RWTH AACHEN UNIVERSITY

REXband

A Multi-User Interactive Exhibit To Explore Medieval Music

Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Marius Wolf

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: Feb 03rd, 2006
Submission date: Aug 03rd, 2006

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, August 3rd, 2006

Contents

Abstract xiii

Überblick xv

Acknowledgements xvii

1 Introduction 1

1.1 Structure 3

2 Goals and Concept 5

2.1 Features 5

2.2 Usage Scenario 8

3 Related work 11

3.1 Jam-O-Drum 11

3.2 Band-out-of-the-Box (BoB) 13

3.3 coJIVE 14

3.4 WorldBeat 15

3.5 MIDI Controllers 15

3.5.1 Commercial Systems 15

3.5.2 Custom Solutions 17

3.5.3 Discussion 19

4 Theory 21

4.1 Overview 21

4.2 Patterns in Interactive Exhibit Design 22

4.2.1 Patterns and Pattern Languages . . . 22

4.2.2 Pattern Languages in Other Domains 23

4.2.3 Borchers's Pattern Language for Interactive Exhibits 24

4.3 Learning and Design Theories 26

4.3.1 Behaviorism and Feedback 26

4.3.2 Constructivism, Situated Learning and Context 30

4.4 Medieval Music 31

4.4.1 Instruments 31

Hurdy Gurdy 31

5 First Prototype: Exploring the Idea 35

5.0.2 Features 36

5.0.3 Helper tools 37

5.1 Implementation in Max/MSP 37

5.1.1 Melodic Correction 38

5.1.2 Drone Strings 39

5.1.3 Basic Drum Track 40

5.1.4 Feedback 40

5.2 Analysis: User Test 42

5.2.1 Procedure 43

5.2.2 Results 43

5.2.3 Design Implications 44

6 Second Prototype: Preview System 47

6.1 Excursus: Horizontal vs. Vertical Prototypes 47

6.2 Design 48

6.3 Implementation 50

6.3.1 The Exhibit in Idle-Mode 51

6.3.2 Playing the Exhibit 52

6.3.3 Modeling a Hurdy Gurdy 53

6.3.4 Giving Feedback 53

6.3.5 Modifying the Hurdy Gurdy 55

6.4 Analysis 58

6.4.1 User Test 58

6.4.2 Expert Review 59

6.5 Last Minute Modifications and On-site setup 60

6.5.1 Feedback from Visitors 61

7 Experimental Feature: Rhythmic Correction **63**

7.1 Algorithm Design 63

7.2 Literature Review 64

7.3 Experiment Concept 65

7.4 First Experiment: Determining Threshold 66

7.4.1 Setup and Tasks 66

7.4.2 Results 68

7.4.3 Discussion 68

7.5 Second Experiment: Testing Usefulness 70

7.5.1 Setup and Tasks 70

7.5.2 Results 70

7.5.3 Discussion 71

8 Third Prototype: Final System **73**

8.1 Design 73

8.2 Implementation 76

8.2.1 Software 76

MIDI Signal Chain 76

State Machine 78

State Controller 79

Main Controller 80

8.2.2 Hardware 80

Harp 80
Frame Drum 81
8.3 Evaluation 83
8.3.1 Controlled User Test 84
Procedure 84
Results 86
8.3.2 Public Test 87
Observations 87
8.3.3 Cognitive Walkthrough 89
8.3.4 Discussion 90

9 From Prototype to Exhibit **93**
9.1 Physical Setup 93

10 Summary and Future Work **97**
10.1 Summary and contributions 97
10.2 Future work 98

Bibliography **101**

Index **105**

List of Figures

1.1 Regensburg 1

2.1 Melodic Correction 7

3.1 Jam-O-Drum 12

3.2 Band-out-of-the-Box (BoB) 13

3.3 coJIVE 14

4.1 Borchers's Pattern Language 25

4.2 Behavioristic Learning 27

4.3 Hurdy Gurdy Mechanics 32

5.1 First Prototype: Overview 36

5.2 First Prototype: Melodic Correction 38

5.3 First Prototype: Drone Strings 39

5.4 First Prototype: Drone Note 40

5.5 First Prototype: Scoring 41

5.6 First Prototype: Generate Feedback 42

6.1 Horizontal vs. vertical prototype 48

6.2 Second Prototype: Rough Structure 51

6.3 Second Prototype Screenshot 52

6.4 BeatTapper 54

6.5 Hurdy Gurdy Front View 56

6.6 Doepfer MIDI board 57

6.7 Hurdy Gurdy Microswitches 57

6.8 Hurdy Gurdy Light Barrier 58

7.1 Rhythmic Correction 64

7.2 Experiment Setup 67

7.3 Experiment Results - Medieval Rhythm . . . 69

7.4 Experiment Results - Straight Rhythm 69

8.1 Modified frame drum 74

8.2 Modified harp 75

8.3 Final System Architecture 77

8.4 Final System: MIDI Signal Chain 77

8.5 Final System State Machine 79

8.6 Vibe Sensor 81

8.7 Harp Velocity Curve 82

8.8 Drum Trigger 82

8.9 Final System: Overview 84

8.10 Main goals and questions 85

8.11 Couven Museum setup 88

9.1 Exhibit: Setup 94

9.2 Exhibit: Hurdy gurdy 95

9.3 Exhibit: Frame drum 95

9.4 Exhibit: Harp 96

Abstract

REXband is a computer music system that allows users to experience medieval music in a new way. Our goals were to provide both an enjoyable experience as well as teach users some basics about medieval music and its instruments while interacting with the system. Designed as an exhibit for the Regensburg Experience (REX), an exhibition that opens in Regensburg in fall 2006, it features electronically modified replicas of medieval music instruments (hurdy gurdy, harp, frame drum) as input devices.

REXband supports users in playing while preserving a high level of musical expressiveness. However, it does not assume any knowledge about medieval music. The computer system behind REXband also plays an accompaniment and gives feedback about the user's playing performance.

We use a combination of standard computer technology, electronic sensors, MIDI and digital audio to create the system. A selection of related work shows other approaches to supporting people in playing music, creating new ways of musical expression or coupling traditional interaction metaphors for instruments with modern technology.

REXband was designed in an iterative process of three iterations. Each iteration begins with designing and implementing a prototype that can then be tested with users. The results from the user tests are taken as input for the next iteration and influence design decisions.

Apart from empirical results, REXband was influenced by learning and design theories, design patterns for interactive exhibits and medieval music history.

Überblick

REXband ist ein computergestütztes Musiksystem, dass es seinen Benutzern gestattet, die Musik des Mittelalters auf neue Weise zu erleben. Unser Ziel ist es, sowohl eine unterhaltsame Erfahrung zu bieten als auch durch die Interaktion mit dem System einige Grundlagen über mittelalterliche Musik zu vermitteln. Dazu stehen Nachbauten von mittelalterlichen Instrumenten (Drehleier, Harfe, Rahmentrommel) zur Verfügung, die mittels Sensoren zu elektronischen Eingabegeräten umgebaut wurden.

REXband ist als Museumsexponat konzipiert und soll ab Herbst 2006 der Öffentlichkeit im Rahmen der Regensburg Experience (REX) zugänglich gemacht werden.

Da bei Museumsbesuchern kein Vorwissen über mittelalterliche Musik angenommen werden kann unterstützt REXband den Benutzer, ohne dabei jedoch die musikalischen Ausdrucksmöglichkeiten übermässig einzuschränken. Das Computersystem, auf dem REXband basiert, spielt außerdem eine musikalische Begleitung zu und gibt Rückmeldung über den Spielerfolg des Benutzers.

Die eingesetzte Technologie ist eine Kombination aus herkömmlicher Computertechnologie, Sensorelektronik, MIDI und digitaler Klangerzeugung. Eine Auswahl verwandter Arbeiten zeigt andere Ansätze, Menschen beim Spielen von Musik zu unterstützen, neue Ausdrucksmöglichkeiten zu schaffen und verschiedene Arten der Interaktion aus der Welt traditioneller Instrumente mit moderner Technik zu verbinden.

