method. On the other hand, parametrically synthesised sounds (e.g. additive, subtractive, amplitude modulation and granular synthesis) provide a greater level of control over the sounds.

Stages of Product Sound Design

Langeveld et al [22] describes four stages of product sound design, which should run parallel to the main design process:

1. *sound analysis* within product usage context
2. *conceptualisation* of ideas with sounding sketches
3. *embodiment*of the concept with working and sounding prototypes
4. *detailing* of the product for manufacturing with sounds fine-tuned to their purpose

The steps are described in more details below. While the examples provided by Langeveld et al mainly focus on consequential product sounds, e.g. the attempt to equip an electric shaver with a *sporty* product sound, the focus is perhaps even more importantly on evaluating the product sounds in context. It can be argued that this *situated* product evaluation should be performed when designing intentional sounds as well.

Sound analysis is to determine when and how the product emits sound and how the sound is incorporated into the human-product interaction [22]. This is carried out in the form of observational research, where high-definition video

[12] musical messages conveying information

[13] e.g., everyday sounds mapped to computer events by analogy with everyday sound-producing events

and audio recordings are used to *place the product in context with the user* in an environment natural to the human-product interactions. Comparisons (e.g. spectrograms) are made between product sounds occurring in natural environment, and the actual sound of the product isolated from environmental effects. In this step, attention should be paid to:

- Acoustic effect of environment on the sound
- Other environmental/product-related sounds that could mask the sound in question
- Interaction of the product with the user and environment
- Facial expressions of users for detecting unpleasant or unwanted sounds
- Stages of product use and occurrence of sound in any given stage
- Duration of the product use and exposure to sound
- Impact of sound on product usability

In order to pinpoint acoustic regions than can cause sensory discomfort and problems with sound, the sound analysis stage should continue by *disassembling the product step-by-step*, recording the sound at each step until the last sound-producing component is left. Acoustical and psychoacoustic analyses should be performed at each step, so it can be identified which existing components make undesirable sounds. Consequential sounds are logically the result of the physical workings of a product, hence the disassembly to identify the undesired sound emitters. However, undesired sound emitters could also be the result of badly designed intentional sound feedback, and similar methods could thus be used to identify the undesired sounding components in the digital domain. The (digital) intentional sounds furthermore hold the advantage of being quick and easily interchanged and manipulated with the proper tools.

Conceptualisation of the to-be-designed new product sounds can now follow, incorporating the desired product experience, which is perhaps defined in the product brief, however with focus on the sound-specific parts of the experience. As an example, a shaver might be described as *sporty* in the product brief, which does not necessarily dictate the sounds having to be *sporty*; the goal could also be fulfilled by applying relating semantic concepts such as *powerful* or *energetic*. These *semantic associations* are in fact an important part in this step, as to determine which underlying concepts can be taken further. Suggested methods in this step are:

- Mind-mapping - systematically unravel meaning of the abstract term of desired experience, relating it to physical properties of objects/interactions/sounds. Define metaphors.

- Bodily explorations - designers put themselves in e.g. a sporty mood to determine which situations feel *sporty*.

- Acting out - physically act out scenarios, by vocalising sounds, moving body parts etc in order to determine physical and temporal properties of the desired experience.

Audiolizing in the form of sound sketching is the next step, which is performed once a concept has been selected, with the ultimate goal of finding auditory links that may underline the selected concept (and desired experience). *Tinkering* is suggested, and performed by designers being encouraged to find ordinary everyday objects that can express the desired auditory expression when in interaction with other objects, movements and actions. Vocalisation could, like vocal sketching (Section 2.1.1, p.7), be used to imitate sound in the desired auditory experience. The phase is completed when the desired auditory expression has been determined. Digital tools combined with physical tools could perhaps, as suggested by Hug & Kemper be utilised in this stage.

Embodiment is the first time designers encounter sounds emitting from the newly designed product. This step deals with the physical product parts, that need to be altered or replaced in order to create the desired auditory experience, and *prototyping* is is thus a central activity at this stage. Acoustical analysis should, as in the sound analysis stage, be performed on each occurring sound. Intentional sounds require more digital techniques (such as musical instruments, recorded and edited with software tools) to construct the sounds, whereas the consequential sounds and applications would require more analogue techniques (such as physical measures to dampen the sounds of e.g. engine noise in a car).

The embodiment phase is complete, when the guidelines for the final prototype is achieved. Apart from the acoustical measurements to determine technical requirements, users could also be included for subjective evaluation of the sounds or evaluation of the interaction by e.g. acting out scenarios with the product towards the designers in a Wizard of Oz fashion. This step would be analog to the Electroacoustic Wizard-of-Oz mockup stage in the procedure described by Hug & Kemper.

Detailing is the final phase, where fine tuning of the product sound takes place, where the final prototype being build and and the product takes its final shape. More extensive user observations and semantic differentials are common at this step. This phase should conclude with the product being ready for manufacturing.

2.3.3 Sonic Interaction Design

Sonic Interaction Design (SID) is a relatively new design discipline. The discipline usually place sound as a continuous feedback modality in the interaction loop. Sonic interaction design have been placed in the intersection between auditory display, ubiquitous computing, interaction design and interactive arts [33].

Since the beginning sound design for products has been focused on the elimination of undesired sounds that are produced through interaction, with engineers and designers using precise measurements to alter the mechanics to eliminate unwanted sounds and produce the desired sounds in their products [18]. In product sound design, the focus has thus been on *reduction*, rather than quality when working with sound, which is still a dominating trend, though the focus has been widened, by incorporating qualitative user tests by e.g. having users assess the annoyance of the sound produced by coffee makers, vacuum cleaners and other products; this type of evaluation though often fail to account for the users' emotional and cognitive responses to the functional and aesthetic aspects of a product [18], which are central to the user experience of a product.

SID aims to move away from the rigid methods and guidelines that have been traditionally adopted in sound and music computing communities, as strictly technical recommendations and formal listening tests are replaced by more explorative design and evaluation principles [18]. The reason for this shift is perhaps, that the increasing availability of hardware and physical computing resources (e.g. Arduino, Raspberry Pi[14]) lowers the entry bar for practitioners to produce sounding artefacts.

2.4 Evaluating Sound Design

While established methods exist for the evaluation of sound [4][5][16], these methods are often characterised by being *passive*, in the sense that the assessment of sounds are usually performed as listening sessions in controlled

[14] `http://www.raspberrypi.org/`

lab settings, rating quality parameters such as *loudness*, *sharpness* or *roughness*. Semantic differentials are also popular to assess how the sounds suit the intended purpose, e.g. warning sounds should produce high loadings on adjectives like dangerous, frightening or unpleasant [16]. While this type of evaluation might be suitable for some tasks, e.g. listening tests to assess the quality sound file compression , other methods are needed when evaluating interaction based sounds. There appears to be a need for new evaluation methods, especially in the area of interactive sound design, moving the sound evaluation out of the traditional *passive* lab settings towards a more *active* situated evaluation.

Traditionally sound evaluation techniques usually involve psychoacoustics to describe and assess the quality of a sound. These quality ratings can e.g. be achieved from listening tests, where users rate the perceived quality of the sound, according to psychoacoustic descriptive parameters or quantities such as loudness, tonality, roughness, sharpness and pitch, in controlled lab settings in order to ensure completely identical test conditions. This have even resulted in signal processing methods and software tools enabling an engineer-friendly instrumental approach to sound quality assessment [5], with the car industry as a great driving force for this type of evaluation of mechanically generated noise such as engine noise and more recently the increasing amount of consequential (or *designed*) sounds present in cars and other modern electronic products.

A big challenge with these traditional *passive* evaluations is thus found in the difficulties faced when using them to evaluate sounds based on the naturally unpredictable nature of human interaction. Sound quality assessment most often involve the assessment of prerecorded sounds, in order to keep control of the experimental conditions.

Bodden [4] compares the advantages of the common laboratory tests versus the less common field evaluation methods for sound quality assessment:

Lab tests

- the test is reproducible
- all subjects have identical test conditions
- if products are compared, they can be evaluated in identical states of operation
- different sounds can directly be compared
- stimuli can adaptively be modified depending on the subjects answer, e.g., to efficiently identify target sounds
- the test is time-efficient

Field tests

- it is a representative situation for the usage of a product in daily life

- a typical handling of the product is possible
- interaction with the product is possible
- subjects can individually select typical or critical states of operation

Susini et al [37] provide an interesting example of how *passive* listening tests yield significantly different results than *active* user interaction with an interface in relation to the ratings of the perceived usability, appraisal and naturalness of discrete interface feedback sounds. They used the example of an Automatic Teller Machine (ATM) to carry out an experiment, where the keypad sound was replaced with sounds belonging to the three different categories of *causal*, *symbolic* and *arbitrary* feedback sounds. The interaction was performed on a simulated ATM GUI[15], with an interface programmed in MaxMSP and input was captured using an USB[16] numeric keypad. Subjects rated the sounds in relation to the appraisal (pleasantness), naturalness (did it sound like a keystroke sound) and usability (did it support the task), with two groups of participants operating either a defective (only 70% of the keystrokes provided feedback) and a fully functional interface (all keystrokes provided feedback). Their results suggested that the manipulation of the interface modified the user's appraisal of the sounds usability and pleasantness. As an interesting point it was found, that the appraisal (how pleasant they were) of the sounds with medium usability (symbolic mapping) were exaggerated when using the system in context - the sounds became less pleasant when operating a defective interface. On the contrary, the natural causal sounds and unnatural abstract sounds remained consistent in the ratings. They conclude by emphasising that designers must test the quality of their sound design by having users actively manipulate the interface. These findings could thus be said to support the need for means for more easily evaluating the sounds in context, as early in the design process as possible.

2.4.1 Heuristics

Hug proposed a series suggestions for a heuristic framework for designing schizophonic[17] interactive commodities [20]. Commodities refereed to everyday objects, referring to the increasingly amount of complex, seemingly

[15]Graphical User Interface

[16]Universal Serial Bus is a standard used for connecting peripheral devices to a computer

[17]Schizozphonia denotes the separation of sound from its natural sources by electoracoustic means [20].

autonomous networked devices available in our everyday lives, such as mobile communication devices, household appliances, clothing etc. embedded with technology. These devices often appear as *black boxes*, where the user is not aware of the inner workings, and sound design therefore pose as a possible way of embedding displays to provide these often small, disappearing devices with sonic identity and means of expression and communication. The work was based on a series of workshops, where interactive commodities were designed by student groups and demonstrated with the Wizard of Oz prototyping techniques. The concepts included a barking wallet expressing skepticism or unfriendliness when the user was about to spend money, and a sofa adapting sonic character based on the social setting. A set of heuristics consisting of a series of semantic differentials was proposed:

Private-Public refers to the considerations needed when designing sounds for private or public spaces. In private spaces, the sound can be much freer, whereas in public more care has to be taken to ensure that the sounds are only heard by the ones concerned. The latter could e.g. be achieved by performing subtle modifications to the sounds only picked up by the ones who know what to listen for.

Tool-Assistant is used to distinguish between characteristics of the sounds representing work that is being done *with* an artefact rather than sound produced by the artefact itself. In the former the sound relates to the qualities of the work being done with the artefact, whereas the latter represent the artefact's behaviour as an assistant to the user with the artefact having varying degree of autonomous character.

Causal-Professional represent different levels of complexity and attention demand. The causal sounds can be designed with minimum complexity, or just enough to make the sound interesting enough to get noticed before the user quickly shifts the attention away again. In the professional sounds, the interaction dictates the complexity of the sounds, as the control gestures, function and procedures are essential.

2.5 Conclusion

In this chapter a series of prototyping methods and tools, as well as state of the art product sound design methods were presented. While the sound design techniques described provide valid and promising suggestions for promoting and simplifying sound design to non-sound designers, much work still lies ahead. This body of previous research serves a solid platform to continue the pursuit of fulfilling the goals of applying relevant prototyping hardware and methods to encourage situated sonic sketching and prototyping.

The availability of solder-free elements and micro-controllers for quick sketching in hardware (Section 2.2.3, p. 13) pose as a great opportunity for a practical approach to bringing sonic sketching and prototyping closer to the designer.

In the FFF approach (Section 2.3.1, p. 15) the issue was approached by proposing a three-step method to help Interaction Designers in the process of embracing sound design in their products. This construction resulted in the three separate (but connected) steps of Foley-recording, with focus on obtaining the proposed sounds, the electroacoustic Wizard of step, creating a *fake* coupling to the interactions and the final step of a functional prototype, where sensor-technology was used to demonstrate the proposed sonic interaction. This process though still suffered from the inhibitions of social discomfort of performing on stage in the steps involving the participants acting out the scenarios on stage. A main concern from the authors were the scripted an rehearsed approach the students had, even in the early Foley stage, which was thought to inhibit the ideation process, as well as the inhibition caused by many participants seeing a working prototype as the only valid form. The setup furthermore requires knowledge of sound production, such as recording and multitrack playback, and the hardware setup for the Electroacoustic Wizard of Oz setup seems a rather complex and stationary setup, with MIDI keyboards, speakers and computers required to play back the sounds.

The advent of solder-free hardware and micro controllers, as well as modern hi-fi prototyping techniques such as HTML5 and smartphones, pose as an interesting opportunity for proposing a perhaps more integrated, compact and portable (hopefully thereby situated) approach.These hi-fi techniques could perhaps be used to streamline the process, without removing the first lo-fi step which could include Foley recordings or vocal sketching techniques - as stated by Buxton (Section 2.2, p. 8) the difference between sketching and prototyping lies in the intent and stage of the design process, rather than the technique or tools used. This might even serve as a bridge towards the PSD approach (described in Section 2.3.2, p. 17) where the tool could assist in the activities of conceptualising, audiolizing, and embodying. An easy to use hardware/software solution, where sensor input could effortlessly be used as input to audio processing, could perhaps be used in the PSD sound analysis step as well, to place the proposed product sound with the user in a natural environment. Hopefully this could address the issues of social discomfort of being on stage, as well as easing the accessibility of functional prototypes to the designers highly valuing this type of prototype. This type of approach could perhaps even serve as a catalyst for promoting the exploration of *active* sound evaluation such as the ATM example by Susini et al (Section 2.4, p. 23), and perhaps even further, by having designers bring the portable sketching, prototyping and perhaps *field* evaluation tools with them, always ready to sketch, iterate and evaluate SID or PSD ideas.

Chapter 3

Methods

This section describes the methods and design approaches used in the design of *Sketch That!* consisting of a physical device embedding sensors and micro-controllers as well as the graphical browser-based interface. Human-Centred Design (HCD) was central in the design of the product, as keeping the goals of the user in focus throughout the process has proven essential to the success of product design.

3.1 Human-Centred Design

When designing product sounds one should consider the most advanced trends in product design, with human-centred design being one of these trends; however the application of this in sound is almost unknown, probably due to lack of prototyping tools and suitable examples [38]. It was thus the aim for the CLOSED[1] initiative to provide a functional-aesthetic measurement tool that could be profitably used by designers [38]. *Sketch That!* aims to contribute to the mission of CLOSED by increasing the availability of situated sketching and prototyping tools to non-sound designers.

Human-centred design for interactive systems address the planning and management concerning both hardware and software components [34]. The approach involves:

1. The active involvement of users and a clear understanding of user and task requirements
2. An appropriate allocation of function between users and technology
3. The iteration of design solutions

[1] Closing the Loop Of Sound Evaluation and Design

4. Multi-disciplinary design

This is furthermore elaborated in the four-step activity lifecycle described below. The process should iterate from the earliest stage of the project through to product completion, which is characterised as being the stage when the product meets its requirements. The process should be prefaced with a planning phase, identifying design activities and dividing responsibilities between the involved parts and time planning in order to ensure that the necessary feedback makes it in time to be incorporated into the project schedule. It is thus important to:

1. Understand and specify the context of use
2. Specify the user and organisational requirements
3. Produce design solutions
4. Evaluate designs against requirements

One of the strengths of iterative design is that the design is based on the learnings from the feedback loop created from the designer's work moving from initial idea conception, evaluation and then back to the designer in form of feedback for improvements. This loop requires the existence of a method and corresponding prototyping tool for the designer to conceptualise and present the ideas for this evaluation to happen. There is a strong need to provide designers with a tool and set of skills to be able to design the sonic behaviour of future products [26]. These principles holds both for the process of developing the tool for the designers, as well as the way of working the tool should afford to the prospective users. The design of *Sketch That!* therefore followed an iterative UCD-based process.

3.1.1 User Experience

User-Centred Design is often used to design with the user experience central, often referred to as User Experience Design (UxD). While in product design, form usually follow function (which makes sense when designing the inner workings of a product), UxD is all about the experience of using the product, and the design should in stead be dictated by the psychology and behaviour of the users [19]. One cannot as such design a user experience, but rather design *for* a user experience [32]. Rather than being limited to aesthetics and function, UxD often deals with questions of context - where should the product be used and for what? The aesthetics might determine form and texture of buttons on appliances, and the functional design makes sure the buttons trigger the right functions. UxD would on the other hand ask: is that button too small for such an important function? This process is of creating engaging and efficient user experiences is called *user experience design*, and the concept is very simple: Take the user into account at any step of the way as you develop your product [19].

