The 6th Finger: Practical Challenges in the Design of a Multitouch Audio Appliance

Wolfgang Beer
Software Competence Center Hagenberg
Softwarepark 21
Hagenberg, Austria
wolfgang.beer@scch.at

Christian Salomon
Software Competence Center Hagenberg
Softwarepark 21
Hagenberg, Austria
christian.salomon@scch.at

Mario Winterer
Software Competence Center Hagenberg
Softwarepark 21
Hagenberg, Austria
mario.winterer@scch.at

Karl Putzhammer
Acousta Engineering
Moosstrasse 60
Salzburg, Austria
putzhammer@acousta.at

Bernhard Schauer
Acousta Engineering
Moosstrasse 60
Salzburg, Austria
schauer@acousta.at

Thomas Rechberger
Acousta Engineering
Moosstrasse 60
Salzburg, Austria
rechberger@acousta.at

CHI'15 Extended Abstracts, April 18–23, 2015, Seoul, Republic of Korea.

http://dx.doi.org/10.1145/2702613.2702961

Abstract

Effective multitouch interaction does not reduce itself to a simple tracking of five fingers, there are several emerging effects that could prevent intuitive interaction in industrial appliances. This use-case describes practical challenges that were documented during the design and implementation of a holistic user interaction design in the domain of high-end audio equipments. The engineering process had to combine tangible user interface controls with state-of-the-art multitouch software fader panels in an intuitive way. This work also gives some background information about complex distributed audio routing equipment and user interaction along with technology and usability issues that appear during the design of a multitouch appliance. Several experiments were implemented in order to gain empiric data to substantiate our practical findings.

Author Keywords

Multitouch gestures; product development; audio editing; audio routing;

ACM Classification Keywords

H.5.2 [User Interfaces]: User-centered design.

Figure 1: Motor Fader Panel.

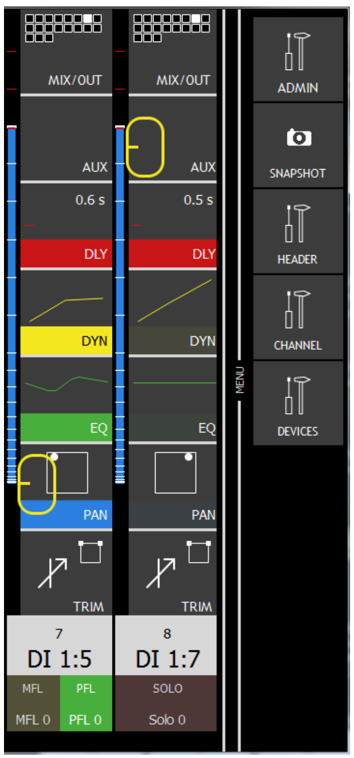

Figure 2: Multitouch Fader Panel.

Introduction

All over the world, high-end audio routing and editing suites are used for managing and mixing a large number of live audio streams in various application scenarios. A typical scenario is the live mixing and broadcasting of radio stations or the live audio direction in large theaters or opera houses. During a broadcast session various uncompressed input and output audio streams, such as voice interviews from different microphones, live transmissions from other stations and music recordings from digital archives are mixed together and routed onto multiple live output channels.

The high number of available channels (in our equipment this means to manage up to 2,048 input and output channels with 192Khz uncompressed, real time audio streams) in combination with highly time critical constraints during operation poses some pressure on the interaction design.