REXband wurde in einem iterativen Verfahren in drei Stufen entwickelt. Grundlage jeder Stufe sind Entwurf und Entwicklung eines Prototyps, der dann mit Benutzern getestet wird. Die Ergebnisse dieser Tests sind dann die Basis für die nächsten Entscheidungen im Entwicklungsprozess.

Neben empirischen Resultaten basierte das Systemdesign auf Theorien zu computergestütztem Lernen, Entwurfsmustern für die Gestaltung interaktiver Exponate sowie mittelalterlicher Musikgeschichte.

Acknowledgements

A lot of people helped in the development process. My gratitude goes to:
Julien Biere and Brigitte Weidmann for managing the whole REX project and supporting my work from the organizational side; Professor David Hiley from the University of Regensburg for his knowledge of medieval music; the musicians of the medieval music ensemble "...sed vivam!" who recorded the musical accompaniment; the "Schule für Tontechnik" in Regensburg for providing a professional recording environment to record the accompaniment; Alois Biberger who built all the beautiful instrument replica; Peter "Gecko" Götz who built the stands for the instruments; Thomas Boettcher for providing insights on hurdy gurdy mechanics; Yvonne Jansen, Stefan Werner and Laszlo Bardos for help with modifying the instruments; Christoph Wilhelm for some very nice photos; and all the users that participated in test sessions and gave valuable feedback.

I would like to personally thank:
My parents, Günther Wolf and Marlis Esser, for constant support during my studies; Edith Wolf and Gerhard Hachmann, for help and advice on all those little things in life; my girlfriend, Birgit Schiffer, for emotional support and being more nervous about this work than I was ;-) ; my supervisor Eric Lee whose office door was always open for me; Professor Jan Borchers, who taught me a lot more than just computer science; and the whole Media Computing Group for providing a very unique, productive and enjoyable working environment that I will probably never forget.

Chapter 1

Introduction

"It takes a great deal of history
to produce a little literature."

— Henry James

Figure 1.1: Bridge over Danube river into Regensburg's city centre.

Music has been an important part of human culture for millennia. What we regard as modern music today is based on a long history of musical tradition. While music culture, instruments and styles have continuously changed over time, it has always been both an influence to human culture as well as a medium of expression. Looking at the music of a certain time or culture can tell us a lot about the people who played it or listened to it.

With computer technology influencing almost every aspect of human culture nowadays, new interactions with music have become possible. By combining these possibilities with the beauty and richness of medieval music, we wanted to develop a system that brings this almost forgotten culture back to life.

The initial idea for this system came up during a brainstorming session in which we collected exhibit ideas for a new museum/visitor centre in Regensburg[1] , a city in Bavaria, Germany. Having a rich history that goes back to medieval times and beyond, the aim of the project dubbed "Regensburg Experience" (or REX[2]) is to present the city and its development over the centuries to both its citizens and the many tourists that are continuously coming to visit the city. REXband is one of several interactive exhibits that are designed and implemented by the Media Computing Group. While other exhibits focus on the city's architecture ("Time Window") or local medieval poetry ("Minnesang"), REXband features medieval music.

It allows the visitor to play music on authentic replicas of medieval instruments, while they hear a medieval dance piece as accompaniment. Using audio samples and decoration, the exhibit creates the atmosphere of a medieval tavern. The instruments behave realistically, but the user is supported by several electronic improvisation aids to allow even people with only little musical experience a successful and enjoyable interaction.

To ensure historic authenticity, the exhibit was created in cooperation with the "Forum Mittelalter"[3] , a group of historians, musicologists, linguists and other scientists concerned with medieval studies that is based in the University of Regensburg[4] . A lot of other people helped create this ex-

[1] http://www.regensburg.de
[2] http://www.rex-regensburg.de
[3] http://www.forum-mittelalter.org/
[4] http://www.uni-regensburg.de/

hibit; see the acknowledgements section for a full list.

1.1 Structure

This thesis is organized as follows:

- In the *Concept*-chapter, we describe the basic ideas behind REXband and present its major features.
- In *Related Work*, we talk about other computer music and improvisation systems.
- The *Theory* behind REXband is covered here: design patterns for interactive exhibits, learning theories and a bit about medieval music and instruments.
- The *First Prototype* was an early version of REXband developed in Max/MSP that was used to explore design ideas and run some first user tests.
- The *Second Prototype* was the first fully working prototype of REXband that was shown during a two-week exhibition preview in Regensburg in summer 2005.
- In *Experimental Feature: Rhythmic Correction*, we explore a design idea to support novice users in playing music and describe the results of a bigger user study.
- The *Third Prototype* is the basis of the final system that is to be installed in Regensburg in fall 2006.
- *From Prototype to Exhibit* sketches how the final exhibit could look like.
- Finally, in *Summary and Future Work* we sum up our results and discuss research ideas that could be pursued in the future.

Chapter 2

Goals and Concept

"Ideas are like rabbits. You get a couple and learn how to handle them, and pretty soon you have a dozen."

—John Steinbeck

REXband was created to explore a new approach of presenting medieval music and culture. Music is an important part of medieval history and the aim of this work is to let visitors experience medieval music for themselves. The visitors should not only be able to listen to and read about medieval music, but also explore for themselves while they can both learn about medieval music and instruments, and enjoy the experience. We focused on the secular music of that time, because we felt that there was a bias towards courtly and churchly culture in public perception, which does not represent the wide variety of medieval culture.

2.1 Features

Authentic atmosphere

We aimed the exhibit at giving the visitor an impression of being inside a medieval tavern. The system creates a constantly audible atmospheric background noise, generated from recorded material. Authentic replicas of medieval instruments (hurdy gurdy, harp and frame drum) are arranged in the room for the visitors to play. All the instruments are invisibly modified to act as electronic controllers for a computer system that is working in the background. The instruments all behave realistically, although they produce only very little sound themselves.

Accompaniment and improvisation aids

As soon as a visitor starts playing one of the instruments, a pre-recorded accompaniment track is played. The track plays for about one minute and consists of a tambourine, a harp, a fiddle and some other instruments played by members of the medieval music ensemble "...sed vivam!". REXband does not require any knowledge about medieval music or music theory and supports users by making it impossible to play wrong notes that do not fit to the accompaniment.

Melodic correction

Our approach to this type of melodic correction is fairly simple, yet effective: As we know the key of the accompaniment piece, we can pick a subset of all twelve possible notes that fits to the accompaniment, map these notes onto a linear scale and implement that in the linear arrangement of keys/strings in the harp and the hurdy gurdy (see Figure 2.1).

Feedback

The system rewards the users with audible feedback for their performance. While playing, the system can play sounds of falling coins, encouraging shouts, stomping feet, etc.. At the end of the accompaniment track, the users are rewarded with an applause. The intensity of the feedback depends on how well the users have played.

Invisible hardware

While making use of computer technology and electronics, none of this should be noticeable for the users. It is very important for us to create the feeling that the users are interacting with musical equipment, not with a computer. Even though most visitors would probably be able to guess that there is modern technology behind REXband and we do not make a secret out of that, we would like to stress the musical aspects of this exhibit, not the technology behind it.