Sound design should consider aesthetics and functionality in context as well, and situated development and evaluation of the design in the relevant context requires means to bring the design into this context. Evaluating the user experience impact of a redesigned turn indicator sound in an upcoming car model ideally requires experiencing it in context of the upcoming car. As most modern cars are equipped with stereo systems and CD-players, this could easily be achieved by playing a recorded sample of a proposed turn indicator sound form a CD in the car. But is that enough? It would likely be more ideal to experience the new turn indicator sound as a result of the actual interaction with the product, namely by triggering the actual new sound from the turn indicator stalk while driving. Equally, evaluating of the impact of a notification informing that the refrigerator door was left open would ideally be performed in the actual context, rather than limiting the evaluation to controlled-setting playback of sound, assessing if the sounds are perceived as urgent or annoying. Install a small audio player with a trigger on a current fridge and program it to play the sound when the door is left open. How many times can the sound be played before it gets *too* annoying? Is it supposed to even get annoying? How annoying? Will the user experience cause the user to get rid of the product after a week?

3.1.2 Interaction Design

Interaction design (IxD) was perhaps first introduced by Bill Moggridge, based on his experiences with creating the worlds first laptop computer. His original focus on the physical design, due to his background in industrial design, soon developed into a great interest in the virtual properties of this new product, which he saw as an opportunity for creating a new design discipline for creating imaginative and attractive solutions in a virtual world; it was initially dubbed "Soft-face" as a merge between software and interface, but later changed to Interaction Design [24]. As in industrial design this discipline should deal with subjective and qualitative matters, starting with the users need and desires striving to create designs giving aesthetic pleasure and lasting satisfaction and enjoyment, which seems to be in par with the overall principles of HCD.

Interaction design has since been described as ,,designing interactive products to support the way people communicate and interact in their everyday and working lives" [32], or put in another way, as creating user experiences that enhance and augment the way people work, communicate and interact. Gillian Crampton Smith sums it up in one sentence as being ,,about shaping our everyday life through digital artefacts - for work, for play, and for entertainment." [35], as an analogy for the way industrial designers shaped our everyday life through objects designed for our offices and homes. Elaborating on this, designing a computer-based system or device is not only about designing what it looks like, but how it *behaves*[35]. Interaction design is still a relatively new discipline, or merge of many disciplines, and these definitions are just a few of many, albeit useful to provide a guiding basis for the overall meaning of the term. The discipline of interaction design has indeed been adopted as an umbrella term for the vast amount of disciplines it takes to design products and user experiences, including user interface design, software design, user-centred design, product design, web design, experience design and interactive system design, as illustrated in Figure 3.1, p. 30.

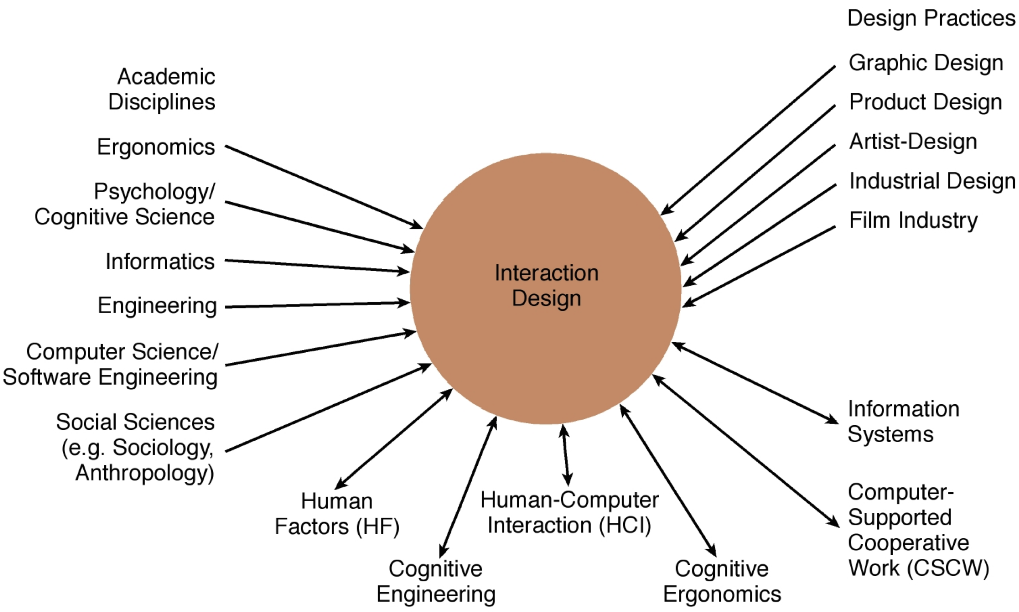

Figure 3.1: The multidisciplinary field of Interaction Design. IxD has been defined as an umbrella term for design encompassing a large mixture of disciplines, methods and philosophies. [32]

Preece et al [32] describes interaction design to involve the four basic activities:

1. Identifying needs and establishing requirements for the user experience

2. Developing alternative designs that meet those requirements

3. Building interactive versions of the designs so that they can be communicated and assessed

4. Evaluating what is being built throughout the process and the user experience it offers

The process of interaction design can thus be said to follow a user centred approach, with the principles described above strikingly close to the activities and principles of HCD described in the beginning of this section (p. 27).

Interaction Design can be said to be concerned mainly with *digital things*, or as working with *digital materials* like software, electronics, communication networks, and the like, which dictates new requirements for e.g. sketching practices compared to the traditional pen and paper [23]. Much the same way, designing sounds for products require different techniques for sketching and prototyping, perhaps to an even higher degree when coupled with interaction design to create interactive sounding products. A main mission in this thesis is therefore to explore the possibilities of sketching and prototyping product sound design.

3.2 Usability

Usability was kept central in the process of designing and implementing *Sketch That!*, due to the mission of developing a sound design tool for non-sound designers. In the early stages the usability assessment was mainly of the analytical form, where the system was assessed by users experienced with interface design. As the process moved on the focus shifted towards elements of more traditional usability testing aiming to include actual prospective users in the evaluation process to see how the system was assessed by the actual users.

3.2.1 Heuristics

When designing for usability, the usability heuristics defined by Nielsen [30] are useful for quickly discovering usability problems in an interface. These methods are often used in the evaluation of website usability, and are thus relevant for the web-based interface proposed in this thesis. The principles are:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognise, diagnose, and recover from errors
10. Help and documentation

These usability principles, which are called heuristics when used for evaluation, were kept in mind throughout the process. The *experts* in these evaluations included fellow Interaction Design and Medialogy students used in the evaluation of the system. Though not seasoned experts quite yet, they were thought to have above average knowledge in the fields of interaction design, usability evaluation and sound design, some in few and some in all of the areas. While this provided helpful in the evaluation of the overall interface, the nature of the project being addressed at non-sound designers meant that the evaluation of the system at some point had to include users from the intended target group.

3.2.2 Testing

Usability testing was performed in the evaluation of the product, in the sense that the product was presented to prospective users, asking them to carry out specific tasks while observing their behaviour. Usability testing emphasises the property being usable, in other words it is the product and not the user being tested in a controlled setting, repeatedly measuring the performance of users performing a repeated task [31]. The main data gathering methods used in this project were observations and user satisfaction questionnaires.

Chapter 4

Design

This section describes the design and implementation of the sound sketching tool, *Sketch that!*, which is highly based on the principles described in PSD (Section 2.3.2, p. 17) and FFF (Section 2.3.1, p. 15). The design followed an iterative process, with each step evaluated by presenting the current design to potential users, collecting feedback and constantly improving the design based on the user input.

4.1 Sound Design Methods

The design was mainly based on the methods described by Hug and Kemper [21] and the methods from product sound design as described by Langeveld et al [22]. The focus was on allowing designers to try out ideas for sounds as early in the process as possible with the goal of enabling a truly situated sound- and interaction driven approach to sound design. The tool should require no prior experience with sound design, specialised software tools or sensors technology. In the *From Foley to Function* approach, the activities in the second step *electroacoustic Wizard-of-Oz mockup* required musical production tools such as multi-sampling software and MIDI keyboards; it is a goal to reduce (or remove) the need for this middle step, and the hardware and software included in this step. The goal remained to create a tool to encourage (and enable) "quick and dirty" audio sketching and prototyping, which should be able to be performed situated in the correct design context: designing for a car? Do it in the car. Designing for a kitchen appliance? Do it in the kitchen. The goals remained close to the five challenges defined by Hug and Kemper in their "From Foley to Function" approach (2.3.1, p.15) and the tool is intended to be applicable in all stages sound design described by Langeveld et al (2.3.2, p. 18).

Vocal sketching has been proposed as viable tool in sonic interaction design, though with several limitations at the current stage (2.1.1, p.7). Vocal sketching as a specific technique is not considered in this phase of the design. While vocal sketching, in the form described by Ekman and Rinott [13], precedes functional prototypes and mockups, the voice is still welcomed as a sound source and could perhaps be encouraged in the Foley stage.

While many product sounds are based on synthesis, this technique is left out at this early stage of the design of the tool in order to keep a reasonable scope. It should however still be considered, when deciding on the platform to use for implementing the tool, as this functionality is likely to be relevant to further development. Apart from the time-factor, a main reason for leaving out synthesis at this early step is furthermore, that the tool is mainly intended for non-sound designers, and synthesis is regarded a relatively advanced technique.

4.2 Physical Interface

Inspired by the principle of sketching in hardware (Section 2.2.3, p. 13), the physical interface should be designed to be compatible with these easily accessible pieces "solder-less" hardware. This is important in order to ensure the possibility of a modular hardware setup, where sensors can ideally be removed or added based on the intent of the sketch or prototype. Since these "solder-less" pieces usually come equipped with the circuits (resistors, transistors etc.), they are as close to with the common micro-controllers (Arduino etc.).

The micro-controllers should be compact to ensure the portability of the interface. They should furthermore be programmed using a common language, such as the Arduino language, which is relatively straight forward to program in, though basic programming skills are still needed. This should ensure a platform with room for future development, as well as being relatively open to the more *advanced* users.

4.3 Product Requirements

The following requirements were defined as possible improvements to the promising methods described in PSD (Section 2.3.2, p. 17) and FFF (Section 2.3.1, p. 15).

1. Require no specialised musical software or hardware tools (e.g. MIDI)

2. As little platform dependency as possible for easy availability
3. Encourage situated sketching and prototyping
4. Be easy and fast to use - as little setup as possible should be required
5. Encourage design of *intentional sounds* in the form of continuous sound feedback (SID) as well as the more traditional *discrete* feedback sounds (Section 2.3.2, p. 17)
6. Compact and portable physical interface to encourage situated sketching
7. Compatible with off the shelf "solder-less" hardware for quick prototyping

These specifications were defined with the goal of encouraging a truly sound-driven approach to product sound design. The principles of sketching, as defined by Buxton (Section 2.1, p. 5)were central in the design process. Designers should be able to quickly change the sounds, show them to other designers or users, dispose of them and try out new ideas in a truly iterative sketching and prototyping process. The tool should however not be limited to sketching, but also accommodate the designers to quickly move the concepts further to the following prototyping stages, lo-fi, hi-fi and experience prototyping as well.

4.4 Specifications

Dictated by the requirements of *easy availability* and *no special hardware/software*, the web browser was chosen as the environment for execution. Browsers are available on all modern personal computers and on virtually any modern smartphone or tablet. The main markup language used in browsers is HTML (A.1, p. 78). As HTML5 alone does not do much other alone than simple audio playback, it was chosen to combine it with a backend of JavaScript, namely Web Audio API (A.3, p. 78). The main aim was to ensure cross-browser and device compatibility from the beginning, not restricting the software to have to run on specific platforms. The advantage of HTML + Web Audio API (and CSS[1] for the styling and layout) was that the application could be written once and then being available on all platforms without the

[1]Cascading Style Sheets are used to decouple the styling and layout from content in web pages

need to port the application to separate platforms, such as OSX[2], Windows[3], iOS[4], Android[5] or other popular platforms.

With a web-based application, the user should be able to type in the URL[6] and be ready to go - no special plug-ins or installations required.

The Web Audio API furthermore contains functionality to program synthesisers, and the door is thus still open for implementing synthesising in the tool at a later stage.

[2] Apples current operating system

[3] Microsofts operating system

[4] Apples mobile operating system found on iPhones, iPads etc.

[5] Google's open source mobile operating system, found on smartphones and other embedded devices

[6] URL: Universal Resource Locator - the address of a webpage

Chapter 5

Iteration 1: Basic UI

In this iteration, the focus was on the graphical user interface and basic logic of the system interaction. A quick wireframe was created to visualise the different steps of the interaction. The wireframe consisted of a series of static images of the proposed graphical interface. For practical reasons the interface was designed to fit a smartphone screen from the beginning. The iPhone 4 resolution of 640x960 pixels was used as reference. Inspired by the three steps in the FFF approach (Section 2.3.1, p. 15) the product was designed to include the following basic interaction steps:

1. Choose sound source
2. Apply effects
3. Play

These steps are the absolute minimum to select a sound, manipulate it with a given set of effects and play it back. It is the first encounter the user is expected to have with the interface, and thus a logical place to verify before starting the implementation.

5.1 Choosing a sound source

This was the first step to be performed by the user. The user was presented with three options: (1) paste URL, (2) Local Sound, (3) Sound Library.

The first step, *paste url*, was mainly designed with smartphone usage in mind. On smartphones, especially iOS are relatively restricted in file access from the browser, and it is thus not as easy as on a desktop computer to download sounds to the device and access them from the browser. The paste

URL option was proposed to accommodate these restrictions, as many sounds are available online including large free (and paid) foley sound libraries.

Local sound was included to let the user choose a sound already stored on their own device. While iOS only allowed file access to images, this restriction was not present on Android. Some users might have a large library of sounds on their hard drive and should be able to use those sounds directly in the application. Furthermore, users might record sounds, save them to their hard drive and load the sounds into the application this way.

The *Sound library* contains a collection of sounds placed on a server, thus made directly available to use in the application. This feature could also be expanded for users to upload sounds, sharing them available to other users.

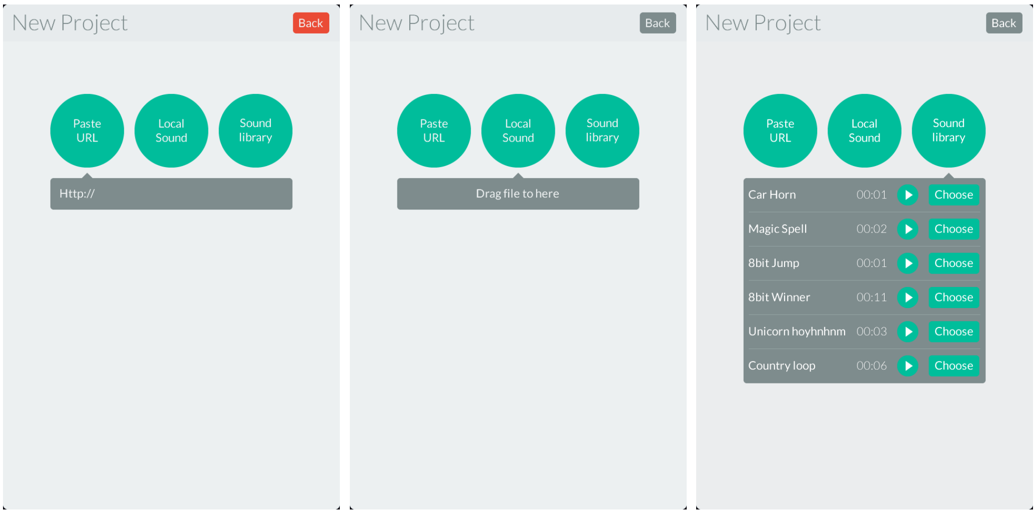

Figure 5.1: The three proposed options for selecting a sound source in the application. From left to right: (1) paste URL, (2) local sound, (3) sound library. See Appendix B.1, p. 83 for larger images.

5.2 Apply effects

The next step was to apply effects (or manipulations) to the sound source. In this first step filters, playback speed and gain were available to choose. The proposed interface, contained a series of buttons in the bottom of the screen, one for each respective effect (gain, filters, speed), and by clicking on the buttons a tab shows up on the screen, where the user could choose what parameter to map the effect to. In the first implementation, two basic sensor inputs were suggested: *distance* and *potentiometer*. As a default option, the

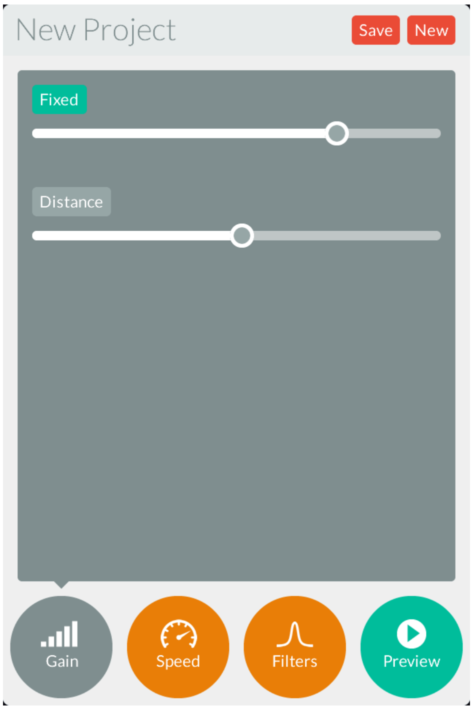

Figure 5.2: By clicking the buttons in the bottom of the screen, it was chosen what effects to apply to the sound and what parameters these should be mapped to. In the picture, the gain-button has been clicked, and the "fixed" slider is chosen as the value for the gain. By clicking filters, a similar tab would replace the gain-tab, allowing to choose what inputs to map the filter frequency to. The green preview button in the bottom right corner plays back the sound with the current chosen settings.

effects are mapped to a fixed position, chosen by a slider visible on the screen. See Figure 5.2 for an example of the interface.