Background

The traditional user interface for audio routing and editing equipment is based on tangible user interfaces, such as push buttons, potentiometers and most importantly on motor fader panels, as it is shown in Figure 1. These motor faders automatically adjust settings and give a fine grain tangible feedback on actual signal levels. The tangible motor faders are directly bound to the multitouch fader panels to show the same signal level adjustment. So if a user changes the signal level by using a multitouch fader the motor fader panel autonomously adjusts its tangible value to the same level. Even today, professional audio experts measure the quality of audio editing equipment by the reactivity and sensibility of its tangible adjustment controls. While there are many scientific studies on improving the user interface of DJs [6] or VJs [2] with modern multitouch environments, we identified a lack of experience in the area of professional audio routing equipment. Based on the early work by Sears and Shneiderman [8], who proved that direct-touch user input is faster than mouse input for target selection, we tried to integrate direct multitouch interaction into traditional audio editing suites. Our hypothesis was that within the scenario of controlling multiple audio channels the efficiency of user input could be additionally increased, compared to Sears and Shneiderman's founding. The multitouch fader panel, as it is shown in Figure 2, is divided into frames of eight single audio channels. By swiping left or right the audio expert can choose out of a variable number of frames and by swiping up and down the user jumps between input and output audio channels. Each audio channel offers a variable number of tools to change the level and frequencies of its audio transmission, such as an Equalizer tool, which is shown in Figure 3. The multitouch fader user interface is directly connected to a tangible motor fader panel where each of the eight channels is represented by a motor fader. So within our research project the primary goal was to combine existing tangible elements with several, distributed multitouch fader panels, as they are shown in Figure 2. First and most critical requirement was to provide extremely low reaction and response times between the multitouch graphical user interface and the mixing hardware.

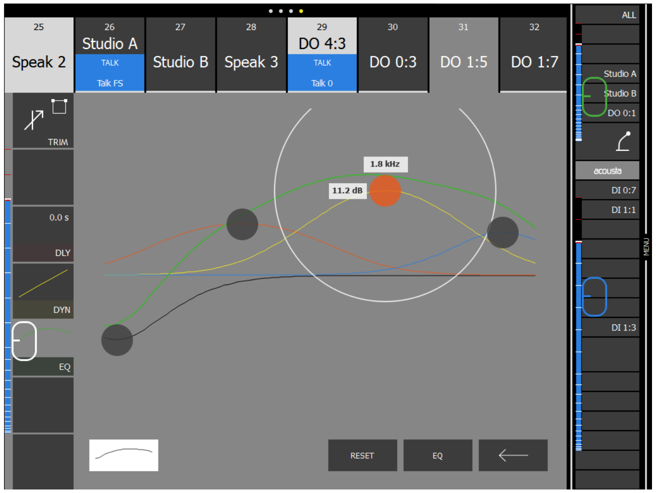

Figure 3: Multitouch Equalizer Manipulation.

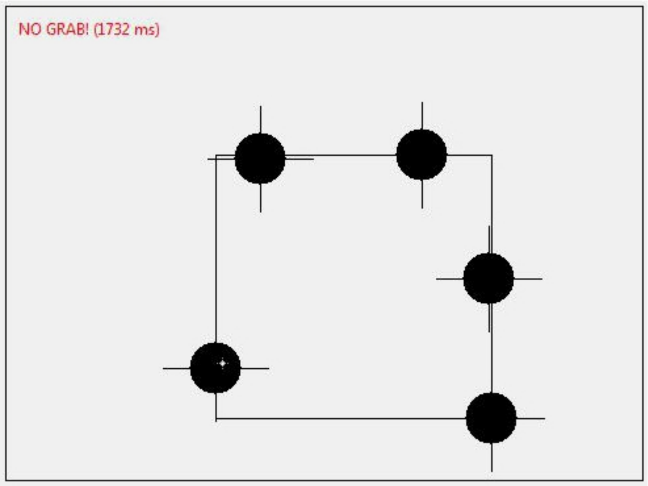

Figure 4: Multitouch Gesture Capture Tests.

A second requirement was to flexibly combine the new multitouch input paradigm with a traditional tangible user interface in a holistic way, as it is shown in Figure 3. A specific problem here was to provide intuitive gestures that enables users to quickly react on live events avoiding wrong usage. Therefore, clear and consistent multitouch gestures are a critical objective for our work. The next section details the challenges that were identified in the course of this project.

Challenges in Multitouch Appliances

Defining an interaction design for a complex, distributed system such as our audio editing appliance is a time consuming and error prone process. Several iterative steps lead to a first user interface design that was continuously evaluated by domain experts.