Pure audio system

REXband was consciously designed as a system without computer-generated visual feedback. Although some of

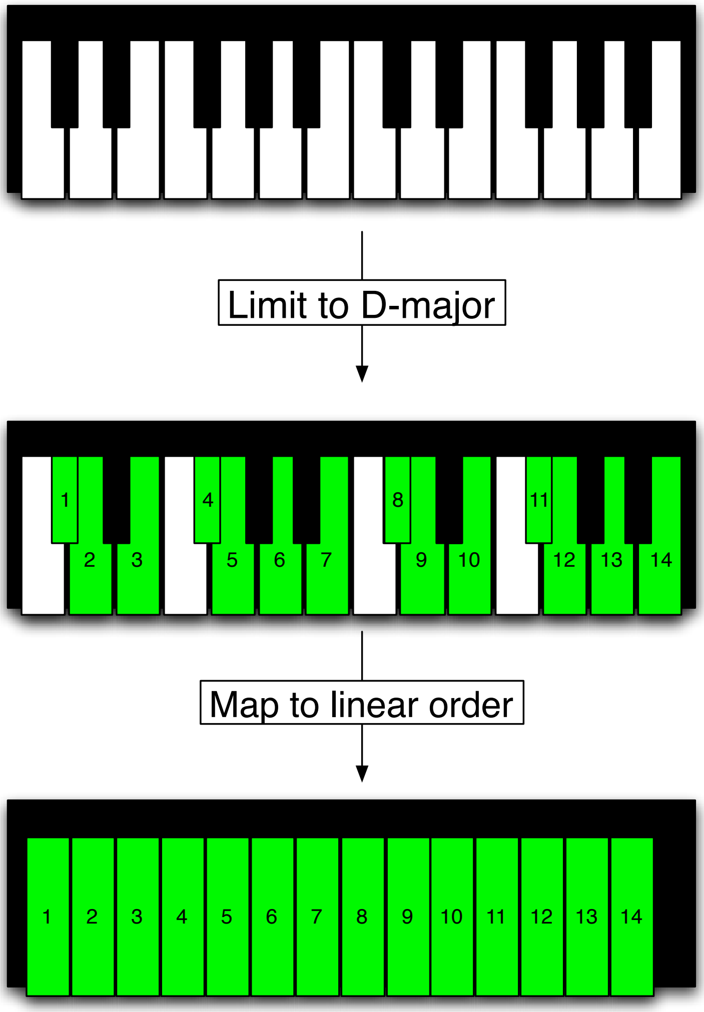

Figure 2.1: Melodic correction is offered as an improvisation aid by REXband. The D-major scale is used as an example in this figure.

the earlier ideas for REXband included visual feedback on video screens, such as a pre-recorded or computer-rendered medieval audience, we soon abandoned these ideas in favor of a system that relies entirely on audio feedback. We wanted to focus the users' attention on the instruments and the atmosphere that the exhibit created, and we felt that videos would add little to the quality of the exhibit, and would moreover be very resource-intensive to produce.

Iterative Design

REXband was developed using iterative design, a design method that emphasizes prototyping and user testing as vital elements during development. The idea is to develop the system from prototype to prototype, with each prototype getting closer to the final system. This approach is promoted by many experts in the human-computer interaction (HCI) community.

2.2 Usage Scenario

We would like to illustrate how REXband could be used with a scenario:

Yvonne, Sarah and David, a group of young people in their twenties, visit Regensburg during a trip through Germany's south. After crossing the Danube river, they notice signs advertising the "Regensburg Experience" and decide to go inside.

When they enter the exhibition, they hear the sounds of instruments being tuned from a semi-secluded corner of the room. As they reach the corner, they see three music instruments on wooden stands, facing each other. They recognize one as a harp, and another as some kind of drum. The third one is unfamiliar to them: a wooden box with a crank on one side and a line of buttons on another.

They look around more and notice that the surroundings are decorated to resemble a medieval tavern: pictures show people dancing, drinking or talking. They can also hear people talk and drink, and a voice says:"How about some music?"

Encouraged by this, David walks up to the drum and starts hitting it with one hand. To his surprise, music suddenly starts around him. Sarah and Yvonne laugh at David's surprised look and decide to join in.

Sarah tries out the harp. She is unfamiliar with the instrument, and starts plucking single strings carefully. The sound of the harp pleases her, and soon she is experimenting with playing simple melodies.

Yvonne approaches the third instrument and carefully starts turning the crank. She hears an unfamiliar humming sound, but she somehow finds it fitting to the music. She starts pressing the buttons and finds out she can play melodies quite easily.

As they become more familiar with their instruments, they hear the sounds of coins falling and encouraging shouts over the music. After about a minute, the music suddenly ends and they hear a loud applause. David, Sarah and Yvonne step back from their instruments and jokingly bow to each other. Then they leave the exhibit, making room for another group of visitors who have watched them play and already seem eager to try it out themselves.

Chapter 3

Related work

"New knowledge is the most valuable commodity on earth. The more truth we have to work with, the richer we become."

—Kurt Vonnegut

Music has been a successful application domain for computer technology for several decades now. A wide variety of hard- and software systems has been developed, and several well-established standards such as the MIDI protocol[1] (Musical Instrument Digital Interface) facilitate the creation of music systems.

In this chapter, we will present several systems that allow new ways of interacting with music, support users in playing traditional instruments, improvise or connect acoustic music instruments with computer systems.

3.1 Jam-O-Drum

The Jam-O-Drum system, developed by Blaine and Perkis [2000], is an interactive music exhibit focussing on collaborative rhythmic improvisation and visualization of musical

[1]http://www.midi.org/

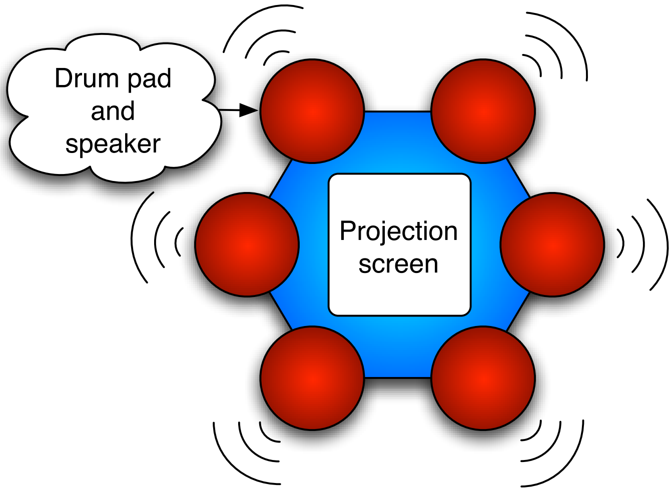

Figure 3.1: Top view on the Jam-O-Drum hardware setup.

Hardware setup

cues. Its hardware consists of a hexagonal table with electronic drum pads and a speaker at each of the six rounded corners. Using a projector, images and animations could be shown on the table (see Figure 3.1). Blaine's team experimented with several custom-made software applications to create an interesting and enjoyable group experience using the Jam-O-Drum hardware.

Example applications

One of the most successful applications was "Call and Response" in which the system would play a rhythm pattern and show a visual pointer to one user, encouraging him to repeat the pattern while showing visual response cues. The authors see musical education as a possible application area for this.

Among the other software systems the team experimented with were "Blisspaint", where users could trigger abstract, colorful animations using the drumpads, and "HexaPong", a game inspired by the classic Pong video-game[2] .

Contrary to REXband, this system focusses on rhythmic, percussive improvisation rather than melodic improvisation. Furthermore, this system has no explicit connections

[2]http://www.pong-story.com/

to historic music instruments; historic authenticity is no relevant design goal in this context.

3.2 Band-out-of-the-Box (BoB)

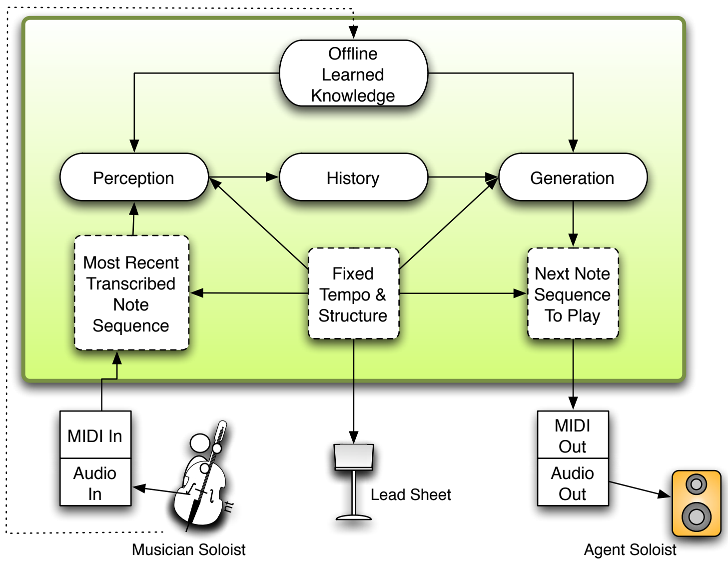

Figure 3.2: Component overview of the BoB-System.