5.3 Evaluation

The initial evaluation was performed as an informal *hallway test* where 5 Medialogy Students were invited to evaluate the non-functional prototype. The subjects were presented with the following task:

1. Choose the sound "8bit Winner " from the library
2. Set the gain to the fixed middle position
3. Play the sound

The test subjects were instructed to explain how they interpreted the interface at each step, i.e. an index card representing each screen in the interface (interviewer notes can be found in Appendix B.2, p. 87). Subjects were not presented with a new step, until they had acted on the preceding step. The the evaluation overall showed that the subjects appreciated the simple workflow, yet a few areas showed room for improvement. Most doubts were found in the first step, where the user could choose between the three options for loading a sound source (Figure 5.1, p. 38).

Choose sound source was not clear to all subjects. Namely *Paste URL* was unclear, as the text did not indicate that it was prompting for a URL for an online *sound* file. A suggestion was to simply change the wording to include the word *sound*. As the wording in the other areas were not completely clear either, it indicated the need for a reworking of the wording in all three buttons. One user suggested starting with showing the library and then only then show the options to paste URL or load a local file if the desired sound were not found in the library.

The "distance" option in the gain sliders was also confusing, which is though understandable, as no distance sensor was implemented nor explained to the subjects at this stage.

Chapter 6

Iteration 2: Functional Interface

This section describes the implementation of the first functional version of the interface. It is divided into three main sections: Back End, Front End, and Tangible User Interface (TUI), with the former referring to the server-side platform needed to make the front end available to the user's browser. The tangible interface refers to the physical interface, the *Sketch Box*, constructed to enable live sensor input to the system.

6.1 Back End

The back end was based on a custom web-server running on a Raspberry Pi connected to a standard consumer grade web router. The server was programmed on the Node.js platform (Appendix A.4), with the Express framework[1] serving the files necessary to run the web application directly in the browser. The server consisted of three main parts, accessible on separate ports on the network:

1. A *static web server*, powered by Express.js served the static files and scripts including the CSS style sheets, HTML pages and JavaScript files used to control the Web Audio API). (Appendix A.1)

2. A *WebSocket* connection was included to provide a real time full-duplex TCP connection between browser and server. This was required to communicate live from sensor input from the physical interface to

[1]Express is a minimal and flexible Node.js framework for simplifying building web applications. `http://expressjs.com/`

the web browser. This feature was implemented using the *socket.io*[2] framework, with the server running on the same port as the static file server.

3. A standard *TCP socket* was opened on a separate port, and was used to receive data from the Arduino-based physical interface. Data received from the Arduino was immediately emitted on the WebSocket connection, making the data directly accessible in the front end browser application.

6.2 Tangible User Interface

The TUI (Figure 6.1) was designed to let the user use sensor data input to control the sound manipulation parameters in the web application. In this first iteration, three sensors (Figure 6.2, p. 44) were included:

1. A *potentiometer* (P160KNP2H) was included to let the user control the filters and gain with an analog input.

2. An infrared *distance* sensor (SHARP 2D120X F 05) was implemented as another analog input to let the user control the filters and gain. This input differed significantly from the potentiometer by not requiring physical contact to provide input. The user would be able to change the sensor value reading by moving their hand, the object or the *Sketch Box* itself, changing the distance between distance sensor and object.

3. Two *push buttons* were installed at two different locations on the *Sketch Box* to allow triggering the sounds without direct interaction with the browser.

The TUI was powered by a Microduino[3] equipped with the Microduino CC3000[4] module for wireless communication. The Microduino was flashed with a piece of software , reading the sensor data every 50ms and sending it over the TCP socket to the server in the JSON[5] format, with a string-value pair corresponding to each sensor name and the value reading, e.g. "$btn1$" : 1.

[2] `http://socket.io/`

[3] The Microduino platform is a small micro-controller (measuring 25.40mm x 27.94mm) compatible with the Arduino System. It is open-source and a series of stackable shields are available. `http://microduino.cc/`

[4] `http://www.microduino.cc/CC3000?filter_name=cc3000`

[5] JavaScript Object Notation `http://www.json.org/`

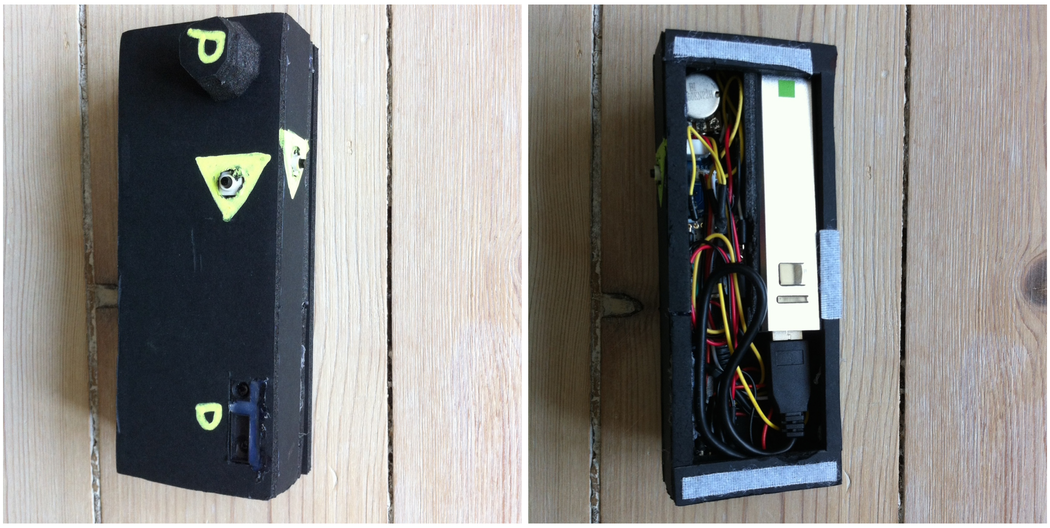

Figure 6.1: The first iteration of the Tangible User Interface. The push buttons are marked by a triangle, the potentiometer by a P and the distance sensor by a D. The aluminium object to the right is a USB battery pack, a Microduino with a wifi-shield is located below the wires.

The entire system was enclosed in a box constructed in black 0.5mm foam-board, measuring 60mm x 40mm x 150mm. A "black box" interface was chosen to encourage the users to experiment with the interface and manipulate the sounds without speculating what hardware was used in the system and how it worked. The system was intended to be used "as is".

6.3 Front End

The application was presented as a web site to the user. The application was written entirely in HTML, CSS and JavaScript with no special plugins needed to run.

CSS alone was used for all graphics. No image files included. This ensured minimum load times on the graphics side.

HTML provided the basic skeleton of the webpage. A single page layout, with dynamically visible and hidden content was chosen to avoid having to reload the webpage. Reloading the webpage would also reload all the scripts, causing all settings having to be configured again. This functionality was achieved using the HTML $< div >$ tag, which denotes a division or section

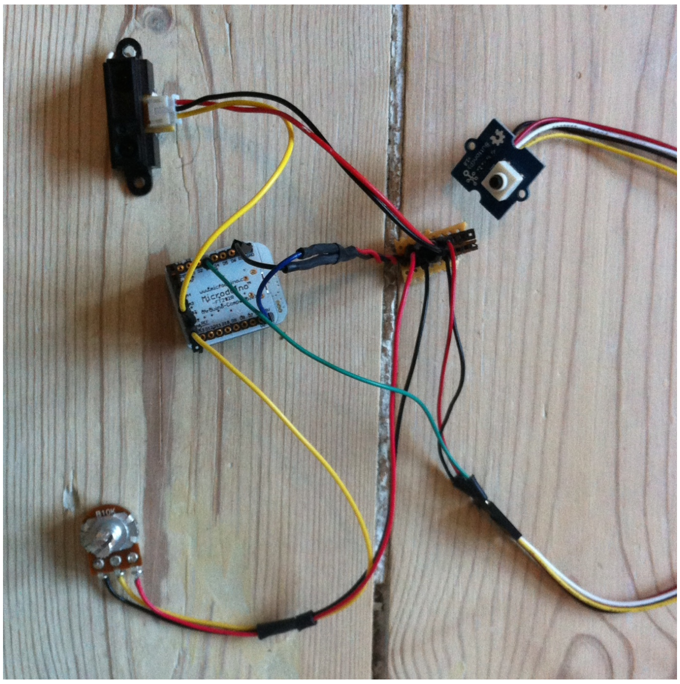

Figure 6.2: The hardware enclosed in the black box TUI. It included a distance sensor, potentiometer, two push buttons (only one in the picture) and a Microduino Core with a CC3000 WiFi module.

in an HTML document. The size, position, colour, visibility and many other parameters of the div can be defined in the CSS. The visibility was toggled on and off based on user input (e.g. when the gain-button is clicked, the div including all the relevant options become visible, while the div with the filter options are made invisible. This ensured that no reload of the webpage was needed.

Web Audio API was the platform for all audio playback, as it runs in most new browsers, including mobile browsers. The Web Audio API was required to enable full control of gain, playback and add filters to the sound sources. The audio graph consisted of three main nodes: (1) Sound source, (2) gain node, (3) filter node (Figure 6.3, p. 45).

1. The *Sound Source* can be any sound file that can be loaded into a sound buffer by the browser. It might be a link to a file available online, a file located on the users own hard drive or a sound from the *sound library* on the server, which is in reality just hyperlinks to the file paths on the server placed in the HTML file. Upon choosing a sound source, the application created an ArrayBuffer to ensure fast playback of the sound file. Once loaded, the application ran a PlaySound() function, which created a buffered sound source, assigning the content of the aforementioned ArrayBuffer to it, connecting the source to gainNode

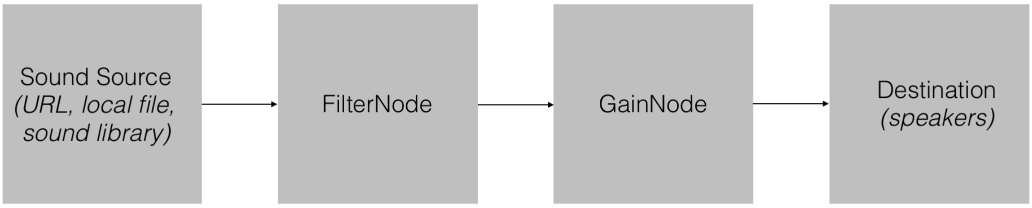

Figure 6.3: The audio graph consisted of the sound source, which was connected to the gain node, the filter node and then lastly to the destination (the speakers of the system).

and started playback.

2. The *FilterNode* was initiated by creating an instance of a bQFilter() class, which took as argument what type of filter should be created. In this iteration the basic "user-friendly filters" [15] LowPass, HighPass, BandPass were made available. After creating the biquad filter of the chosen type, the sound source was first disconnected from the GainNode, then connected to the FilterNode and the FilterNode finally connected to the Destination (speakers). The middle frequency of the filter was controlled by user input in the GUI if the user chose *fixed* or by the sensor readings from the sensors in the *Sketch Box*, received over WebSocket if the user chose either of the sensor input sliders as the mapping parameter.

3. The *GainNode* was created when the browser loaded the AudioPlayer.js script the first time (when the page is loaded) and connected to the Destination. The gain multiplier (a value between 0 and 1) was controlled by user input on a slider in the GUI (if the user chose *fixed*, or by the sensor reading[6] received via the WebSocket connection, which was then scaled to the 0-1 range.

6.4 Implemented Features

The functional prototype (Figure 6.4, p. 46) was a limited version of the non-functional one described in iteration 1 (Section 5, p. 37) in this iteration included the steps (1) New project, (2) Library, (3) Input mapping.

[6]The analog value-range depended on the sensor used, e.g. the distance sensor and potentiometer provided different ranges

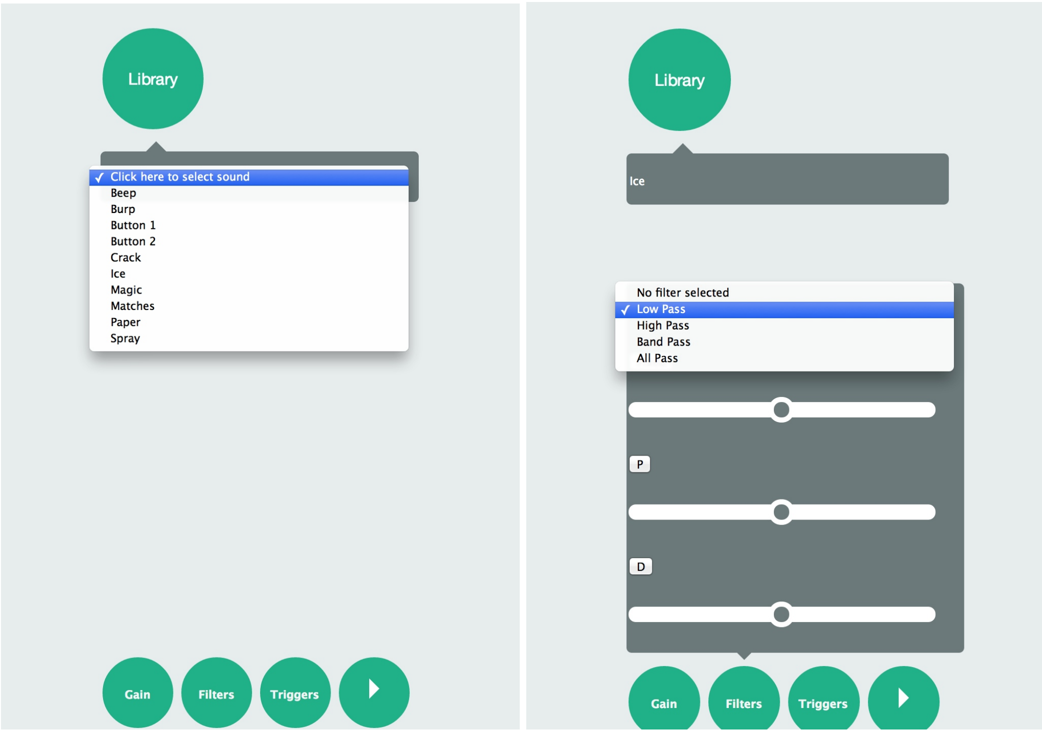

Figure 6.4: Left: The users could select a sound from a list of predefined sound sources on the server. Right: Four filters (Low Pass, High Pass, Band Pass and All Pass) were available to choose from. The sliders below P and D showed live input from the sensor reading.

(1) New project was the first screen shown when opening the application. To start, the user could press the *New project* button. *Load project* was not included, as the functionality was not implemented yet.

(2) Sound library was the next step available for the user. It included a series of sounds available for the user to choose from. *Paste URL* and *Local sound* were not included at this point.

(3) Input mapping was the third step available to the user, in the form of *Gain* and *Filters*. When a user pressed either of these, a tab would show up on the display showing three options of mapping the parameter to user input, *fixed*, *P* and *D*, each of which were activated by clicking a button with the respective labels each over a slider (Figure 6.4, p. 46). Only one mapping could be made for each parameter (e.g. gain could be mapped to D and filter could be mapped to P (or D or fixed), but gain could not be mapped to D, P or Fixed at the same time).

6.5 Evaluation

A test was carried out to evaluate the current implementation. The focus at this stage was on the usability of the interface, namely if it was possible for users not familiar with the system to pick up with limited instructions and perform a simple task using the system. The users were tasked to *create a sound for a door that closes*. The setup consisted of a 13" MacBook Pro placed on a small standing desk, running the web application in a Google Chrome browser, and the TUI placed next to it (Figure 6.5, p. 48). The procedure was as follows:

1. The subjects were asked to answer a series of questions regarding their experience with design, prototyping and sound (questionnaire included in Appendix D.2, p.94).
2. The task was presented: *Design a sound for a door that closes*. The subjects were not given a specific time frame, but asked to contact the interview manager when they were ready to present a sound.
3. The subjects were asked to explain what they did (what sound did they choose and what parameters were mapped to what input). Finally they were presented with the last part of the questionnaire (Appendix D.2, p. 94), which contained questions addressing the usability of the interface.

6.6 Results

11 subjects (1 female, 10 males), aged between 23 and 44 (SD=6.905) participated in the test. The test was performed at the midterm exhibition at the university at dedicated stall in an open area, with subjects recruited among students and staff of the university (Figure 6.5, p. 48). The subjects were informed that they were free to use the GUI or the TUI, and that the TUI used wireless communication and could therefore be moved around. The subjects were furthermore informed about the building door next to the stall with the hope that they would include it as a prop or inspiration in the process.