Individual Gesture Variation

During the definition of specific multitouch gestures within our interaction design it was observed that even a quite simple gesture, such as a three finger down slide, was not reliably detected. After some user tests – a screenshot of our test capture screen is shown in Figure 4 – we soon found out that the variation of the captured gesture was very high between different people. We discovered some very distinct interpretations of one and the same gesture with different people performing it. Our observation goes perfectly along with the study Blaica et al.[3] performed in 2013 that showed that people can even be identified with an accuracy of 94.69% just by analyzing their individual multitouch point signature.

Multitouch and Direct Manipulation Alternatives

Another challenge was to support multitouch gestures along with traditional mouse based direct interaction. The

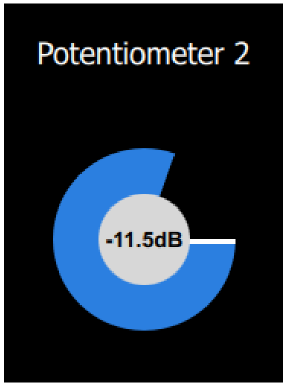

Figure 5: Potentiometer Widget.

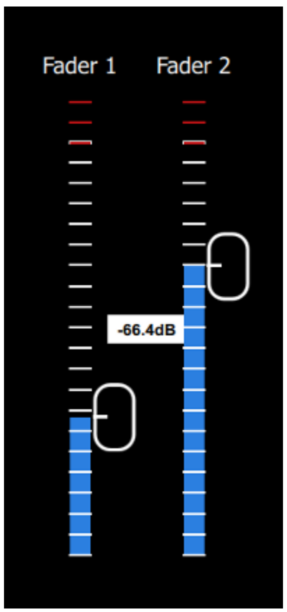

Figure 6: Fader Widget

problem identified here is twofold as modern user interface frameworks still struggle to support both paradigms and the underlying operating system adds some unpredictable behavior. The other more strategic challenge is to define a holistic multitouch interaction design that minimizes the interference with mouse based direct interaction. In case of our touch fader panel this was quite problematic as the screen is stuffed with audio channel widgets. So if a user intends to perform a multitouch gesture but clicks a widget instead, it automatically triggers an unintended action. In case of accidentally muting the live channel of a radio transmission this causes serious troubles.

Our experience here was that in case of a holistic interaction design the engineering process has to cover all available screen interaction and cannot be defined in an incremental process. This much differs from previous interaction designs where each widget is self responsible for handling the user input individually. In multitouch interaction designs also the context and surrounding of a widget and its gestures has to be taken into account.

Multitouch Reaction Time
In combination with our strong requirement to build a highly responsive user interface, the individual reaction times of different multitouch displays played an important role. During our tests with various multitouch panels we discovered a range from 6ms up to 100ms reaction time for tracking 5 touch points. Beside the touchscreen reaction time, also the additional operating system and gesture detection overhead has to be taken into account. On the other hand, the visual feedback of the actual audio level as well as the visual reaction after the user set an action (such as muting a channel) depends on the update frequency of the selected display hardware. So even if we know from previous experiments, that audio experts are able to detect action to audio asynchronism above 25ms [1], the selection of hardware and software already defines a specific worst case scenario.

Widget Framework
Another challenge during the design of our multitouch appliance was the lack of a standardized widget framework that supports the seamless integration of multitouch events and gesture recognition. Despite the fact that there exist many different research prototypes, there is no equivalent framework available that offers product quality. Examples for research frameworks are Proton++ [5] a framework for declarative multitouch UIs, PyMT [4] a Python based multitouch user interface toolkit for rapid prototyping of interaction techniques or Midas [7] a logic framework for the definition and detection of multitouch gestures.

Experiments and Results

Due to the identified drawbacks of existing frameworks we implemented a custom, flexible, and reliable widget and gesture framework, that was tested concerning usability and ergonomics. For this purpose we developed a specific use-case driven test application based on our multitouch framework. This usability test application is divided into 11 steps, each of them testing a specific aspect of our framework supported by an experimental user interface. A step consists of a task description, a canvas containing multiple user interface elements such as faders shown in Figure 6, potentiometers shown in Figure 5, and buttons. Each user has to finish all the given tasks before proceeding to the next step.