Improvisational music companion

BoB, developed by Thom [2000], is an interactive improvisational music companion that plays together with a real musician. It listens to what the musician plays and responds in an appropriate manner. To ensure this, the system needs to be trained for the musician before a playing session. BoB also plays a fixed accompaniment and knows when to improvise itself and when to let the musician improvise. The different solo parts are known to BoB and the musician as they use the same lead sheet. Figure 3.2 shows an overview of the system components as described in the paper.

This system is targeted towards experienced musicians and does not provide any support for the player's instrument. Using jazz music as musical background for BoB, it also differs greatly from the focus on medieval music in REXband.

3.3 coJIVE

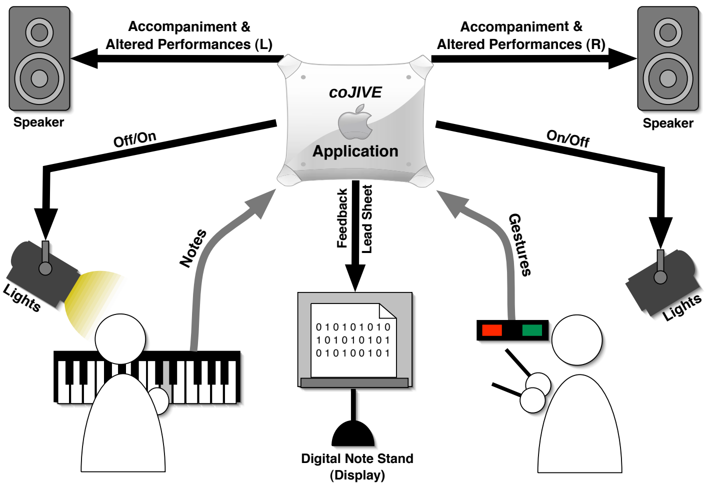

Figure 3.3: The coJIVE System and its components. (Source: Buchholz [2005]; reprinted with permission)

Collaborative Jazz Improvisation

coJIVE is a collaborative multi-user jazz improvisation system, developed by Buchholz [2005] and Klein [2005]. Using a standard MIDI keyboard and a pair of Buchla Lightning II[3] batons, it allows users with varying levels of musical experience to improvise to a fixed recording (see Figure 3.3). Using artificial intelligence, the system supports novice users as well as more experienced players in creating aesthetically pleasant music by dynamically selecting appropriate notes based on music theory. coJIVE also also shows a lead sheet on the screen and gives visual cues to the players suggesting when to play a solo and when to play in the background.

REXband and coJIVE share the idea to support musical novices in playing, but the musical backgrounds (jazz and medieval music) are again very different. This also leads to clearly distinct design decisions concerning instruments, melodic correction and system setup.

[3] http://www.buchla.com/

3.4 WorldBeat

Selection of music applications

WorldBeat, developed by Borchers [1997] and presented as an exhibit in the Ars Electronica Center[4] in Linz, Austria, showed how a wide variety of musical applications could be controlled with a relatively simple input device. Using only a pair of Buchla Lightning II batons, users could conduct a synthesized piece of music, improvise on an "invisible xylophone", play a musical memory game or use the baton as a selection device in a menu.

WorldBeat shows a much wider approach to letting musical novices experience different aspects of music. It offers several very different interaction styles and does not aim at modeling certain instruments realistically. Contrary to REXband, sound generation is based mostly on synthesized sounds rather than a combination of recorded audio and wavetable synthesis.

3.5 MIDI Controllers

While piano-style keyboards are still the standard input devices for playing synthesized music, a lot of other controllers have been developed to implement different interaction metaphors for synthesizers.

3.5.1 Commercial Systems

There is a wide variety of systems by major companies that allow controlling synthesizers using a guitar, drums or other popular music instruments.

Electronic drum systems

The Roland[5] RT-10 drum trigger modules allows a standard acoustic drum kit to send MIDI signals when a drum is hit. The trigger is mounted at the metal rim of the drum and connected to a trigger MIDI converter, such as

[4]http://center.aec.at/

[5]http://www.roland.com

the Roland TMC 6. Other systems, such as the Yamaha[6] DTXpress or the Roland V-Drums, rely on custom electronic drumpads. While some electronic drumpads resemble smaller and flatter versions of real drums, other models consist of plain, flat rubber cylinders that contain the necessary electronics and also send a trigger signal when hit.

Guitar MIDI interfaces

Yamaha and Roland are also offering systems to control a synthesizer using a normal electric guitar. Both the G50 (Yamaha) and the GI20 (Roland) require special pickups that need to be mounted below the strings of the guitar. These pickups are connected to guitar MIDI interfaces which then produce MIDI signals.

Doepfer MIDI controller

To modify key instruments, such as older electronic organs, that do not send MIDI signals by themselves, Doepfer[7] offers the CTM64 ("Contact To MIDI Interface"). This MIDI controller board has connectors for switches and potentiometers. It converts these input signals to various MIDI messages, such as note or pitch wheel messages. While this system was designed for keyboard-style input devices, it can also be used for other purposes. In REXband, this controller was used to modify the hurdy gurdy (see 6.3.5—"Modifying the Hurdy Gurdy").

i-CubeX toolkit

i-CubeX by Infusion Systems[8] is a toolkit to connect different types of sensors to a computer. The sensors are connected to a controller and transmit data to the computer via MIDI. The controller can be programmed to send data in various MIDI message formats and to do simple preprocessing (threshold filtering, average, ...). Infusion Systems offers both standalone software as well as Max/MSP[9] -patches to program the controller and access sensor data . Available sensors include vibration, bend and heat sensors.

Infrared Baton System

The Buchla Lightning II system consists of two infrared-emitting batons that are tracked by an infrared receiver. This data can then be used to trigger sounds directly or fed into a computer system via MIDI as a two-dimensional position information that can be used for further processing.

[6] http://www.yamaha.com
[7] http://www.doepfer.de
[8] http://www.i-cubex.com
[9] http://www.cycling74.com

Music-related controllers have also been embraced by the video game industry. Examples for this are Nintendo's Donkey Konga[10] or Drum Mania for Sony's Playstation 2[11] . A survey of alternate input devices for video games is presented in Blaine [2005].

Video game controllers

3.5.2 Custom Solutions

Some interesting modifications of acoustic instruments exist that show the possibilities of combining traditional instruments with modern electronics. While some of these instruments are commercially available, they fall into another category than the off-the-shelf technology described above.

A MIDI version of the hurdy gurdy was done by instrument maker Neil Brook[12] . He modified one of his own electro-acoustic hurdy gurdies with magnetic switches and crank control and designed a custom controller to translate the output to MIDI.

MIDI Hurdy Gurdys

German instrument builder Dieter Gotschy also mentions having worked on a MIDI version of a hurdy gurdy in an interview with "Folker!" magazine (see Pollack [1999]). Very little details are given, but the project was apparently never finished due to lack of funding.

UK-based company Accordion Magic[13] offers enhancing any accordion with the capability to send MIDI messages. They use proprietary MIDI hardware and install electronic switches together with the mechanical switches of the original instrument.