As expected, the results showed that most subjects had prior experience in creating prototypes. The vast majority either agreed or strongly agreed, while only two disagreed, indicating that most subjects were familiar with prototyping.

Figure 6.5: Left: The test-stall was located in a large open area with several other students exhibiting their prototypes. Right: A subject is interacting with the distance sensor in the prototype.

A comparative statistical analysis between the GUI and TUI [7] was performed on the Likert Scale items *hard to use*, *responsive*, *useful*, *fun* and *annoying*. The four point scale contained the items (1:strongly disagree, 2: disagree, 3: agree, 4: strongly agree). A Wilcoxon Signed Ranks test showed no statistically significant difference between the two input methods, except on the *annoying* parameter ($Z = -2.236$, $p = 0.025$), with median 1.5 in the TUI and median 2 in the GUI. This indicated, that the TUI caused a bit less nuisance when directly compared to the GUI.

Most of the subjects started by browsing the library exploring the sounds as inspiration for the design task, resulting in sounds ranging from big old creaking fairy-tale doors to science fiction space ship doors, and everything in between. However, it seemed as a general trend that most participants, with the task of designing a door sound in mind, listened carefully to the sound, attempting to find a sound that *could* sound like a door. Some preferred the short sounds, which surprisingly positioned the burp sound as one of the mostly used sounds, while others preferred more "composite" sounds, such as the matches, which consisted of the short sound of the matches being struck followed by the longer phase of the ignition and the wood catching fire. A few subjects asked for a larger selection of sounds, while the vast majority appeared to make do with the available sounds. The participants thus seemed to understand and utilise the concept of Foley, in the sense of finding existing

[7] questions 11 + 12 in the questionnaire Appendix D.2, p.94

sounds that *could* sound like a door, though in a crippled sense, since there were no options to record or add custom sounds. Regarding missing features, the following requests appeared, mainly suggested by the subjects experienced with sound design:

- Control the sound envelope
- Multitrack playback including placing the sounds in time
- Modulation based on physical properties of the sound, e.g. size (large, giant) or material (e.g. wood)
- Loop sound option
- Visual confirmation of settings (e.g. gain mapped to "D")

6.6.1 Discussion

The results indicated a tendency towards designing a sound for how a traditional door sounds, rather than how a door *should* sound, which was most likely a result of the very open task definition. This was not thought to have a significant impact on the tests validity in terms of assessing the usability of the current implementation, but did call for a more well-defined task in future iterations, and perhaps even more a relevant task in form of basing the scenario on a product calling for interaction sounds. It was rather disappointing that no subjects used the door as a prop in the process; most participants remained stationary next to the laptop, using the Sketch Box as a mean of tweaking the parameters, rather than using it *situated* for prototyping the sound in context of interaction with an actual door. There could be several reasons to this, but as one subject noted the option of physically toggling between different effect mappings on the *Sketch Box* itself might encourage moving away from the graphical interface even more. Other possible reasons could be that the *Sketch Box* was simply too big, and did not afford being placed so it could be triggered by the door. The physical construction of the TUI might even have afforded this behaviour even further, as especially the potentiometer is ideal for being twisted by hand, and might just have served as an alternative to the soft-sliders used for setting the gain or filter frequency. A hope was, that the Sketch Box would e.g. have been placed on the door, using the distance sensor or push button to sense the *interaction with the door*. In future iterations, this behaviour could perhaps be encouraged by providing means of fastening the Sketch Box to objects (e.g. velcro or other adhesives), providing external sensors (e.g. distance sensor on an extension wire) and probably most importantly by defining a more suitable task encouraging those activities and situated prototyping even more.

From a usability perspective, the foremost concern was the lack of visual confirmation when clicking the "P", "D" or "Fixed" buttons in the GUI to assign these to controlling the gain or filter sliders, leaving the only feedback of the actions in the sonic modification of the sounds. It proved difficult for many subjects to remember what parameter was mapped to what slider. Suggestions for improvements included toggle-buttons (on/off), colouring the slider or simply removing or "greying out" the sliders not in use.

In the original plans, the "choose sound source" was included as a separate step before showing the gain and filter options. In this iteration, the library option was placed in the "main" windows together with the gain and filter in order to use the preview button at the bottom right corner. It was observed that the subjects made great use of this, judging from the large number of times most subjects chose a new sound in the library.

As a concluding remark, the majority of the subjects reported having fun using the interface (especially the physical interface).

Chapter 7

Iteration 3: Design a Sound

In this section, the third iteration of the product design is described. The main focus in this iteration was to evaluate the system in connection with performing a more specific sound design task. In the previous iteration a series of issues with the UI were discovered, which were addressed before presenting it to prospective users. Deciding on the fidelity of the prototype before presenting it to prospective users is always a balancing art. In this iteration, the users are presented with a relatively high-fidelity prototype graphically. While, on one hand, this might cause users to hold back criticism to the interface, as users tend to view higher fidelity prototypes as less susceptible to suggestions for improvements, the fidelity of the GUI was intended to minimise nuisance with especially GUI issues stealing focus from the sound driven design. On the other hand, the fidelity of the audio processing backend was kept lo-fi, which was firstly to avoid the risk of introducing too many unnecessary features, secondly to observe how non-sound designers used this very limited available tool set, and thirdly to invite users to provide suggestions for new features, with the latter reason pointing back to the first reason.

7.1 Updated Product Requirements

Based on the findings in the previous iteration focus was placed on improving the following areas:

1. Encourage situated sketching
    (a) Clear evaluation task
    (b) Task-appropriate sensors

2. Functionality
    (a) Clearer visual feedback for button presses and sensor-mappings in GUI
    (b) Allow users to record and upload their own sounds

While the previous iteration resulted in several valid suggestions for new features, such as multi-track playback and amplitude-envelope control, these were regarded relatively advanced features and thus not staged for immediate implementation: the focus in this iteration remained on a simple implementation including only the most basic features to avoid the risk of presenting a tool too close to traditional sound production tool to non-sound designers.

7.2 Implementation

A series of changes to the *Sketch Box* were performed according in order to make the interface more suitable for situated sketching. The interface was redesigned, with the task of designing a sound for a door entry chime for a hotel lobby open 24 hours a day, able to change according to time of day (see Appendix D.1.1, p. 93 for a more elaborate description).

The focus in this iteration was on modifying the GUI in order to improve the usability of the system. The underlying functionality (filters, gain etc.) of the interface remained the same as in the previous iteration, with the audio graph (Figure 6.3, p. 45) unmodified. The audio graph remained the only part of the system not undergoing major revisions in this iteration.

7.2.1 Visual GUI feedback

The most prominent GUI changes were found in the sensor-parameter mappings, where the focus was on providing clear visual feedback when users configured a sensor-parameter mapping. In the previous version, clicking the software buttons to map the gain or a filter to a sensor-input provided very little visual feedback to the user (the standard HTML button elements were suppressed and released confirming the button was clicked), but after that the users were only able to identify the mappings by listening to the sounds. New visual confirmation of the mappings were implemented with the goal that the user should be able to visually confirm what mappings had been made when browsing the *gain* and *filters* menus. The mappings are best described visually, as seen in Figure 7.1, p. 53.

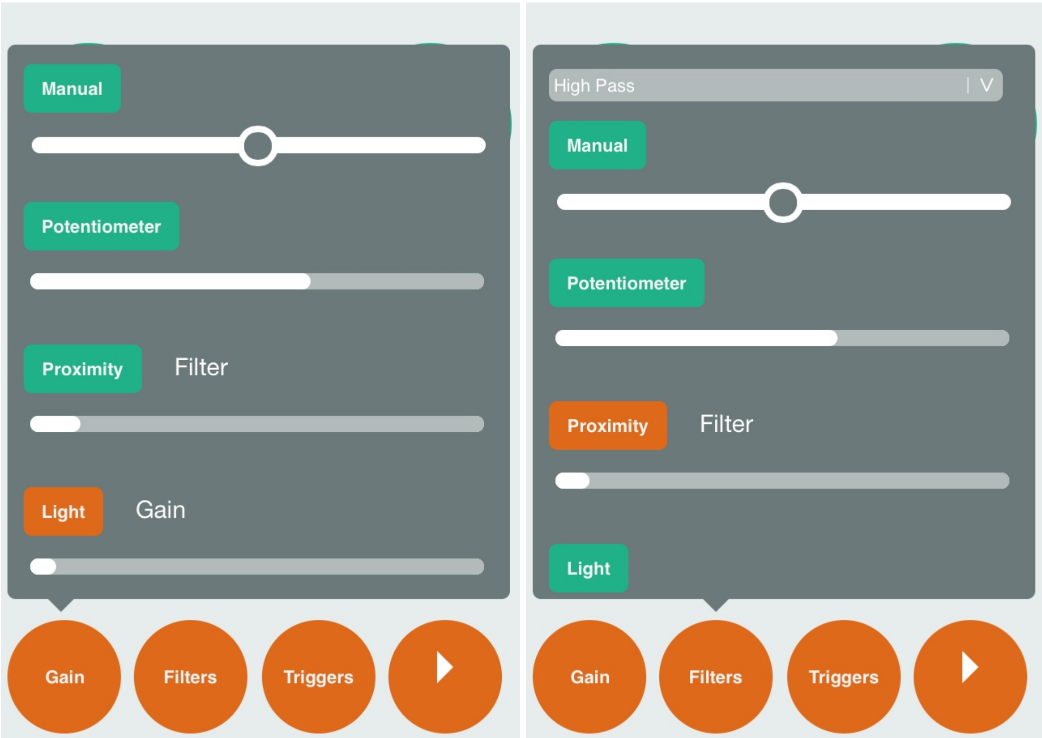

Figure 7.1: Left: Gain is mapped to the light sensor, causing the gain (or volume) of the playback to change according to amount of light registered by the *Sketch Box*. Note the *filter* tag next to the proximity button, indicating that a proximity-filter mapping has been configured in the filter-menu. Right: The proximity sensor controls the frequency of the high pass filter. When the *manual* button is pressed (in either of the menus), the sensor-mappings are cleared and the parameters are only controlled by the GUI slider thumb.

The slider thumbs seen on the manual sliders in Figure 7.1, p. 53, used to change the slider value of the gain or filters were now only available on the manual sliders, which are the only sliders where users are able to set values manually in the GUI. The remaining sliders were changed to the HTML progress bar element, as an attempt to more clearly indicate that the values represented input from the sensors in the *Sketch Box*, and thus not available for direct manipulation in the GUI. The sensor reading is represented by the white bar relative to the maximum value available represented by the underlying grey bars.

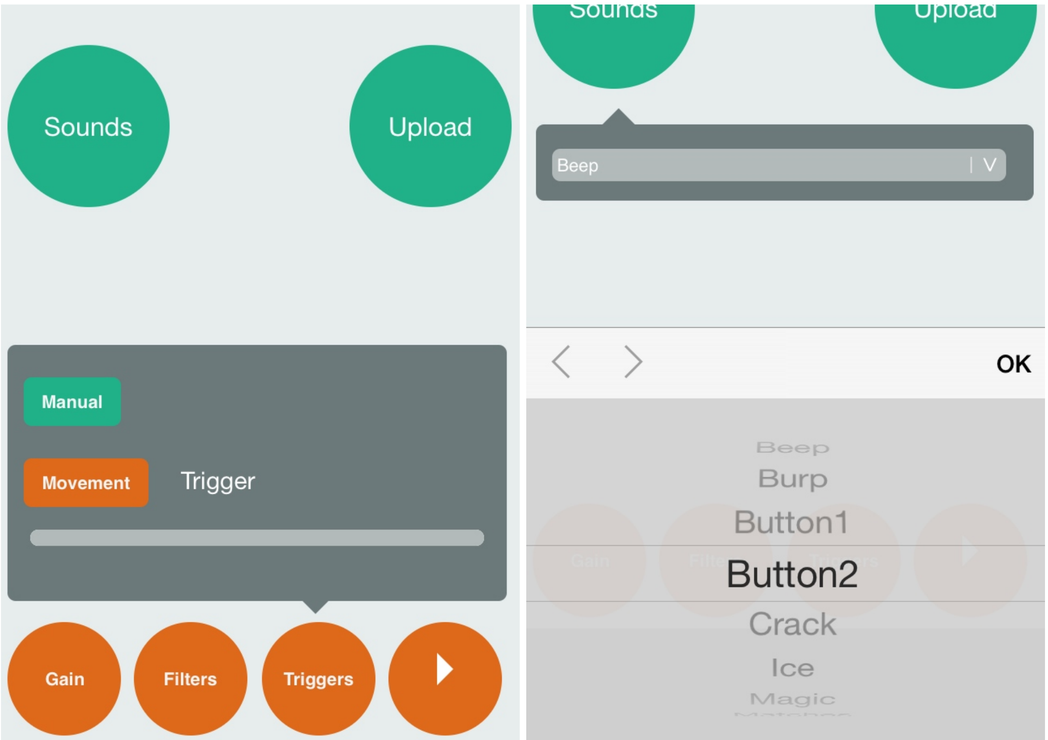

Figure 7.2: Left: The movement sensor is configured to trigger the playback of the sound. The grey bar denotes that no movement is detected - when movement is detected, the entire bar turns white. Right: The native iOS shows the list presented to the user after clicking the drop-down menu to replace the current sound "Beep" with "Button2".

7.2.2 Uploading and recording sounds

An upload function was implemented in order to let users add their own sounds to the library. Implementing this function required the following functionality:

1. Browse and choose sound file to upload
2. Inform the user about the upload progress
3. Add the newly uploaded sound to the sound library list

1. A HTML $< \mathit{input} >$ element of type=file accepting only audio-files was configured to allow the user to browse local files and choose what file to upload. Selecting a file to upload automatically triggered the upload process.

Server-side the Formidable[1] Node.js module was configured to handle the upload request and copy the received file to the correct directory on the server.

2. The server was programmed to continuously send JSON messages containing data about the current upload progress over the web-socket connection to the browser. The JSON messages included bytes received and bytes expected, allowing a client-side script to calculate the current progress. A client-side script listened for the values from the server, triggered a progress bar to appear in the GUI to inform the user that an upload was in progress.

3. When selecting a sound for upload an ArrayBuffer was immediately created so the user was able to use the sound immediately without waiting for the upload progress to finish. The sound should however still be available for the user in the library next time browser was opened. In order to avoid loosing the sound file upon closing the browser, the HTML5 LocalStorage[2] functionality was configured to save an updated list of the sound library. As LocalStorage uses the JSON format to save data in the browser, the updated sound library was saved in the format:

$\{"URL_1" : "Name_1", "URL_2" : "Name_2", ..., "URL_N" : "Name_N"\}$

Each time a new sound was uploaded, a new URL-Name pair was generated using the file-name, added to the sound library object after which a function updating the list in the GUI and saving the JSON object to the LocalStorage in the browser. This caused the updated sound library to be available on that particular device until the user cleared the browser data.

7.2.3 Recording Sounds

The upload functionality proved problematic to implement, as the locked functionality of especially iOS currently does not support access to the microphone from the browser, apparently due to security reasons. Uploading other files than images from the web-browser is currently also not supported on iOS, and it is thus not possible to record a sound in the native iOS sound recorder (Voice Memo) and upload it to the *Sketch That!*web app. The only option for sharing the audio recordings without connecting the iPhone to the computer is to send them by e-mail, which could be considered in a

[1] `https://github.com/felixge/node-formidable`

[2] `https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM/Storage#localStorage`

future implementation, as it require programming a mail-server to receive the e-mails, retrieve the files and update the list of available files in the browser.

Due to the aforementioned issues, it was chosen to explore the possibilities of the Android system. A *Samsung Galaxy Nexus* smartphone was acquired, and provided highly useful, as clicking the upload button in the GUI provided direct access to either choose a file from the smartphones file system or directly launch the built-in voice recorder. It was thus possible to record a sound and upload it directly to the web app, as seen in Figure 7.3, p. 56.