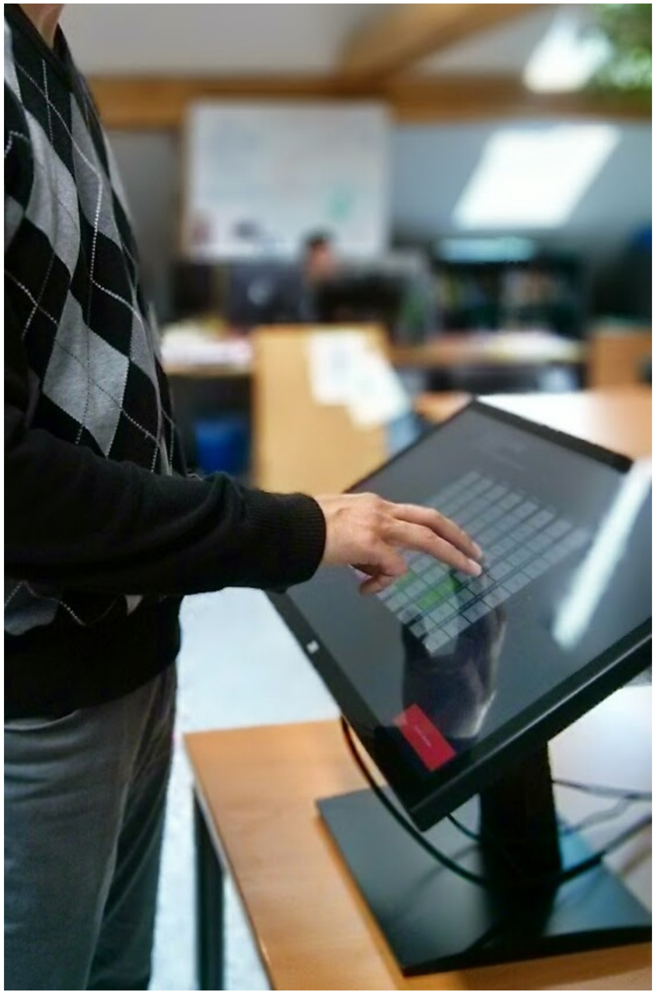

Figure 7: Usability Test Setting.

Some examples within our usability test steps are: changing the state of a button, setting the value within a given range of a fader or potentiometer (the range is in all cases 4 dB, all widgets are linearly scaled between [-100dB, +10dB]), or performing a gesture. The current value of a widget is displayed by an indicator as shown in the Figures 6 and 5. Prior to a test step the functional principle of widgets and gestures is described along with sample widgets, which is used to get familiar to the mode of operation. The potentiometer can be positioned in two different ways: Either the desired value is selected directly by touching a point on the ambient circle (direct mode), or gradually by pressing any point on the circle and moving the digit around the center of the widget (touch and hold mode). The fader does not allow direct select. It can only be manipulated by dragging the fader handle (see Figure 6) (touch and hold mode).

The usability test was conducted with 27 participants that had no previous knowledge nor any professional experience with comparable fader and potentiometer widgets. The test application was displayed on a 24" multitouch monitor (native resolution of 1920x1080 pixels) that supports individual tracking of 20 touch points with a touch point latency of 6ms and a video response time of 16ms. As shown in Figure 7, the adjustable monitor was inclined vertically about 60 degrees to simulate the final product as close as possible. In addition to the observation of the participants every user interaction like touches and gestures were logged to a database. Based on the collected interaction data and after analyzing our observations we encountered following noticeable facts:

1. Although it does not have a scale, a third of all participants used the direct mode to manipulate the potentiometer. They gradually refined the value by repeating this trial and error strategy. Those people required nearly twice the time (8.4 sec in average)

than the people that used the touch and hold mode (4.9 sec in average).

2. When using the touch and hold mode to change the value of a potentiometer for more than 40 dB, about 60 percent of interaction time were spent in fine tuning the potentiometer within a 3 dB range. This value drops to around 25 percent of the time for fader value changes. This indicates, that labeling and displaying of the current value must be improved for potentiometers.