MIDI accordions

Schiesser and Traube [2006] present an "electronically-augmented saxophone". It consists of a toolkit of sensors that can be mounted onto an acoustic saxophone and mapped to various parameters in Max/MSP. Their set of

Augmented saxophone

[10]http://www.nintendo.com
[11]http://www.sony.com
[12]http://freespace.virgin.net/hurdy.gurdy/midigurdy.html
[13]http://www.accordionmagic.com/

sensors includes switches, sliders, inclinometers (measuring the angle between instrument axis and floor) and more. The authors' next step is to test various sensor configurations and mappings with performers and composers.

Drums

Maki-Patola et al. [2006] placed a camera underneath the drumhead of an acoustic djembe drum. They can thereby track the player's hand position and adapt a fixed sequence of computer-triggered drum samples accordingly. While the individual hits of the drum patterns are automatic, the user has direct control over loudness, tempo and timbre. User tests showed that many users found it easier to play musically interesting rhythms with the augmented djembe. The authors also observed a novel playing-style that combines the automated patterns with traditional playing techniques.

Light harps

Several people have experimented with light harps or laser harps, a relatively new kind of music instrument with its design based on a classical harp. The metal or nylon strings of the acoustic instrument have been replaced with light beams that trigger notes when when the beam is interrupted. Examples for this concept have been built by Andrew Kilpatrick[14] , Laser Spectacles Inc.[15] , Favilla and Cannon [2006] and others. While often visually appealing, these instruments lack the tangible feedback of an acoustic harp.

MIDI harps

MIDI-enabled harps that do have tangible feedback and can be played very similarly to a traditional harp have been built by David Kortier[16] . He offers both acoustic instruments as well as harp-shaped MIDI controllers (not producing any sound themselves) and combinations of these. His approach is based on sensing the vibration of each string, with his custom hardware being able to handle up to 48 strings as input.

Flute

Various sensors exist to use the air jet in a flute as an input channel for an electronic music system. da Silva et al. [2005] present some possible solutions and propose a pres-

[14]http://www.andrewkilpatrick.org/mind/laserharp/
[15]http://www.laserspectacles.com/
[16]http://www.kortier.com/

sure sensor inside the flute with its output mapped to a filter control.

3.5.3 Discussion

The systems presented in this section show that a lot of innovative and well-designed solutions for connecting various traditional instruments with a computer already exist. However, none of these shares our focus: to create robust, authentic, electronically augmented replicas of medieval instruments. The systems presented in this section focus mostly on artistic expression or providing toolkits for creative instrument design. While these are certainly honorable goals, we wanted to follow a different approach. Nevertheless, our work has been influenced by the systems described above and would have been much harder to build without existing technology.

Chapter 4

Theory

"The best effect of any book is that it excites the reader to self-activity."

—Thomas Carlyle

As REXband has several very different goals, the theory that inspired our work comes from a multitude of fields and disciplines.

4.1 Overview

Interactive exhibit design raises some interesting questions that are not as pressing in other application areas of computer technology. An interactive museum exhibit can be considered as a "walk up and use" system where only very little training and instruction is possible. Other challenges include robustness, easy handover or the selection and design of non-standard input devices. We use *interaction design patterns* to solve some of these design problems.

REXband is meant to not only be fun and enjoyable, but also to give visitors of REX an impression of medieval music and culture. To pursue these goals, we built on both classic and modern *learning and design theories* and ensured authenticity of content with advice and literature recommendations from Professor David Hiley, an expert in *medieval music* working at the University of Regensburg.

4.2 Patterns in Interactive Exhibit Design

Special requirements for exhibits

Designing an interactive museum exhibit is inherently different to creating other types of interactive systems. In a museum setting, almost no background knowledge or training can be expected from the user. The time of visitors using an exhibit is usually very limited, so that no on-site training is possible. For interaction design, this means that only very little time will be available to provide an enjoyable, memorable experience and convey information. Too much visible technology might also confuse visitors and keep them from using the exhibit. And naturally, an exhibit also has to attract visitors' attention first.

Design patterns for interactive exhibits

Borchers [2001] has created a pattern language to address these and other issues. Going back to the earlier pattern idea of the architect Alexander et al. [1977], this represents a collection of tried and tested solutions that is understandable by both designers and end users, thus allowing easier communication of design decisions and goals between different participants in the design process. We will first explain what a pattern language is and how it can be used, present a brief look at what pattern languages exist so far and then show examples on how Borchers's pattern language can be helpful for designing interactive exhibits.

4.2.1 Patterns and Pattern Languages

"Quality Without a Name"

Although Borchers references earlier examples, Alexander can be considered as the modern pioneer of design patterns. Based on the observation that buildings and towns that people enjoy living in have a certain, timeless "Quality Without a Name", he attempts to capture knowledge about these qualities in a pattern format. In contrast to other authors who have used the pattern format (e.g., Gamma et al.

[1995]), his patterns are meant for both experts in the field (i.e., architects) and non-experts who will use or live in the buildings that are to be created. This serves to communicate knowledge about successful ideas between architects as well as other people working with architects, thereby creating a common vocabulary and facilitating communication between the different parties involved in a construction project.

Alexander's pattern format includes:

- the *name* of a pattern (e.g., "STREET CAFE"),
- a *ranking* of the author's confidence in the validity of the pattern (one to three stars),
- a *picture* showing an example application,
- the *context* of the pattern (references to higher-level patterns),
- a *problem statement* (shows briefly which problems the pattern addresses),
- a more detailed *problem description* (including examples),
- the *solution* that the pattern offers,
- a *diagram* that illustrates the solution,
- *references* to lower-level patterns.

From patterns to pattern languages

The elements *context* and *references* show that a pattern usually does not exist on its own. Alexander's patterns form a pattern language in which the patterns are arranged by the level of scale they apply to. This level of scale ranges from entire city districts (e.g., "IDENTIFIABLE NEIGHBOURHOOD") to individual parts of buildings or rooms (e.g., "SITTING WALL").

4.2.2 Pattern Languages in Other Domains

The idea to use a pattern language for collecting and communicating information has been picked up by other peo-

Software design patterns

ple and applied to various domains. Gamma et al. [1995] show a collection of patterns for object oriented software construction. Other than Alexander's patterns, this pattern collection is meant mostly for experts in the field. While the book was quite successful, this different approach has also received some criticism (e.g., Alexander [1996]) as it excludes non-experts from using the patterns as common vocabulary with experts.

Patterns in HCI

"Common Ground"[1] by Tidwell [1999] and more recently Tidwell [2005] show HCI pattern languages that preserve Alexander's original idea of a common vocabulary between experts and non-experts. Although the format of the individual patterns as well as the arrangement of the patterns is different here, the original idea is still clearly visible.

4.2.3 Borchers's Pattern Language for Interactive Exhibits

Borchers [2001] presents his pattern language for designing interactive exhibits in a way that is directly inspired by Alexander's work. The pattern format is very similar, but the arrangement of the patterns is not purely based on a spatial scale, but rather by different aspects of the system ("COOPERATIVE EXPERIENCE", "CLOSED LOOP", ...) and level of abstraction. Figure 4.1 shows an overview on Borcher's patterns and the relations between them.