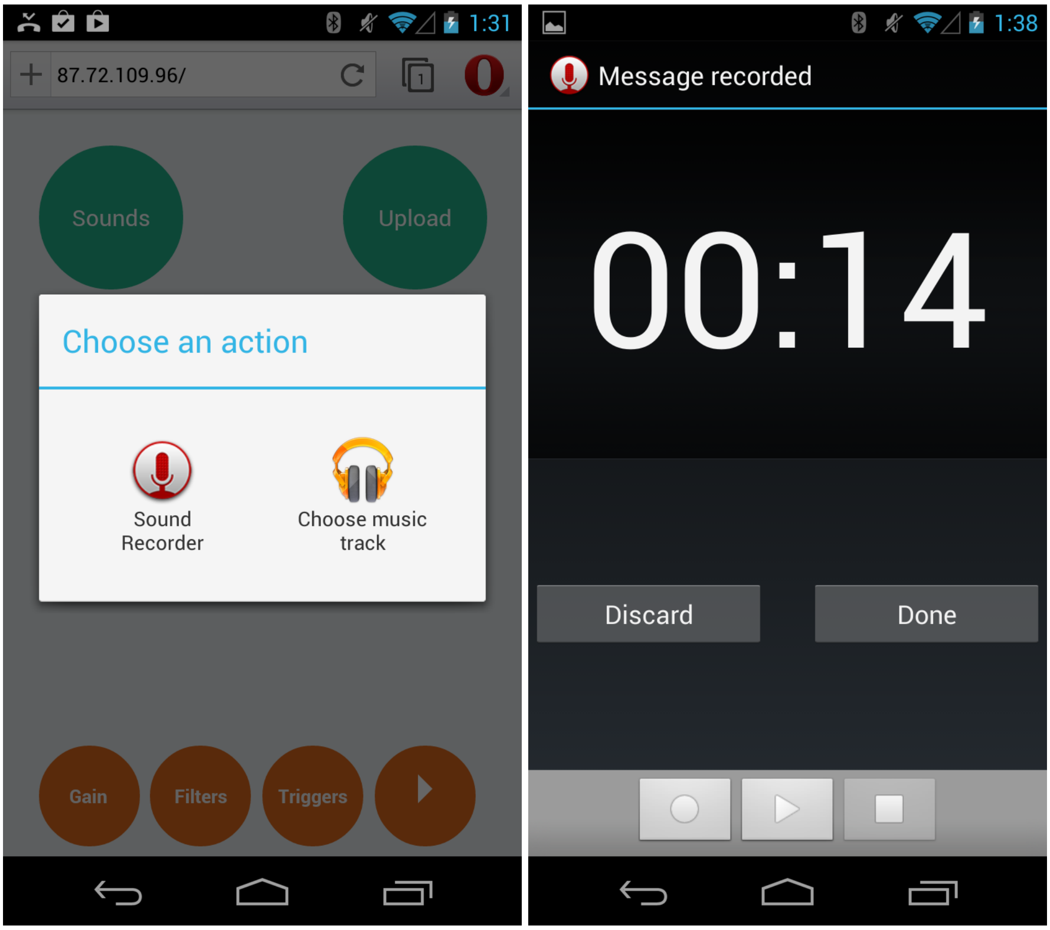

Figure 7.3: Left - Clicking the upload file button in the browser gave direct access to a dialog where the user could choose between uploading a file or launching the Android Sound Recorder application. Right - The Android Sound Recorder. Clicking done caused the file to immediately upload to the sound library.

This functionality was though not available on all Android smartphones. The added functionality of the sound recorder though came at a price -

Android has a bug causing sound from the Web Audio API to play back from the relatively weak earpiece speaker used when performing a phone conversation, and not the more powerful media speaker available on most phones to play back music and videos. This problem did not exist on the iPhones. As the playback volume was too low to be useful the phone was connected to a more powerful external speaker, significantly reducing the portability of the system. This is an issue that will need to be addressed in possible future iterations.

7.2.4 Sketch Box v2

The *Sketch Box* was updated with new sensors and a new casing. A PIR (Passive InfraRed) sensor was implemented and programmed to be available in the GUI as a trigger-mapping option (see Figure 7.2, p. 54). The users were thus able to use the sensor as a trigger to start sound playback when movement was detected around the sensor. A LDR (Light Dependent Resistor) was also implemented and programmed to appear next to the existing proximity and potentiometer mapping options in the filter and gain menus (see Figure 7.1, p. 53).

The electronics were implemented in a new enclosure (see Figure 7.4, p. 58) laser cut from 4mm oil tempered hard board. This provided a more durable and sturdy enclosure than the previous foam-board based enclosure. The basic cutting plans were created using the free online box-plan generator makercase[3], and modified in Adobe Illustrator to include custom mounting holes for the electronics. See Appendix D.3, p. 98 for laser cutting plans. The enclosure measured 35x55x175mm in external dimensions.

7.3 Evaluation

The main focus in this iteration was, apart from addressing the issues identified in the previous iteration, to present a specific sound design task to potential users. A qualitative evaluation was performed by presenting the interface (GUI+TUI) to the test subjects with instructions to carry out a specific design task.

The task description from the previous iteration was revised (see Appendix 2, p. 93), and the participants were now required to design a door entry chime to inform the clerk in a hotel lobby, open 24 hours a day, when guest entered the lobby. Subjects were encouraged to use the sensors in the *Sketch Box* to explore changing sound character based on the mappings, specifically

[3] `http://www.makercase.com/`

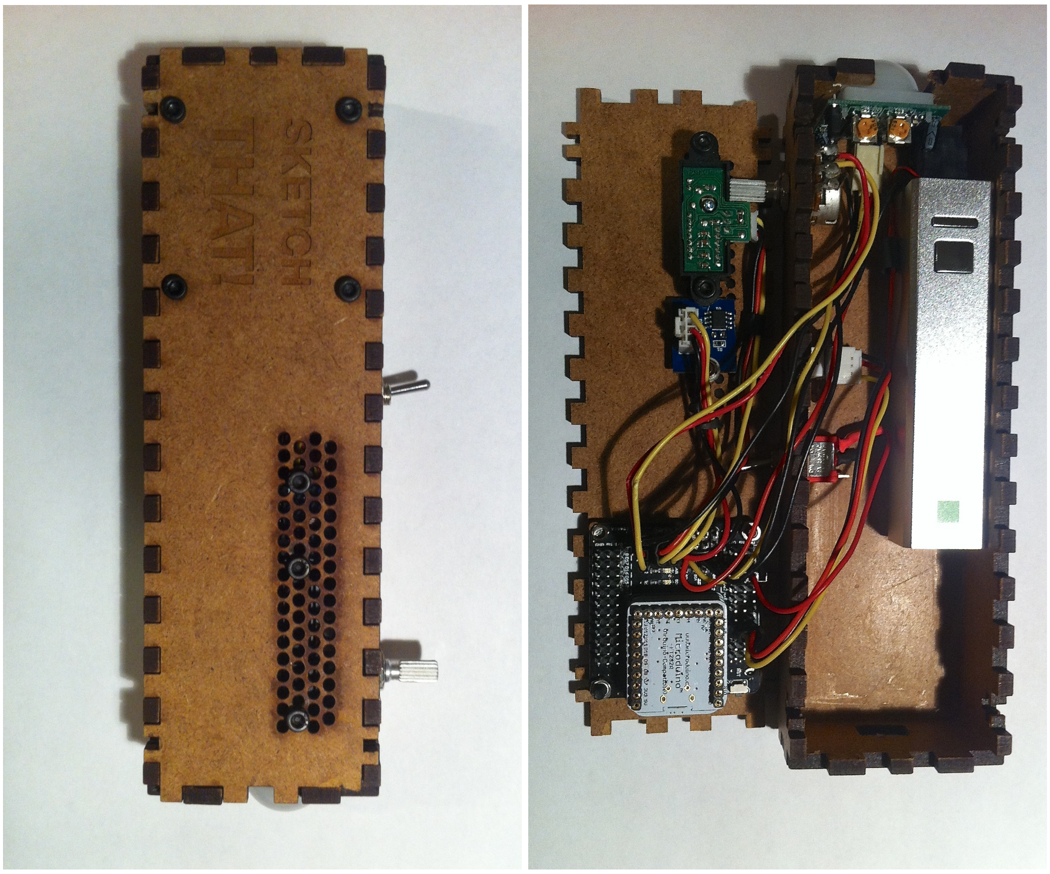

Figure 7.4: Left - the redesigned *Sketch Box* cut from 4mm oil tempered hard board. The LDR and Proximity sensors are hidden behind the perforated front plate. The toggle button on the right side is an on-off switch, and the bottom-right is the potentiometer. At the very bottom part of the PIR sensor is visible. Right - *Sketch Box* on the inside with the big battery pack mounted next to the sensors and micro controllers.

that the sound should change character based on time of day. The task was specified to encourage the subjects to explore the sensors available in the *Sketch Box*, e.g. by mapping the light sensor to the gain to make the volume change based on the amount of light around the *Sketch Box* or mapping one of the filters to the light sensor. The subjects were not specifically instructed to use the light sensor in order to give the subjects as much creative freedom as possible, and in order to observe how the subjects explored the available sensors in the sketching process and observe how the subjects interpreted the available input and control options (filters, gain, triggers, sensors). A more elaborate description of the test procedure can be found in Appendix D.1.1,

p. 93. They were introduced to the interface and available sensing devices (proximity, light and movement) and encouraged to use them creatively in the sketching process.

7.4 Results

17 subjects (4 females, 13 males) participated in the evaluation, aged between 20 and 52 (SD = 7.713). The evaluation was divided into 13 sketching sessions, with 3 of the sessions being groups and the remaining 10 single subjects. The subjects were recruited on the campus, representing the disciplines Medialogy, Interaction Design, Service Systems Design, Sustainable Design and Architecture. The participants were offered the chance to win a pair of movie tickets in return of participating in the evaluation.

7.4.1 Encouraging situated sketching

The revised TUI appeared to encourage the subjects to perform *situated sketching*. Apart from the newly added sensors in the Sketch Box, this is most likely also a result of the refined design task description.

The majority of the subjects used the interface actively to perform *situated*sketching, however to different degrees. Most subjects were quick to understand the movement sensor, and used it actively to trigger the sounds based on movement. While most subjects triggered the sounds themselves, a few subjects took it even further, and brought the tools out of the test location to use passers-by in the open campus areas as sound triggers.

The *situatedness* of the sketching sessions were rated according to the following conditions:

1. The PIR[4] trigger was configured and tested at the door
2. The LDR[5] was configured
3. Created Foley recording

Each session was awarded one point for each of the aforementioned conditions observed in the sketching session, and the results are illustrated in Figure 7.7, p. 62. The observations showed that around half (7 of 13) of the sessions met all three conditions, gaining the score 3 in the *situatedness*. Only two sessions resulted in a score of 0, meaning that they in no way attempted to

[4]Passive InfraRed sensor. Can sense movement of people, objects or animals
[5]Light Dependent Resistor, changes resistance based on level of light

Figure 7.5: Left: One subject used the sound recorder to directly upload his own sound. Right: *Sketch That!* "in the wild", using passers-by to trigger sounds.

use the features of the interface to perform situated sketching. The majority of the subjects were Medialogy students, which is an important note, as the subjects in this category can be assumed to possess technical know-how and experience with sensors technology and user interface design. When grouping the Medialogists into one group and the remaining disciplines in another group, while taking into account that the Medialogists represent 8 of 13 total sessions, it still appears that most sessions gained a high score in the amount of situated sketching performed. The subjects were divided into the two groups of *designers*and *Medialogists*. Even though the Medialogy education is indeed concerned with design, this distinction was made due to the aforementioned point of the Medialogist education being generally very technical, with students highly familiar with programming, hardware and software prototyping.

A graphical comparison of the profiles (Medialogists and designers) can be seen in Figure 7.6, p. 61. A Mann-Whitney U test showed no significant difference between the groups regarding the experience with designing sound, though the Medialogists appeared to be more experienced creating prototypes ($U = 29.5, ns$), and more focused on the technical part of the design($U = 22.5, ns$), whereas the designers seemed to consider sound earlier in the design process ($U = 24, ns$).

In 10 of 13 sessions the subjects added new sounds to the library, being it Foley recordings, vocal sketches or sounds found on the web (e.g. subject 16

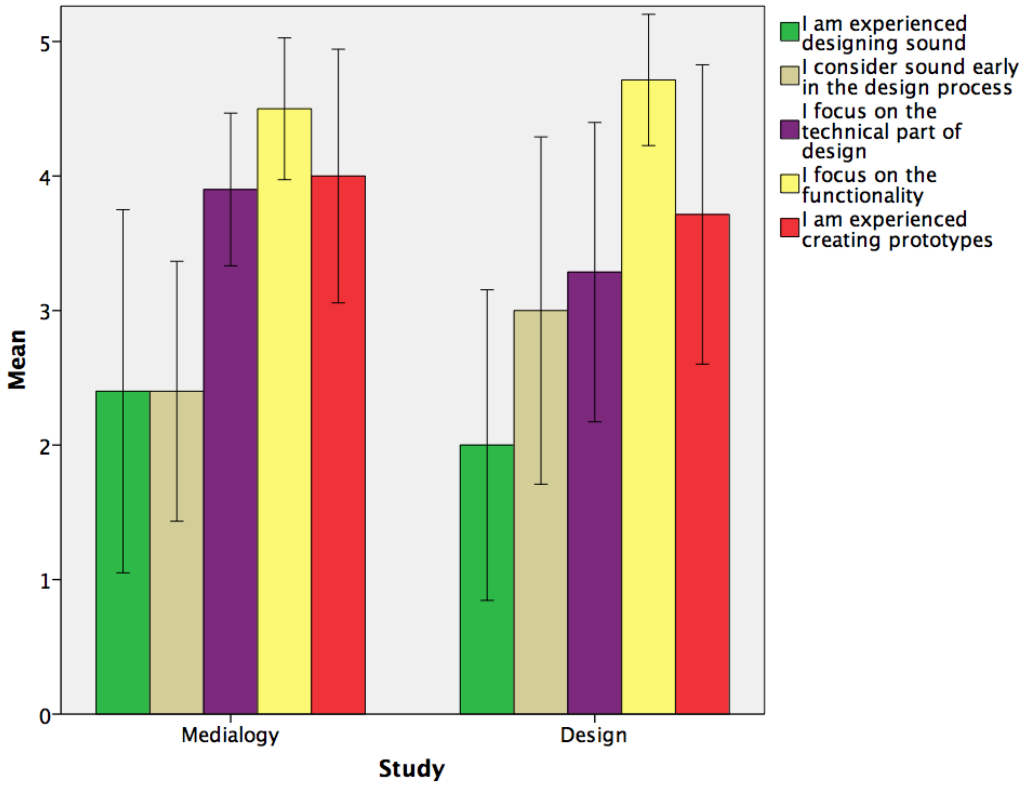

Figure 7.6: Comparing the profile of the designers vs the medialogists showed no significant differences between the groups.

found a bird-sound).

As a final observation in the *situatedness* of the sketching 2 of 3 of the sketching sessions performed by groups scored 3 points in the situatedness of the sessions. The last group, receiving 0 points, had major difficulties navigating the interface, and only managed to browse the sound library concluding that *magic* was the least useless sound. The group refused to vocal sketch or record their own sounds, as no musical instruments or obvious sound producing objects were available on the test location. This is perhaps an indication of the issue of social discomfort related to performing in front of others, and it might be caused by the interviewer being present in the room while the subjects used the interface.

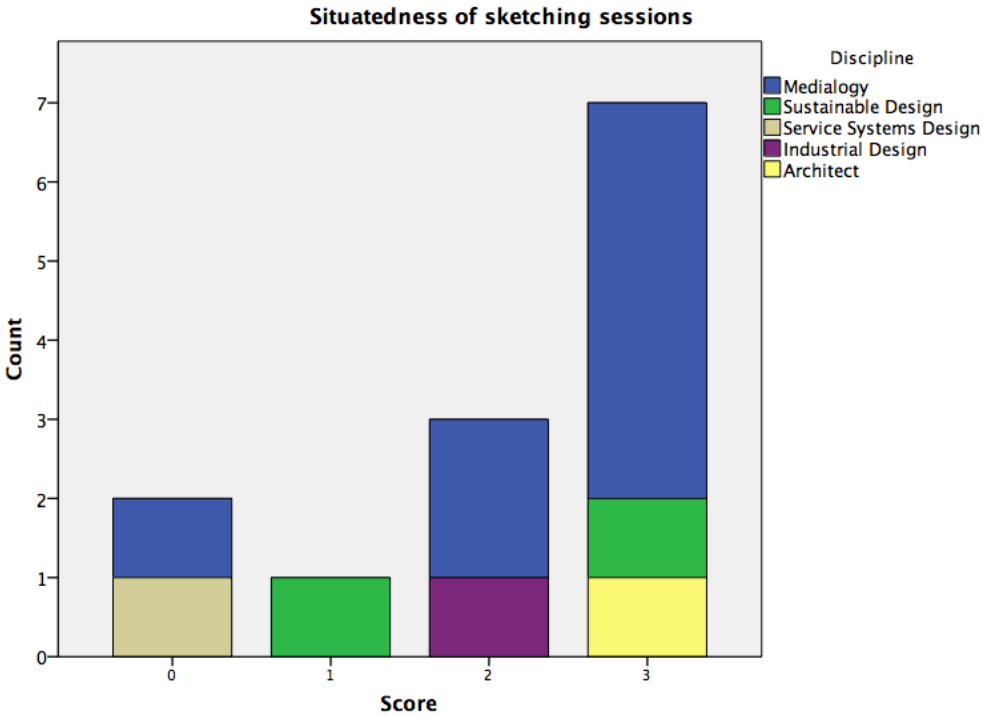

Figure 7.7: Each session was awarded with a score for the *situatedness* of the sketching based on how many of the situatedness criteria were met. The x-axis represents the three point options (0, 1, 2, 3) and the height of the column denotes how many sessions reach each score. There was a total of 13 sessions. The bars are furthermore divided by the respective disciplines of the subjects.

7.4.2 Functionality

A main requirement was to improve the visual feedback in the GUI when the user defined a sensor-parameter mapping (e.g. gain mapped to potentiometer value). The observations showed clear signs of improvement as most participants did not express doubt as to if the button presses were registered. A few users though expressed a wish for improvements on the user friendliness of the interface, especially the GUI feedback when defining a sensor-parameter mapping. The meaning of the words (gain, filter or trigger) showing up when e.g. pressing the *proximity button* to map proximity to control a *filter* parameter was not clear to all subjects. One subject suggested using icons in stead, which was indeed the plan in the original GUI layout (see Figure B.3, p. 86) but unfortunately did not make it to implementation in this iteration.