3. After introducing the test users to the operation modes of a potentiometer, 8 of them tried to apply the direct interaction mode to the fader as well, although they have already used drag and hold properly in tests before.

4. Moving the handle of a fader to a mid-value was in average faster (4.1 sec) than changing the value of a potentiometer by touch and hold (4.9 sec). Setting the value of a fader to its minimum or maximum only took 1.3 seconds in average.

5. We experienced some usability problems when testing the manipulation of multiple faders at the same time, especially when the test users had to manipulate 4 faders at once. Having the faders arranged with a horizontal distance of 22 mm, 74 percent of all participants tried to perform this task with one hand only. 80 percent of them lost the fader they controlled with the ring finger during movement. Also the mean processing time of 10.2 seconds indicates that this task was rather hard to perform and that an additional (hardware) device with tangible elements is essential.

6. One test scenario tested an experimental grab gesture, which is defined as a grab-like movement of all five digits on the surface. Our implementation approach considers this gesture as successfully performed by verifying that the bounding box of five touch points contracts to a certain degree. In addition, the final bounding box (on touch release) must lie completely within the initial bounds of the touch points (see Figure 4). More than 15 percent of all grab gestures were not recognized using this approach. Investigations revealed that users tend to use the thumb as steady anchorage point and do not move it at all. Instead they just move the fingers towards the thumb. As a result, the tip of the thumb performs a rolling movement causing the corresponding touch point to move slightly outside its initial bounding box.

We experienced a similar usability problem as described above when testing an early version of the framework implementing a two-finger vertical flick gesture. The recognition of the flick gesture did not work for most of the tries of a single person, whereas it worked reliable for all other testers. After additional tests with extended interaction logging we found out that this person unintentionally touches the screen with an additional digit, the thumb.

Discussion

In this case-study a new user interaction design for an audio routing and editing appliance was discussed. The interaction design is based on a multitouch graphical user interface combined with tangible elements, such as motor faders and push buttons. We presented several challenges that are related to the low maturity of actual multitouch interaction software frameworks. We are confident that the introduction of multitouch interaction within audio routing and editing equipment will increase the efficiency of these workplaces significantly. The work also discusses several issues, such as reaction time and mouse point and click integration that come along with the design of a multitouch appliance. An overview of a usability study with 27 participants has been presented, along with several observations that were identified by a detailed analysis of the resulted datasets. According to the fact that our usability study was performed with 27 participants that had no previous knowledge of audio equipment, our next step is to perform a similar usability test with domain experts.

References

[1] Adelstein, B. D., Begault, D. R., Anderson, M. R., and Wenzel, E. M. Sensitivity to haptic-audio asynchrony. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, ACM (New York, NY, USA, 2003), 73–76.

[2] Banerjee, A., Burstyn, J., Girouard, A., and Vertegaal, R. Waveform: Remote video blending for vjs using in-air multitouch gestures. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, ACM (New York, NY, USA, 2011), 1807–1812.

[3] Blaica, B., Vladušlč, D., and Mladenić, D. Mti: A method for user identification for multitouch displays. *Int. J. Hum.-Comput. Stud. 71*, 6 (June 2013), 691–702.

[4] Hansen, T. E., Hourcade, J. P., Virbel, M., Patali, S., and Serra, T. Pymt: A post-wimp multi-touch user interface toolkit. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, ACM (New York, NY, USA, 2009), 17–24.

[5] Kin, K., Hartmann, B., DeRose, T., and Agrawala, M. Proton++: A customizable declarative multitouch framework. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, ACM (New York, NY, USA, 2012), 477–486.

[6] Lopes, P. A., Ferreira, A., and Pereira, J. A. M. Multitouch interactive djing surface. In *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*, ACE '10, ACM (New York, NY, USA, 2010), 28–31.

[7] Scholliers, C., Hoste, L., Signer, B., and De Meuter, W. Midas: A declarative multi-touch interaction framework. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '11, ACM (New York, NY, USA, 2011), 49–56.

[8] Sears, A., and Shneiderman, B. High precision touchscreens: Design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud. 34*, 4 (Apr. 1991), 593–613.