Though not all patterns are applicable to REXband (e.g., "IMMERSIVE DISPLAY"), some convey valuable design experience. Some examples:

- ATTRACT-ENGAGE-DELIVER explains the general interaction path a user takes through an interactive exhibit. The system should be designed so that it first attracts users, engages them and deliver one of the "messages" which the systems is meant to convey.
- ATTRACTION SPACE introduces an "idle-mode" in which the system tries to attract users. The system

[1] http://www.mit.edu/~jtidwell/common_ground.html

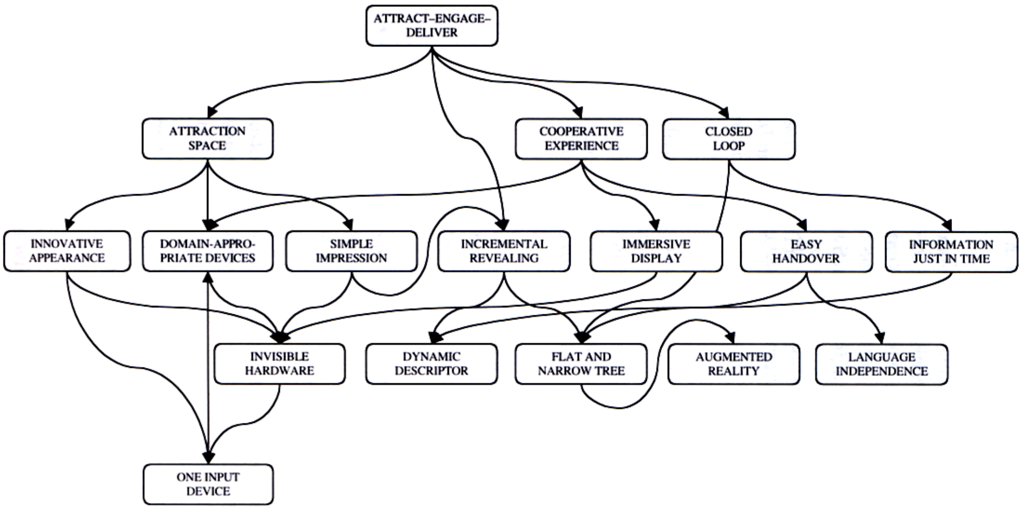

Figure 4.1: Overview of Borchers's Pattern Language.

should also have a defined space in which it attracts visitors and not frequently violate this space.

- COOPERATIVE EXPERIENCE stresses the point that museums are often visited by larger groups and exhibits should therefore not be designed to be used by one person at a time only.

- Because one user often takes over from a previous user in the middle of the interaction in, EASY HANDOVER should be possible. Using the system should only require minimal knowledge about the the previous user's input and make it easy to restart the interaction from the beginning.

- Since most users will not engage in a long interaction in a museum setting, the interaction should be designed as a CLOSED LOOP that always goes back to a starting state.

- While mouse and keyboard are the primary input devices for interacting with computers, DOMAIN-APPROPRIATE DEVICES should be offered in an exhibit.

- A lot of people who are not familiar with electronics and computer technology will be driven away by a system that shows more of that technology than necessary. INVISIBLE HARDWARE lowers this interaction threshold.

4.3 Learning and Design Theories

While REXband is certainly meant to provide an entertaining and enjoyable experience, it also aims at conveying information about medieval music and instruments. The design of these aspects is inspired mostly by two important learning theories, namely behaviorism and situated learning. A good overview on theories is given by Kerres [2001].

Even though behavioristic theory is widely considered as limited and outdated in some aspects, some concepts (such as the stressed importance of feedback) are still helpful for designing interactive systems.

While feedback was an important focus in the design of the system itself, the theories of situated learning and constructivism emphasize the importance of the context. Designing the whole exhibit (which does not only consist of electronics and software) and placing the exhibit in the museum are aspects which are strongly influenced by considering the physical surroundings (context). This assumption is also supported by our user tests.

4.3.1 Behaviorism and Feedback

Behavioristic basics

Behaviorism is a theory developed in the 1960s by psychologist B.F. Skinner. It focusses on on explaining human behavior by looking at the sensory input (“stimulus”) and the output (“response”) and puts only little (if any) emphasis on cognitive processes leading to decisions and actions.

This theory leads to a teaching approach that aims mostly at training the learner to show a certain response to a given stimulus. A schematic diagram of the process is given in Figure 4.2.

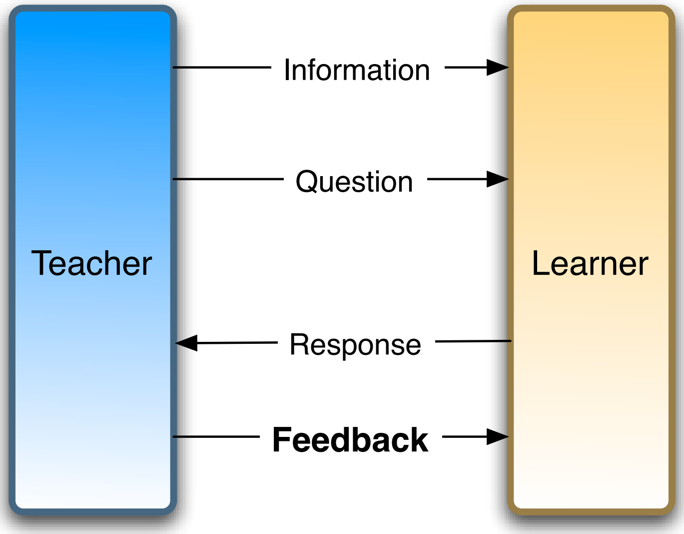

Figure 4.2: Behavioristic view on teaching process. This model emphasizes the importance of feedback.

Applying the theory to practical learning

The teacher (or teaching computer system) first gives some information to the learner, which is the content that is to be learned. He then asks a question (stimulus) to test if the information has been learned. Depending on the answer (response) of the learner, the teacher gives an appropriate feedback.

Behavioristic learning stresses the importance of appropriate feedback. Three different kinds of feedback can be used:

- *Reinforcement:* The learner is rewarded for a correct answer or desired behavior. The idea behind this is to encourage the learner to give this response again to the same stimulus in the future.
- *No feedback:* Instead of giving negative feedback, the teacher can ignore the response or give neutral feedback. Behavioristic theory predicts that the given response is less likely to occur again if there is no feedback from the environment.
- *Punishment:* The learner is criticized or punished for

his behavior to discourage him from giving the same response again in the future. This should only be used in exceptional cases.

Feedback should be given soon after the response to stress the connection between response and feedback. However, feedback is not necessarily required after each response. Using the feedback, the learner should learn to give the appropriate response even without or with less regular feedback.

The application of behaviorism on eLearning-systems is often referred to as “programmed instruction”. Computer systems were seen as suitable for implementing these concepts since a teaching system (in contrast to a human teacher) would never become impatient and give appropriate feedback all the time. The learner would also not have to fear embarrassment in front of a class when making mistakes and could repeat a lesson as many times as desired.

Criticism

Behaviorism was the predominant concept for designing eLearning systems for a long time; teaching methods were refined to better adapt to the learner’s individual learning speed and abilities, and other learning theories have been inspired by behaviorism. These essentially propose similar models, but focus more on the communication of content instead of feedback (“cognitive approaches”).

While some of the core ideas of behaviorism are still widely considered as useful, many theorists and practitioners in learning nowadays regard purely behavioristic teaching approaches as too limited. Many see a behavioristic approach as inappropriate to learn connections between facts and support a deeper understanding of a given subject. Criticism is also backed with empirical studies questioning theoretical assumptions of behaviorism, such as the stressed importance of feedback.

The Media Equation

While we agree with the criticism, we felt that the idea of giving immediate and reinforcing feedback is useful and can help make using our system more interesting and entertaining. This is backed by Reeves and Nass [1996, 2002], who offer some interesting studies and insights on how people treat technical devices like social actors.

In chapter 4 (“Flattery”), the authors give anecdotal and empirical evidence as well as literature references on how praise given to the user by a computer system affects the user's perception of that system.

Experiment on praise from a computer

Their own work on this subject includes an experiment in which a user was to play a variant of the ”Twenty Questions”-game with a computer. The user was told to think of an animal that the computer then tried to guess. The computer could only use questions which can be answered with “yes” or “no”. If the computer was unable to guess that animal, it would ask the user to suggest a better question to distinguish their animal from others. The computer would then give a rating of the user's question to the user and either praise the user for making up a good question or criticize him for suggesting a bad one. The feedback was completely unwarranted in some cases and based on actual evaluation in others. The users were informed about the current setting at the beginning of the experiment. At the end of one session, the user had to fill in a questionnaire that asked for:

- The user's rating of his own performance in the game.
- The user's rating the computer's performance in the game.
- How much the user liked the computer.