The observations and conversations with the subjects during the evaluation indicated that particularly the concept of filters was unclear in many of the sketching sessions. Several of the more technical participants expressed the wish for more work to be placed in the *tuning* of the filters, more particularly that the scalings of the filters were confusing. On some of the filters (depending on the sound played back) the filter only had an effect on the extremes of the scales (when the slider was close to minimum or maximum value).

Examining the differences between the groups further, it was observed that most subjects in belonging to the more technical Medialogy group appeared to be relatively familiar with the concept of filters, sensor input and mappings. These concepts were less clear to the subjects belonging to the *less technical* disciplines. This became apparent as most *technical* subjects experimented with the filter mappings, having a clearer goal of the manipulation they wished to perform on the sound (e.g. get a "deeper" sound from a low-pass filter) and asked elaborate questions to the workings of the filters. The less technical designers mostly experimented briefly (if at all) with the filters and gave up. Being familiar with the concept of filters furthermore made it easier to discuss and explain the functionality, compared to the subjects less familiar with filters due to lack of common vocabulary. The concept of *gain* was furthermore not clear to all subjects, but was generally easily picked up when explaining it as *volume*.

7.4.3 User Experience

The user-experience of the interface was again measured using a set of questions regarding the users satisfaction of using the interface. Comparing the rating of the TUI (Sketch Box) with the GUI (browser-based) in a part of the tool Wilcoxon signed ranks test showed no significant differences in rated usability, measured on rated ease of operation, responsiveness, usefulness, fun, nuisance and confusion ($p > .05, ns$). The subjective ratings aimed at assessing the usability of the interface, sorted by Medialogists and designers can be seen in Table 7.8, p. 64.

A Mann-Whitney U non-parametric test comparing the subjective ratings of the interfaces with the type of designer as the independent variable, showed a significant difference in ratings on 5 of 12 points, namely:

1. The GUI was hard to use ($U = 10, p < .05$), with the *designers* rated the GUI as harder to use

2. The GUI was useful ($U = 15.5, p < .05$), with the *designers* rating the usefulness lower

3. The TUI was useful ($U = 7.5, p < .05$), with the designers rating the usefulness lower

4. The TUI was fun ($U = 9.5, p < .05$), with the designers rating the TUI as less fun

5. The TUI was confusing ($U = 9.5, p < .05$), with the designers rating the TUI more confusing

In the remaining categories, no significant differences were found between the groups. These results should though be noted in light of the relatively low and unequal sample size. That being said, there seems to be an indication that the interface is rated more positive by the Medialogists than the designers.

Lastly the subjects were asked to rate how satisfied they were with the sound they designed, if they thought the interface encouraged situated sketching and lastly the portability of the interface. The results can be found in Figure 7.9, p. 65. Comparing the two groups in a Mann-Whitney U test indicated that the Medialogists found the Sketch Box more portable than the designers ($U = 18, p < .05$). The remaining results yielded no statistically significant difference. Overall the interface was rated

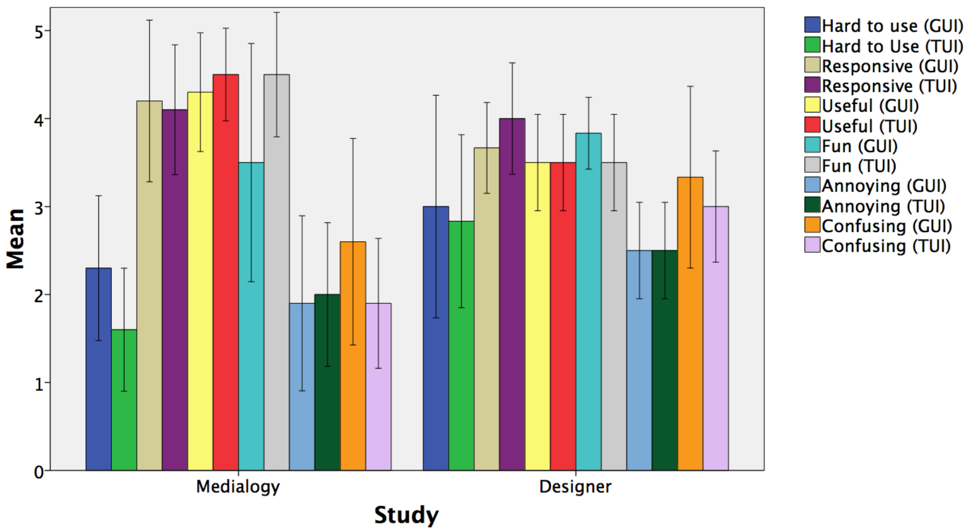

Figure 7.8: Comparison of the subjective usability ratings of the interface between the presumably technical group of Medialogists and less technical Designers.

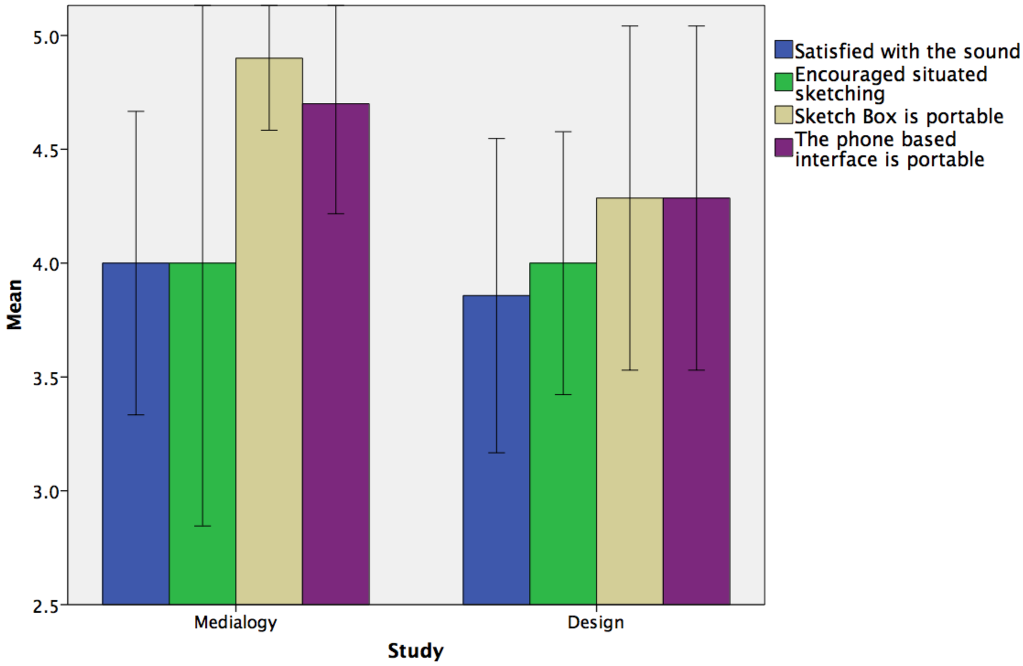

Figure 7.9: Comparison of the subjective usability ratings of the interface between the presumably technical group of Medialogists and less technical Designers.

7.5 Discussion

First of all, the shift from laptop to smartphone as the platform clearly encouraged more participants to use the interface to perform situated sketching, however still to a varying degree. The most prominent cases of situated sketching sessions included users bringing the interface with them, leaving the test location to record Foley sounds or include students in the hallways to test the triggers and chosen sounds. Including the option to record sounds directly in the interface appeared to encourage the subjects to adapt Foley-based techniques as well as Vocal Sketching when no appropriate sounds were available in the relatively limited sound library.

From the few subjects who chose to search online free sound libraries to acquire new sounds, the process of acquiring the sounds appeared slower and difficult, due to the large libraries, which takes time to browse, often result in sounds in different formats or wrong length, requiring additional editing in audio software before uploading the sound. This is a process that could perhaps be improved by linking to specific online libraries, and including a snipping tool in the software to quickly trim the sounds.

Judging from the evaluation, the large amount of work placed into improving the GUI had a positive effect on the usability of the interface. This was especially apparent in the area of the sensor-parameter mappings (gain-LDR, filter-potentiometer etc.), where less observations or comments indicated the users having difficulties registering if a mapping had been configured or not. Furthermore, the redesign of the *manual* and *sensor-based* sliders, more specifically removing the slider-thumb from the sensor-based values, seemed to help the users understand that the values were not to be directly manipulated from the GUI, as opposed to the previous iteration where several users tried pulling the sensor-based sliders to adjust the value.

There is still significant work ahead in improving the usability of the interface, with the issue of presenting relatively advanced technical concepts such as filters to non-sound designers perhaps as the most prominent. Work should be placed into finding a suitable vocabulary, as well as the correct visual representation of the control-parameters, being it in the form of appropriate icons, data visualisations (waveforms or spectrograms), or perhaps something completely new. Issues in this area include deciding on how much sound-specific vocabulary to include (or remove) as well as defining appropriate metaphors to explain concepts such as filters to non-sound designers. These issues could perhaps be addressed by listening tests, where non-sound designers are instructed to manipulate a sound using e.g. a low-pass filter (without specifically introducing the filter-terminology), asking the subjects to explain in their own words what happens to the sound. More work should be placed into evaluating the functionality, as well as considering if the concept of filters should be presented differently or if it is even a suitable manipulation parameter for non-sound designers.

A main issue after all is thus if it is possible to design an interface for users with *no* knowledge of sound design whatsoever, that is should the users be required to study basic concepts of sound and music computing as in the FFF (Section 2.3.1, p. 15) approach?

Chapter 8

Discussion

Sketch That! is still a work in progress. However the evaluation of the current stage of development showed a tool with potential in the area of situated sketching and prototyping. The tool could be used as a simple way to capture vocal sketching as well as Foley sounds, and quickly try coupling these sounds to interactions in a situated manner.

Vocal Sketches are usually captured using video recordings, if they are to be used as guiding elements in the design process, and this interface could thus be another way of capturing these vocal sketches at a higher fidelity. On one hand the amount of participants performing vocal sketching in the design process was though relatively low, which could be due to the earlier identified issues of social discomfort using this particular technique. On the other hand, the participants might simply have found the other sound sources more handy. This might be at the cost blurring the lines between sketching and prototyping, and this type of vocal sketching would perhaps fit better under the description of vocal prototypes (lo-fi or hi-fi). However, as stressed by Buxton, the definition of a sketch is rather a question of intent, than a question of the means used to create the sketch or prototype - if the tools are inexpensive and quick to use, it might still be possible to create plenty of timely, ambiguous, disposable, appropriately refined renderings of the ideas, keeping a minimum of detail, while still suggesting and exploring rather than confirming.

The research of current state of the art conducted while designing this interface identified several tools and methods for sound design, with the PSD (Section 2.3.2, p. 17) and FFF (2.3.1, p. 2.3.1) approaches in their respective disciplines of Industrial Design and Interaction Design as the most prominent. While these tools show great potential and are already being used in their respective educational design programmes, there appeared to be room for an attempt to make these approaches even more situated. It is believed that the portability and ease of use of Sketch That! holds potential to address this. The process used in this project could perhaps turn into a way of performing quick and dirty *functional* prototyping, thus perhaps relieving some of the issues pointed out in the From Foley to Function technique.

Sketch That! provided an easy to use interface, with no special software or plugins needed to run. The tool is portable and relatively easy to use, though work is still ahead to make the tool more attractive to the less technical designers. This conclusion though assumes, that the right tool for the right people will increase the interest in the area of non-sound designers.

Sketch That! is still in its infancy as a design tool, and the evaluations still suggested several areas of the UI with room for improvement - this holds for the TUI and GUI alike.

Chapter 9

Conclusion

This thesis was initiated with the following objectives:

1. Identify current trends in sound design and sketching
2. Increase the availability of sound design to non-sound designers
3. Encourage situated sonic sketching and prototyping
4. Identify relevant prototyping hardware and methods

The investigation indicated that this is an area in great development, with several emerging trends such as Sonic Interaction Design, which calls for new sketching, prototyping and evaluation methods. The traditional methods of *passive* sound quality evaluation sees new challenges with the advent of increasingly interactive products, perhaps especially in the area of wearable pervasive technology, hiding in plain sight in the increasingly daily use of small interactive, connected commodities. Some products, like the car or hearing aid might demand eyes-free interaction, while the eyes-free interaction developed in the emerging field of SID might set the stage for future categories of products. While controlled lab tests might be perfectly suitable for assessing intentional sounds non-interactive products, such as media players or the reduction of undesired consequential product sounds in industrially designed products, such as the electric razor, the implementation of intentional feedback sounds (continuous or discrete) in the otherwise silent electronic devices of the future require new means of sketching and evaluation.

A major challenge in the process was to find a way of encouraging situated sketching, as it requires an interface, which affords it, but also a suitable evaluation task to encourage it. The process constituted an attempt to provide an experience prototyping driven approach to sonic interaction design.

Several stages were observed in the evaluation of the system, where subjects performed situated sketching, using techniques such as vocal sketching and Foley recordings, some ending up even performing experience prototyping by moving the sketch away from the test location to test the interaction in hallway tests.

Given the sparse investigation of the current tools (or lack of tools) used by non-sound designers it is hard to conclude if the objective of making sound design more available to this target group was met. The evaluation though yielded positive results in terms of the usability of Sketch That! when tested on non-sound designers, even though the designers appeared slightly more skeptical towards the interface than the more technical group of Medialogists, this difference though not statistically significant. This comparison of the technical and less technical group is though interesting, resulting in the question of how much inherent knowledge the users of such an interface should be required to possess. Ideally the tool would be available with no prerequisites, which poses significant challenges in presenting relatively basic sound design concepts such as filters and gain in a jargon easily digestible to non-sound designers.

Even though the rather short term test performed to evaluate the current state of the interface yielded interesting and promising results, in only left the author with the urge to perform further work on this interface to see how it could perform in workshops or longer term evaluation with designers actively using *Sketch That!* as a tool for simple SID.

Chapter 10

Future Work

It would be interesting to perform a longer term evaluation of the system to see how the system compares with or complements previously proposed systems such as the *From Foley to Function* approach.

Another interesting area to investigate further would be the situated sketching and prototyping experience the mobile browser platform could potentially offer. Could this result in more frequent sound prototyping or new prototyping contexts? The mobile browser could for instance be utilised as a *cultural probe* in this evaluation, where the users could be encouraged to use the interface to sketch and prototype sounds in their daily lives. The strongpoint of the mobile browser is, that the users would be able to bring the interface with them, with little (or no) special equipment needed. Having the tools available on their own device (the smartphone) have several advantages, including the users not having to bring special equipment, plus it's a piece of equipment the users are likely to bring with them anyways.

The physical tool could be smaller and more portable, which could e.g. be achieved by building a modular setup. With small micro controllers, tiny sensing devices could be built and placed somewhere continuously sending data back to the web-app. These devices should be compact enough to place and conceal in existing products or prototypes to encourage sketching in the sonic exploration of product- and interaction design.

Chapter 11

Acknowledgments

Thanks to Mads Bredager for helping me out with the graphical layout. Thanks to Michael Schmidt and Flemming Koudal for proofreading the manuscript. Thanks to my supervisors for supporting the project idea and professional guidance through the process.

References

[1] Ehsan (Mozilla Foundation); Berkovitz Joe (Invited Expert); Bossart Pierre (Intel Corporation); Carlson Eric (Apple Inc.); Geelnard Marcus (Opera Software) ... ZERGAOUI Mohamed (INNOVIMAX) Adenot, Paul (Mozilla Foundation); Akhgari. Web audio api - w3c working draft 10 october 2013. URL `http://www.w3.org/TR/webaudio/`.

[2] John Allsopp. Html5 for creative directors. ebook, https://itunes.apple.com/dk/book/html5-for-creative-directors/id581827782?mt=11, November 2012.

[3] Rafael Ballagas, Faraz Memon, R Reiners, and Jan Borchers. iStuff Mobile: prototyping interactions for mobile phones in interactive spaces. *... Mobile Interaction Devices ...*, 2006. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.7811\&rep=rep1\&type=pdf`.

[4] M Bodden. Perceptual Sound Quality Evaluation. *Proc. Internoise*, pages 1–6, 2000. URL `http://www.product-sound.com/assets/images/intern\_00.pdf`.

[5] Markus Bodden. Instrumentation for sound quality evaluation. *Acta Acustica united with Acustica*, 83(January 1996):775–783, 1997. URL `http://www.ingentaconnect.com/content/dav/aaua/1997/00000083/00000005/art00008`.

[6] Eoin Brazil and Mikael Fernström. Auditory Icons. In Thomas Hermann, Andy Hunt, and John G Neuhoff, editors, *The Sonification Handbook*, chapter 13, pages 325–338. Logos Publishing House, Berlin, Germany, 2011. ISBN 978-3-8325-2819-5. URL `http://sonification.de/handbook/chapters/chapter13/`.

[7] SA Brewster, PC Wright, and ADN Edwards. Guidelines for the Creation of Earcons. *Adjunct Proceedings of HCI*, pages 1–4, 1995. URL `http://www.dcs.gla.ac.uk/~stephen/earcon\_guidelines.shtml`.

[8] M. Buchenau, I.S. Francisco, and J.F. Suri. Experience Prototyping. In *Conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 424–433. ACM, 2000.