Praise is not questioned

The results showed that users gave higher ratings in all three categories if the computer praised them for their questions, regardless of whether or not the feedback was warranted.

Unwarranted criticism is dismissed

Criticism was treated differently by the participants: Undeserved criticism led to the users rating their own and the computer's performance better. However, if criticism was warranted, people gave low ratings on their own performance.

“Substitute sugar for vinegar”

The authors argue that there is an asymmetry between criticism and praise, as there seems to be no perceived difference between unwarranted and sincere praise (in contrast to criticism). Because the subjective quality of a system can be improved using praise, they suggest to use praise more in interactive systems to improve the subjective quality and value of the system ("substitute sugar for vinegar").

4.3.2 Constructivism, Situated Learning and Context

Affordances

Constructivist theory has become increasingly popular in learning in the last 20 years and represents a major paradigm shift compared to behaviorism and its successors. This shift was based, among others, on the observations of Gibson [1979] who found certain "affordances" in people's environment that can be understood without cognitive processing. Norman [2002] also uses the term "affordance" which "refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how how the thing could possibly be used".

Situated learning

This leads to an approach called "situated learning" that focusses more on the context of actions and learning processes. According to this theory, knowledge is not stored in people's minds, but created in every situation based on the current context and the person's previous experience. Norman refers to this as "knowledge in the head and in the world". He stresses the fact that while people can perform appropriately in a given situation, they will not necessarily be aware about every detail of their actions or be able to explain it when asked.

Applied to learning, these ideas have inspired some interesting models, such as "anchored learning" (finding an anchor in the content and using it to help with focus and motivation) or "cognitive apprenticeship" (the learner plays the role of a craft novice that learns from an expert by example and interaction).

4.4 Medieval Music

Separation of secular and churchly music

Medieval music was separated into secular and churchly music. This division was especially obvious in musical education, where a lot of teachers refused to use the works of "pagans", e.g. classic roman literature. van Waesberghe [1969] calls this the "dualism of churchly and secular music".

There was also a strict separation of instruments used by clerics or by secular musicians. While some monks started experimenting with secular instruments such as the harp around 950 AD, especially the church officials tried to dissociate from the Roman and Greek legacy. We will focus on secular music and its instruments in our work.

4.4.1 Instruments

The instruments that were used for REXband are the hurdy gurdy, harp and frame drum, popular instruments in secular medieval music. The general principle of medieval harps is very similar to modern harps, so we will not go into detail here. A frame drum is a flat drum with a diameter of 20-50 cm that is played while holding it in the one hand, hitting it with the other.

However, the hurdy gurdy is a lesser known instrument today and we do not assume general familiarity with it. We will therefore give some details on its mechanics and functions.

Our description is based mostly on talks with musicians, instrument builders and historians as well as live demonstrations of instrument replicas.

Hurdy Gurdy

Hurdy gurdy in medieval times

The hurdy gurdy is a music instrument whose first appearances in Europe can be traced back to the tenth century

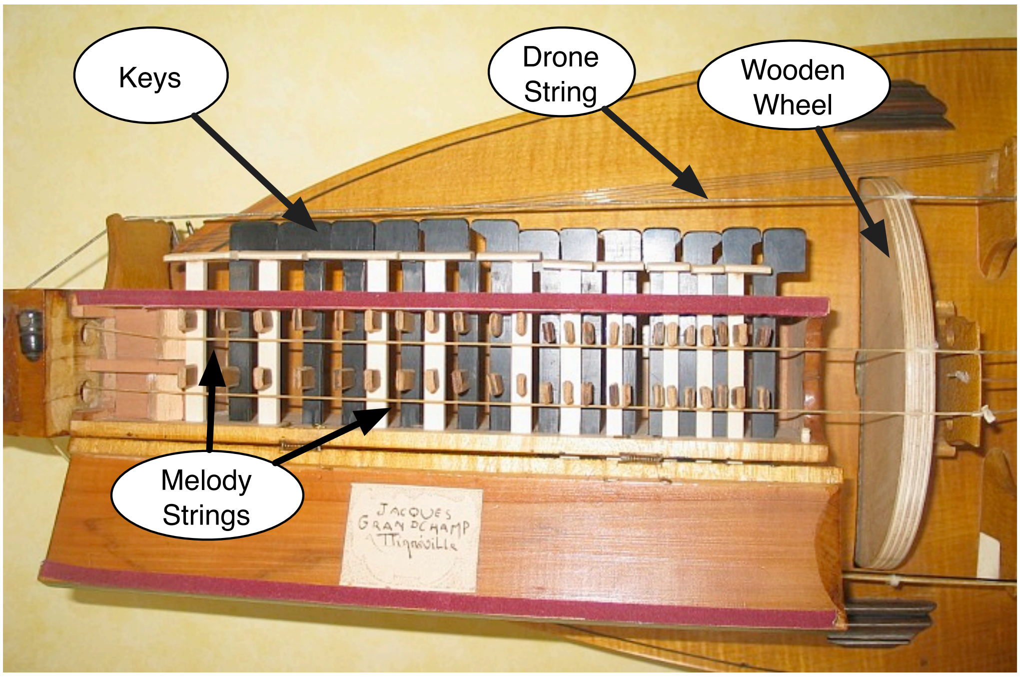

Figure 4.3: Hurdy Gurdy Mechanics (Source: Wikimedia Commons[a])

[a]http://commons.wikimedia.org

A.D.. Though the body shaping and keyboards changed through the centuries, the basic mechanics (see Figure 4.3) for sound generation rely on the same principles:

Mechanics

A set of strings is pulled over a wooden wheel that is covered with rosin. The wheel can be turned using a crank, causing the strings to vibrate. Melodies can be played using a small keyboard on the side of the instrument. The strings can be divided into three categories:

- *Drone Strings:* These strings are played all the time while the crank is being turned. They always play the same note and are not affected by the turning speed of the crank. Drone strings can be tuned to different notes and be switched on and off as needed for accompaniment.
- *Melody Strings:* The melody strings can be shortened at certain fixed positions using the keyboard, result-

ing in played notes at different pitch levels. Because of the position of the wooden wheel, only one melody note can be played at the same time. Higher notes will always have priority over lower notes here as they are closer to the wheel. This property can be used to create a warbler-like sound when playing. Like the drone strings, the melody strings are not affected by the turning speed of the crank and can be tuned as needed.

- *Dog Strings:* A dog string is basically a drone string that is held by a loose bridge. When the crank is being turned slowly, the dog string behaves like a normal drone string. However, when the crank is given a stronger impulse, the dog string makes the loose bridge vibrate. Players can use this to add a percussive element to their play, especially in dance pieces.

Different models

The number and types of used strings varies between different models. Some hurdy gurdies have up to 4 dog strings, earlier models work completely without them. The instrument can be tuned to play various melodic scales using the knobs that hold the strings at their ends and by adjusting the exact positions on which a button press shortens the melody strings. Some people understand the principles better by seeing a hurdy gurdy as a "mechanical violin".

Chapter 5

First Prototype: Exploring the Idea

"A journey of a thousand miles begins with a single step."

—Lao-tzu (604 BC - 531 BC)

A first prototype was built using Max/MSP[1] , a tool to easily prototype applications that rely heavily on audio, MIDI or image data. In Max/MSP, the developer creates "patches" in a graphical programming language that consist of blocks representing certain functions and lines representing the data flow between them. The schematic architecture of our patch is shown in Figure 5.1 .

Picking a sound

We used the standard Mac OS MIDI soundbank for this prototype. After a few tests, we decided to choose a saxophone sound to mimic the hurdy gurdy sound. Although the sound is not very similar, it is close enough for a first prototype and it had the necessary characteristics needed without actually having to model the sound of a real hurdy gurdy.