[9] Bill Buxton. What sketches (and prototypes) are and are not. *Workshop at CHI06*, 2006. URL `http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:What+Sketches+(+and+Prototypes+)+Are+and+Are+Not+.\#0`.

[10] Bill Buxton. Design as a Dreamcatcher. In *Sketching User Experiences: Getting the Design Right and the Right Design*, chapter 1, pages 25–221. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007. ISBN 0123740371, 9780123740373.

[11] Bill Buxton. Sketches are not Prototypes. In *Sketching User Experiences: Getting the Design Right and the Right Design*, chapter 1, pages 138–141. Focal Press, San Francisco, CA, USA, 2007.

[12] Bill Buxton, Bill Gaver, and Sara Bly. Chapter 1 : NONSPEECH AUDIO. In *Audio UI*, volume 4, chapter 1, pages 1–7. 1994. URL `http://www.billbuxton.com/Audio.TOC.html`.

[13] Inger Ekman and Michal Rinott. Using vocal sketching for designing sonic interactions. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, pages 123–131. ACM, New York, NY, USA, 2010. ISBN 978-1-4503-0103-9. URL `http://doi.acm.org/10.1145/1858171.1858195`.

[14] A. Farnell. Sound language and knowledge. In *Designing Sound*, pages 110–114. University Press Group Limited, 2010. ISBN 9780262014410. LCCN 2009050741. URL `http://books.google.dk/books?id=YGymQwAACAAJ`.

[15] A. Farnell. User-friendly filters. In *Designing Sound*, page 215. University Press Group Limited, 2010. ISBN 9780262014410. LCCN 2009050741. URL `http://books.google.dk/books?id=YGymQwAACAAJ`.

[16] Hugo Fastl. Psychoacoustic: Basis of sound quality evaluation and sound engineering. *The Thirteenth International Congress on Sound and ...*, 2006. URL `http://www.iiav.org/archives\_icsv/2006\_icsv13/pdf/icsv13Final01113.pdf`.

[17] K Franinović and S Serafin. *Pedagogical Approaches and Methods*. MIT Press, 2013. ISBN 9780262018685. 125–150 pp. URL http://books.google.dk/books?id=54_uDvxQe94C.

[18] K. Franinović and S. Serafin. *Sonic Interaction Design*. MIT Press, 2013. ISBN 9780262018685. LCCN 2012028482. URL http://books.google.dk/books?id=54_uDvxQe94C.

[19] Jesse James Garrett. *The Elements of User Experience - User Centered Design for the Web and Beyond*. New Riders, second edition, 2011.

[20] Daniel Hug. Barking Wallets and Poetic Flasks: Exploring Sound Design for Interactive Commodities. In Stefania Serafin and Karmen Franinović, editors, *Sonic Interaction Design*, chapter 18, pages 351–367. MIT Press, Cambridge Massachusetts, 1 edition, 2013. ISBN 978-0-262-01868-5.

[21] Moritz Hug, Daniel & Kemper. From foley to function: A pedagogical approach to sound design for novel interactions, 2014. URL http://journal.sonicstudies.org/vol06/nr01/a03.

[22] Lau Langeveld, René van Egmond, Reiner Jansen, and Elif Özcan. *Product Sound Design: Intentional and Consequential Sounds*. In Tech, 2013. URL http://cdn.intechopen.com/pdfs/43375/InTech-Product_sound_design_intentional_and_consequential_sounds.pdf.

[23] Jonas Lowgren. *Interaction Design - brief intro*. The Interaction Design Foundation, Aarhus, Denmark, 2013. URL http://www.interaction-design.org/encyclopedia/interaction_design.html.

[24] Bill Moggridge. Introduction. In *Designing Interactions*, pages 1–14. The MIT Press, 2007. ISBN 9780262134743.

[25] Bill Moggridge. People and prototypes. In *Designing Interactions*, pages 647–735. The MIT Press, 2007. ISBN 9780262134743.

[26] Stefano Delle Monache, Pietro Polotti, and Davide Rocchesso. A toolkit for explorations in sonic interaction design. In *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, AM '10, pages 1:1–1:7. ACM, New York, NY, USA, 2010. ISBN 978-1-4503-0046-9. URL http://doi.acm.org/10.1145/1859799.1859800.

[27] D A Norman. *Using Sound for Visibility*. Basic Books, New York, New York, USA, 2002. ISBN 100465067107. 102—-104 pp.

[28] Sile O'Modhrain and Georg Essl. Perceptual Integration of Audio and Touch: A Case Study of PebbleBox. In Karmen Franinović and Stefania Serafin, editors, *Sonic Interaction Design*, chapter 6, pages 203–211. MIT Press, Cambridge Massachusetts, 1 edition, 2013. ISBN 978-0-262-01868-5.

[29] Dan Overholt. A system for sketching in hardware: Do-it-yourself interfaces for sound and music computing. *Sound and Music Computing Conference: Illusions- ...*, 2012. URL http://vbn.aau.dk/files/71485998/SMCrevisedSubmission2012DanOverholtPhotoReady.pdf.

[30] J Preece, Y Rogers, and H Sharp. Inspections: heuristic evaluation. In *Interaction Design: Beyond Human-Computer Interaction*, chapter 15.2, pages 686–702. Wiley, West Sussex, 2 edition, 2007. ISBN 9780470018668.

[31] J Preece, Y Rogers, and H Sharp. Usability testing. In *Interaction Design: Beyond Human-Computer Interaction*, chapter 15.2, pages 646–666. Wiley, West Sussex, 2 edition, 2007. ISBN 9780470018668.

[32] J Preece, Y Rogers, and H Sharp. What is Interaction Design? In *Interaction Design: Beyond Human-Computer Interaction*, chapter 1, pages 1–37. Wiley, West Sussex, 2 edition, 2007. ISBN 9780470018668.

[33] Davide Rocchesso, Stefania Serafin, Frauke Behrendt, Nicola Bernardini, Roberto Bresin, Gerhard Eckel, Karmen Franinovic, Thomas Hermann, Sandra Pauletto, Patrick Susini, et al. Sonic interaction design: sound, information and experience. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pages 3969–3972. ACM, 2008.

[34] H. Sharp, Y. Rogers, and J. Preece. *Interaction design: beyond human-computer interaction*. Wiley, 2007. ISBN 9780470018668. LCCN 2006030649.

[35] Gillian Crampton Smith. What is Interaction Design? In *Designing Interactions*, chapter Foreword, pages vii–xxi. MIT Press, 2007. ISBN 0-262-13474-8.

[36] Boris Smus. *Web Audio API*. O'Reilly Media, first edition, 2013.

[37] Patrick Susini, Nicolas Misdariis, Guillaume Lemaitre, and Olivier Houix. Naturalness influences the perceived usability and pleasantness of an interfaces sonic feedback. *Journal on Multimodal User Interfaces*, 5(3-4): 175–186, January 2012. ISSN 1783-7677. URL http://link.springer.com/10.1007/s12193-011-0086-0.

[38] Patrick Susini, Nicolas Misdariis, Guillaume Lemaitre, Olivier Houix, Davide Rocchesso, Pietro Polotti, Karmen Franinovic, Yon Visell, Klaus Obermayer, Hendrik Purwins, and Kamil Adiloglu. Closing the loop of sound evaluation and design, 2006.

Appendix A

Prototyping Tools

In this section, a brief overview of popular software and hardware suitable prototyping is presented. The list is not limited to audio specific tools, as the physical interaction is also an integral part in the sound design process.

A.1 HTML5

HTML5 has been the new buzzword for a while when the conversation is around new web technologies. It is actually an upcoming standard for the old Hypertext Markup Language, launched in the early 90s "but, what more or less anyone, whether they?re a designer, developer, journalist, pundit, or analyst, actually means when they refer to HTML5 is, to put it simply "sexy new web stuff" (or in the more family-friendly version "New Exciting Web Technologies") [2]." Main benefits of HTML5 include:

- Simpler and less expensive than HTML4 + JavaScript.
- Mostly backwards compatible.
- Device-independent. Most HTML5 features work in browsers on desktop devices as well as mobile devices.
- Browsers often accept "loose" markup, which makes it much more forgiving than more traditional programming languages, where a single missing semicolon can be fatal.
- Richer set of UI elements for sophisticated and user friendly applications.
- Particular benefits on mobile platforms (easy access to native system-dependent keyboards or num pads on e.g. the iPhone).

- Easily distributed. Just open the URL in a browser.
- Applications can be stored on the device for offline functionality.

A.2 CSS

Cascading Style Sheets (CSS) is used to specify the look and formatting of a document written in a markup language. A main benefit from using CSS is that the designers and developers can separate the content from the presentation, with the content being e.g. the pictures, text etc, and the presentation being the formatting in form of fonts, colours and layout. In the most recent version, CSS3, a series of new elements were introduced, with CSS transitions and CSS animations as the most prominent [2]. Furthermore, increased support for shadows, web-fonts, gradients and rounded corners makes it easier to include elements that have traditionally been designed in software such as Adobe Photoshop; this approach often results in a design heavily based on image elements, which have several drawbacks, with the most significant being poor load times (image files takes significantly longer time to transfer than text documents), and problems with scaling to different browser window sizes (e.g. mobile vs desktop browsers).

Transformations are simple animations following e.g. the change of background colour or width. Applying these are as easy as changing the background colour in the CSS document [2].

Animations enable keyframe animations which have traditionally been created in specialised animation software, such as Adobe Flash or by loading video content [2]. The relevance of this, is again the ability to distribute it across platforms, without hurting the content loading times.

Canvas allows to use JavaScript to draw objects, paths, images and text in the browser. Due to the heavy integration with JavaScript, it is increasingly used for browser-based games and other interactive content [2]. The canvas element can even change based on user input, such as tapping, clicking or typing. This allows interactive, dynamic graphical content across almost all modern devices with a web browser without the need for plugins.

Audio and video , with the former being the most relevant aspect in this project, is natively supported (read no plug-ins needed), making including a video or audio file as easy as just adding a simple HTML tag $< videosrc =$

”...” $>< / >$ or $< audiosrc =$ ”...” $>< / >$. Fallbacks on older technologies, such as Flash, is possible for backwards compatibility.

Device APIs makes it possible to access device-specific features [2], such as:

- GPS location
- Compass
- Accelerometers
- Gyroscopes
- Camera (not on all platforms yet)

Earlier, the need for access to these features often caused the requirement of a ”native” app, written in e.g. the more traditional programming language Objective-C on the iPhone or Java on the Android platform, causing the need for several developer teams, which increased cost and development resources needed. Platforms become less of a concern with these features, making for easy distribution to various devices, hopefully speeding up the development.

Offline applications are furthermore closing the gap to native applications. The ability to store applications offline on the devices, so they can be opened in browsers without internet access can both be an advantage in terms of accessibility as well as increasing the performance of the applications, once they have been loaded and cached for the first time. Offline applications furthermore makes it possible to create mobile applications, that are actually webpages, without the user even noticing that it is not a native application.

A.3 Web Audio API

Web Audio API is not really a part of HTML5, but based on JavaScript, and is gaining support in modern browser on both personal computers and mobile browsers. It is to be seen as an addition to the ¡audio¿ element i HTML5, as the HTML5 implementation of a tag only offers basic playback, streaming and control (play, pause, stop etc). The Web Audio API is a work in progress and is envisioned to support a large range of reasonably complex games and interactive applications, including musical ones, and ideally it should be able to support any use case which could reasonably be implemented with an optimised C++ engine controlled via JavaScript and run in a browser [1].

Web Audio API is build around the concept of an AudioContext, which is build up of AudioNodes. Each node can have *inputs* and *outputs*, except *source nodes* and *destination nodes*. The former have no inputs and only a single output, whereas the latter have one input and no outputs, as it represents the final destination to the audio hardware [1]. The API includes most nodes, which might be relevant in an audio prototyping and sketching context, such as filters (biquad), compressors, oscillators, spatialisation/panning, buffers, gain, envelopes and many more, potentially providing a strong tool for audio designers to implement audio on the web, and on web-based devices in this world of increasingly connected products and applications.

Broadly speaking the nodes can be of the following types [36]:

1. *Source nodes:* Sound sources such as audio buffers, live audio inputs, ¡audio¿ tags, oscillators, and JS processors.

2. *Modification nodes:* Filters, convolvers, panners, JS processors, etc.

3. *Analysis nodes:* Analysers and JS processors.

4. *Destination nodes:* Audio outputs and offline processing buffers.

The Web Audio API is built around the concept of a audio graphs, similar to the ones used and visualised in visual programming software such as PureData or MaxMSP as nodes being connected by wires. This concept is generally inspired by the days of the analog synthesiser, such as the Moog Synthesiser[1] where wires were used to connect the different modules, passing the sound from a source, through several nodes (or modules) and finally out through an audio output which could be connected to speakers.

A.4 Node.js

Node.js, *Nodejs* or simply *Node*, is a JavaScript based server side scripting language. What makes this language interesting is, that it runs code written in the same language as web designers and developers use for coding webpages. It runs on Google's V8 JavaScript engine, which is in part responsible for the speed of execution, the lack of which has often been described as a significant drawback of interpreted language, such as JavaScript. Node is based on a single thread (opposed to other server side languages often running multiple threads) which makes it much more simple to write and maintain .

[1]http://en.wikipedia.org/wiki/Moog_synthesizer

A.5 Smartphones

Due to the increasing computation capabilities, smartphones have been proposed as powerful portable prototyping platforms. *iStuff Mobile* is an example, where the smartphone is equipped with external sensors to prototype interaction design situated on the smartphone [3]. The software runs as a background service on the smartphone, allowing the user to interact normally with the smartphone. Meanwhile a piece of software, based on Apple's node-based visual programming language, Quartz Composer [2], runs on a desktop computer receiving the sensor input and mapping it to interaction feedback on the smartphone while the user is operating it. This allows situated prototyping of interaction and sensor input.

[2] http://en.wikipedia.org/wiki/Quartz_Composer

Appendix B

Iteration 1: Basic UI

B.1 Index Cards

This section includes larger pictures of the static graphical layout in the first iteration. These images were used as index cards in the first usability test.

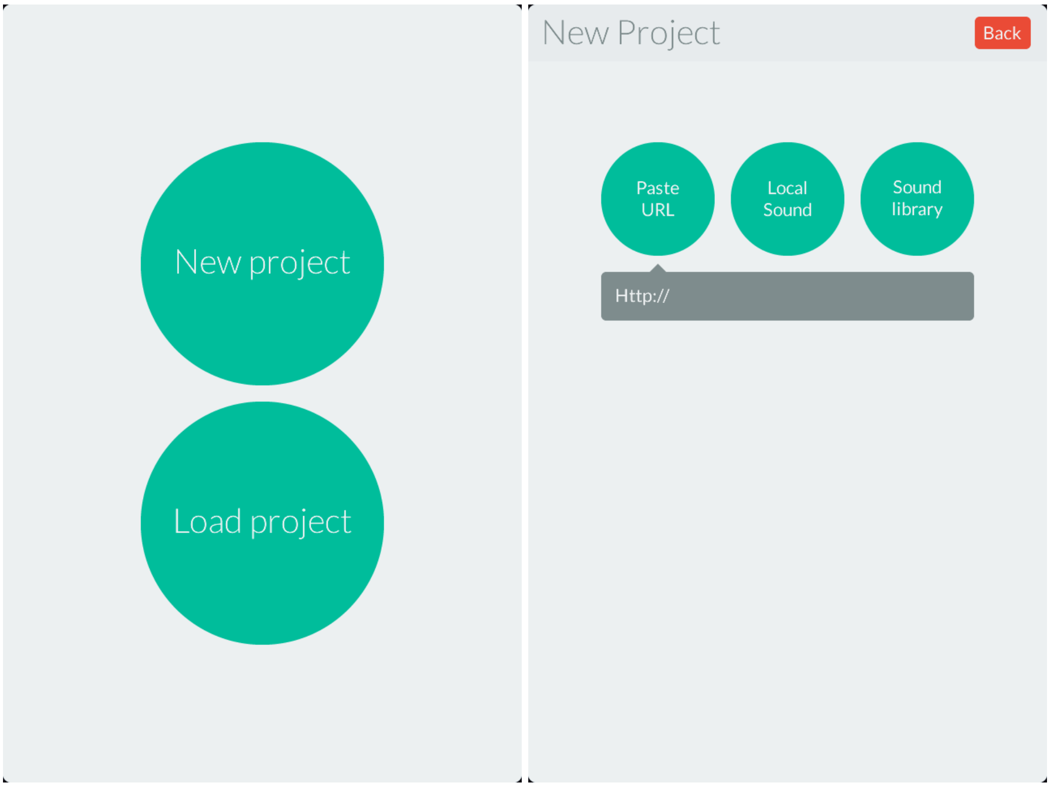

Figure B.1: Left: The first screen presented when opening the web application. Right: The picture shown, when the paste URL button has been clicked. This screen will appear after the "new project" has been pressed.

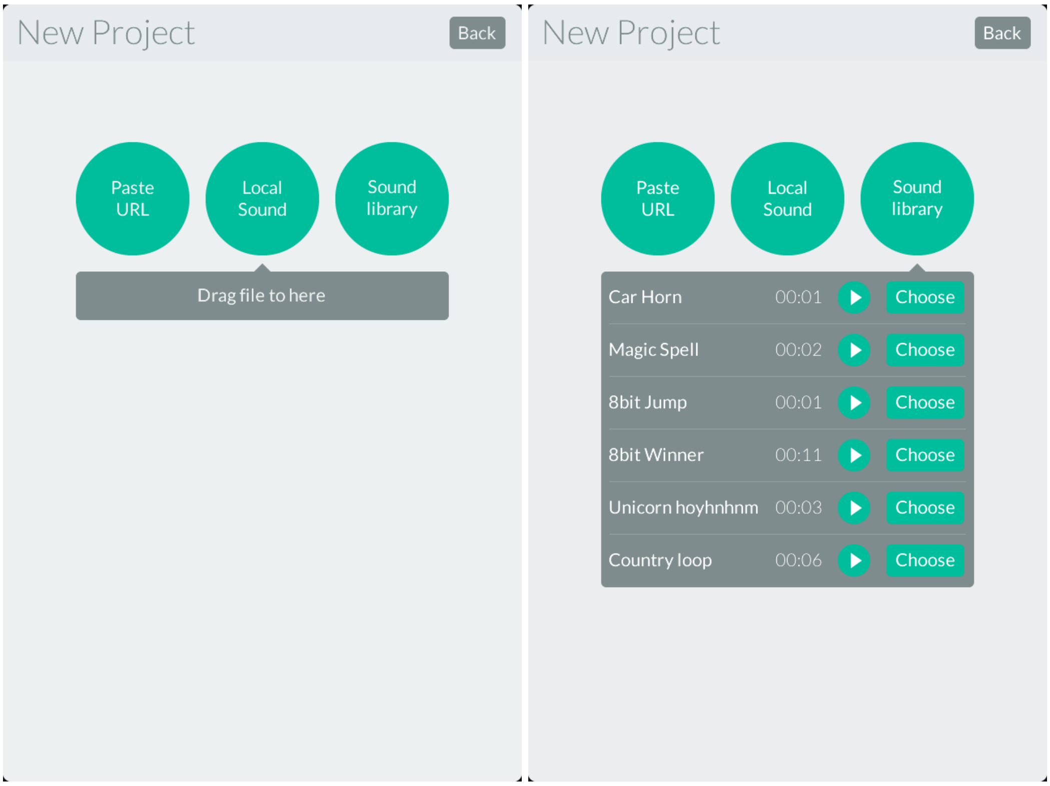

Figure B.2: Left: *Local sound* invites the user to load a sound from the device's memory. It can be dragged (in a desktop browser) or clicking the text will open a load file dialog. Right: *Sound library* allows the user to choose a sound from a predefined library on the server.

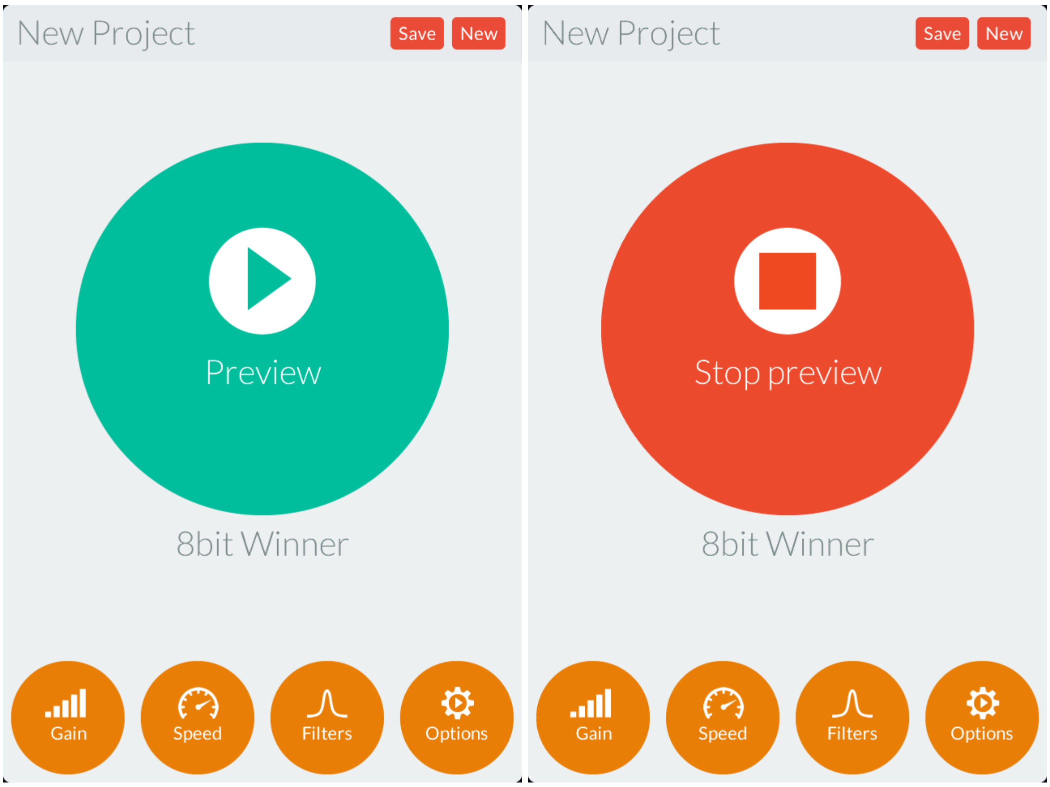

Figure B.3: Left: The screen presented to the user after selecting a sound in the preceding step. Clicking the big green preview button plays the sound as it is. Right: The screen after pressing the preview button. Press the red button to stop preview.

B.2 Evaluation

The subjects were presented with the preceding index cards on the screen of an iPhone 4. Notes and observations from iteration 1 test below:

Subject 1 Paste URL should not be the first option. URL to what? It was not clear that it was an URL to an online sound source. Wording could perhaps be modified to include URL to "sound".

Subject 2 What is speed? Is it the time between playbacks?

Subject 3 Local sound could also mean library. Maybe "add sound" should be placed as option when browsing the library. It would make sense to have all sounds placed at one place (the library) as the first thing shown when opening the app. The users could browse sounds and then only add if they think a sound is missing.

The subject chose the correct sound and understood the function of the small inline preview button. Chooses "fixed" and slides the gain to the middle position. "Fixed" was interpreted as a master value (composited of other variables), where "distance" was just one modulation of it. Gain is a sum of all the parameters.

What is save-project? What does a project include? Several sound files or just one? A project is "big". Suggestion: Sound for a toy-car. Scroll-wheel. It encourages continuos interaction sounds.

Subject 4 Fast. After pushing the green preview button in the button, and the image was changed to the big red stop-bottom. Where did the play-button go? The subject thought pushing the button in the button would bring back the big green play button.

Subject 5 "Save option" confusing. Do i need to save every step? Other than that simple interaction. Very logical flow. Do you need the new project all the time? Ability to save workspace should be there.

Appendix C

Iteration 2: Functional Prototype

C.1 Questionnaire

The subjects were presented with the questionnaire included on the following pages. The first part (until "Test time!") was filled out before testing the interface, and the latter part after interacting with the interface. Recorded answers are included on the cd in the folder `/T2/midterm_data.xls`, as well as interviewer notes in the folder `/T2/t2_interviewerNotes.pdf`

Sketch that!

Below are a series of questions regarding your experience with sound design. Take your time to answer, and don't hesitate to ask if in doubt.

1. **Sex**
   Mark only one oval.
   - Male
   - Female

2. **Age**

   ....

3. **Occupation**

   ....

4. **Product Design**
   Below are a series of questions addressing your experience and relationship with design.
   Mark only one oval per row.

	strongly disagree	disagree	agree	strongly agree
I consider sound early in the design process				
I focus on the technical part of design (how it is implemented)				
Product sound design is important				
I am experienced designing sound				
Sound design is the last thing i think about				
I focus on the aesthetic part of design				
I focus on the functionality				
I am experienced creating prototypes				
Product sound is annoying				

5. Elaborate on your experience with prototyping and design here

6. **Where does sound design belong in the design phase?**
First? Last? Nowhere? Why?

7. **Do you know any sound prototyping or design tools?**
Which do you use and why?

8. **Do you have any musical experience?**
Mark only one oval.

- yes *Skip to question 9.*
- no *Skip to "Test time!."*

Musical experience

Skip to question 11.

9. **How many years experience do you have with music?**
Mark only one oval.

- 0-1
- 1-5
- 6-10
- more than 10

10. **Describe your experience with music**
Do you play an instrument? Produce music? Sing in the shower?

Test time!

It's time to try the prototoype. Do not continue until after you have tried the prototype.

TASK:
Design a sound for a door that opens.

After the design task

Now that you have tried the interace, I would like to ask you to comment on the experience.

Skip to question 11.

11. **I think the graphical interface was**
Mark only one oval per row.

	strongly disagree	disagree	agree	strongly agree
Hard to use	⬭	⬭	⬭	⬭
Responsive	⬭	⬭	⬭	⬭
Useful	⬭	⬭	⬭	⬭
Fun	⬭	⬭	⬭	⬭
Annoying	⬭	⬭	⬭	⬭
Easy to navigate	⬭	⬭	⬭	⬭
Confusing	⬭	⬭	⬭	⬭

12. **I think the physical interface was**
Mark only one oval per row.

	strongly disagree	disagree	agree	strongly agree
Hard to use	⬭	⬭	⬭	⬭
Responsive	⬭	⬭	⬭	⬭
Useful	⬭	⬭	⬭	⬭
Fun	⬭	⬭	⬭	⬭
Annoying	⬭	⬭	⬭	⬭

13. **Describe the scenario you designed the sound for**
Why did you choose that scenario?

14. **Did you miss any control options or sounds?**

15. **General comments**
Anything you might have in mind.

Appendix D

Iteration 3

D.1 Test

This section provides a more detailed description of the test in iteration 3. The procedure is described in detail and the questionnaire and notes and observations are included.

D.1.1 Test Guide

1. *Welcome participant.*

   ,,In this test you will be presented with a toolset for designing interactive product sounds. The tool is intended for designers interested in exploring the possibility of using sound in their products, and does not require any experience with sound design. I will introduce your tools, and then you will have the chance to try out the tools on a specific design task."

2. *Introduce task* ,,Design a sound for a door entry alert. The door alert should be placed in a hotel lobby open 24/7. The door chime should change according to time of day, so it's perhaps less intrusive and conveys a different mood at night." *Encourage situated sketching, by providing the following instructions:* ,,Imagine this is the hotel lobby. There's the desk, where the clerk will be sitting, the lobby area and the door to the street over there."

3. *Introduce subject to GUI and TUI*

   ,,Here are your tools. You'll find a graphical interface on the smartphone, where you can select an existing sound or record a sound. You'll find a series of menu buttons in the bottom of the screen, which brings up a menu, where you can apply filters and choose how to control these.

This could be manually by sliding on the phone screen or by using the sketch-box."

4. *Encourage experience prototyping*

   "You should see the interface as a tool to explore and try the sound and interaction for yourself. Use the sketch box and phone as props together with the interaction with the door."

5. *Questions*

   Are you satisfied with the designed sound?

   Answer questionnaire.

D.2 Questionnaire

The subjects were presented with the questionnaire included on the following pages after performing the design task. Recorded answers are included on the cd in the folder `/T3/test3.sav`, as well as interviewer notes and observations in the folder `/T3/t3_interviewerNotes.pdf`

Sketch that!

Below are a series of questions regarding your experience with sound design. Take your time to answer, and don't hesitate to ask if in doubt.

1. **Sex**
   Mark only one oval.
   - Male
   - Female

2. **Age**

   ....

3. **Occupation**

   ....

4. **Product Design**
   Below are a series of questions addressing your experience and relationship with design.
   Mark only one oval per row.

	strongly disagree	disagree	neither	agree	strongly agree
I focus on the aesthetic part of design					
I am experienced designing sound					
I consider sound early in the design process					
I focus on the technical part of design (how it is implemented)					
I focus on the functionality					
I am experienced creating prototypes					

5. **Where does sound design belong in the design phase?**
   First? Last? Nowhere? Why?

   ....

   ....

   ....

   ....

   ....

6. **What's your favourite sound design tool?**
   Why?

7. **I think the graphical interface was**
   Mark only one oval per row.

	strongly disagree	disagree	agree	strongly agree
Hard to use				
Responsive				
Useful				
Fun				
Annoying				
Confusing				

8. **I think the physical interface was**
   Mark only one oval per row.

	strongly disagree	disagree	neutral	agree	strongly agree
Hard to use					
Responsive					
Useful					
Fun					
Annoying					
Confusing					

9. **Whats missing?**
   Were you missing any tools or options to design the sound? Or do you have suggestions for functionality that should be included?

10. **General comments**
    Anything you might have in mind.

11. *Mark only one oval per row.*

	Strongly disagree	disagree	neither	agree	strongly agree
I was satisfied with the resulting sound	⬭	⬭	⬭	⬭	⬭
The interface encouraged situated prototyping	⬭	⬭	⬭	⬭	⬭
The sketch-box is portable	⬭	⬭	⬭	⬭	⬭
The phone-based interface is portable	⬭	⬭	⬭	⬭	⬭

D.3 Laser Cutting Plans

The laser cutting plans for *Sketch Box v2* are shown in Figure D.1, p. 99. A full-size version can be found on the cd.

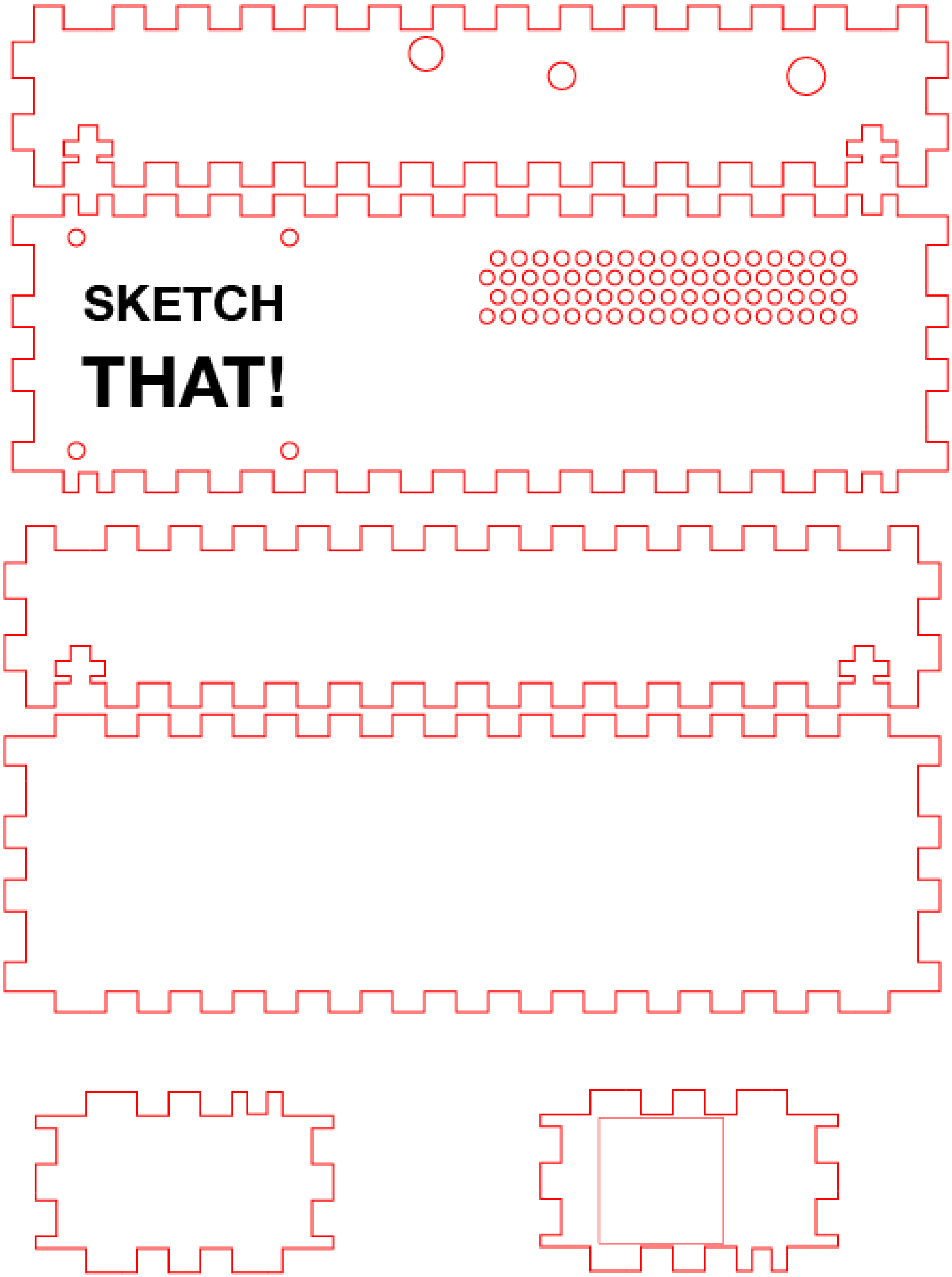

Figure D.1: Sketch Box v2 laser cutting plans.