[1]http://www.cycling74.com

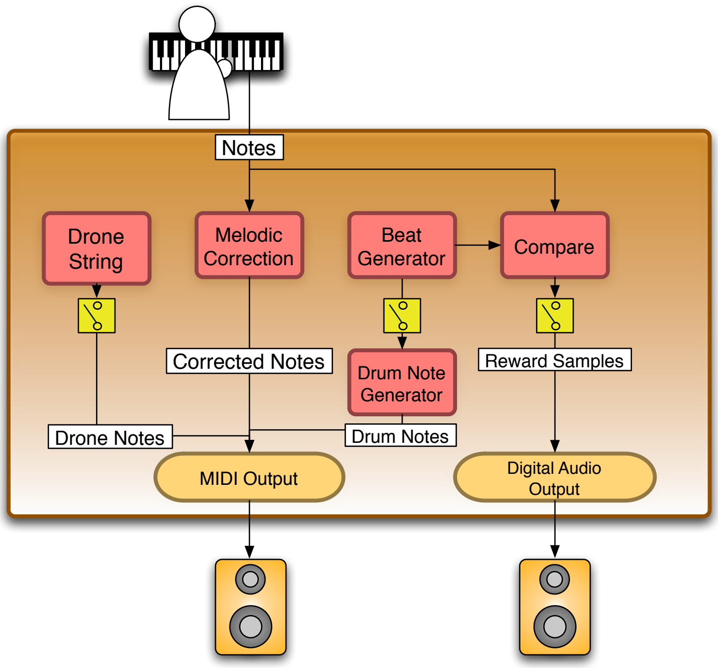

Figure 5.1: Max/MSP prototype schematic.

5.0.2 Features

Correcting "wrong" notes

- *Melodic Correction:* The user can play a note on the keyboard. The system will apply a static mapping onto the note and thereby move it into a predefined, static melodic scale (here C-major). The system then plays the note via a MIDI instrument.

Modelling a hurdy gurdy

- *Drone Strings:* A hurdy gurdy consists of different types of strings, some of which are playing the same note all the time ("drone strings", see Section 4.4—"Medieval Music" for more details). We modeled this characteristic by including two drone notes (tuned to C and G) that can be switched on and off seperately.

Rhythmic accompaniment

- *Basic Drum Track:* The system can create a very simple rhythm track consisting of a straight 4/4 bassdrum. This is used to give some rhythmic guidance to the user and is also the basis for measuring the user's playing performance.

- *Feedback:* The system rates the player's performance based on a simple metric, calculates a score and gives feedback accordingly. It computes the score solely from the player's rhythmic accuracy, using eighth notes as reference. For every played note that is within a certain interval of an eighth note in the given rhythm, the user gets a point. For every note that is not within that interval, the user loses a point. The score can never be lower than 0. If the player earns a certain amount of points, the system gives an acoustic feedback in the form of rewarding audio samples.

Rating player's performance

5.0.3 Helper tools

We also used a few other tools to help with the design of this prototype:

- SimpleSynth[2] is a simple software synthesizer tool that we used to access Apple's default General MIDI sound set.
- MIDI Monitor[3] displays MIDI commands that are triggered by or enter into the system. It is very useful for debugging purposes.
- MIDI Keys[4] offers a on-screen MIDI keyboard whose output can be sent easily to different destinations.

5.1 Implementation in Max/MSP

A Max/MSP patch allows to include patches in box representations ("sub-patches"). These can be used in higher level patches, creating a hierarchical patch structure. We embedded certain parts of the patch in sub-patches to increase readability and maintainability. We will show the core parts of the various features here.

[2]http://www.pete.yandell.com/software/
[3]http://www.snoize.com/MIDIMonitor/
[4]http://www.manyetas.com/creed/midikeys.html

5.1.1 Melodic Correction

When the system receives a note message, it is split into three components, which are treated seperately: Pitch, velocity and MIDI channel. The pitch value is passed through a sub-patch which is shown in Figure 5.2 .

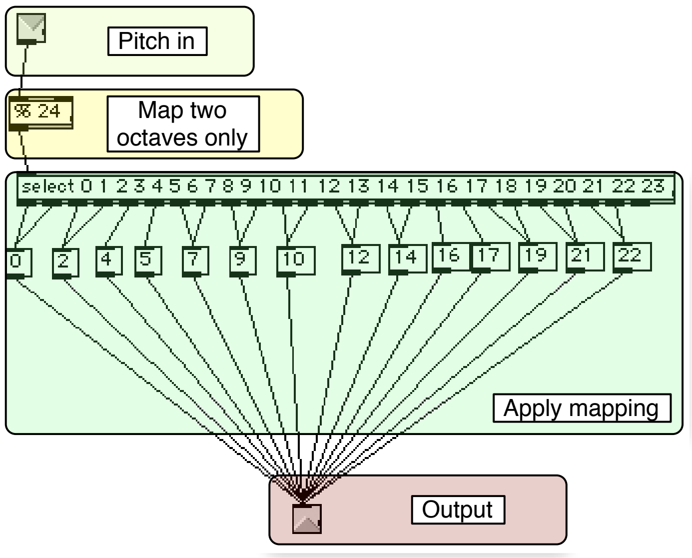

Figure 5.2: Sub-Patch implementing a static mapping of notes.

The pitch-value is first divided by 24, which means that the entire note spectrum is reduced to two octaves. This simplification seemed appropriate as a real hurdy gurdy also usually has no more than 2 octaves, sometimes even less. The select-patch, which is comparable to a "case"-statement in the C programming language, then maps all 24 possible notes to a C-major scale using the next lower note if the played note is not part of that scale. In the post-processing that comes after this sub-patch, the system adds 48 to the note value to prevent it from sounding too low. The note is then played with the initial velocity value, but on a different MIDI channel to prevent collisions.

5.1.2 Drone Strings

The system offers two drone strings that can be switched on and off as desired. We chose C and G as possible drone notes, which is common in hurdy gurdys. Playing the drone notes can be started or stopped using checkboxes, as shown in Figure 5.3 .

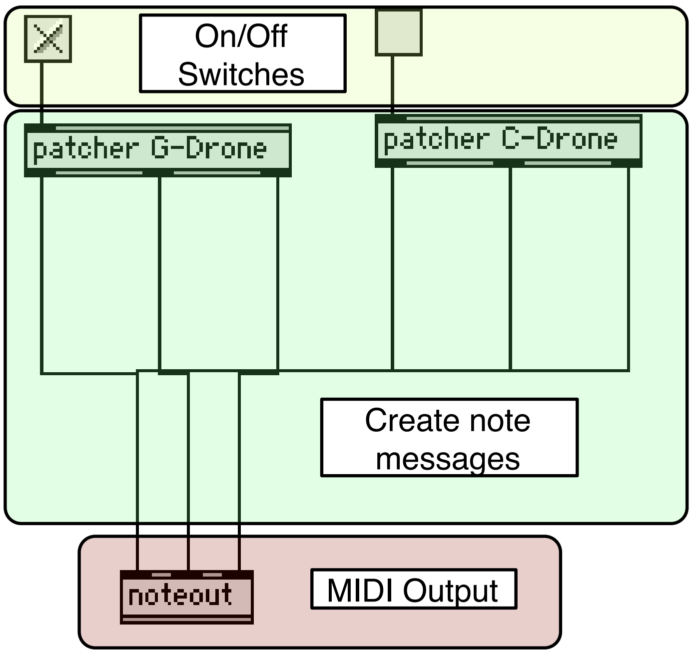

Figure 5.3: Drone Strings with switch boxes.

The check boxes function as triggers and status bits at the same time. Whenever they are checked or unchecked, a note-on (for starting the drone notes) or a note-off message (for stopping the drone notes) is composed in the C- and G-Drone sub-patches (called "patcher" in Max/MSP). As Figure 5.4 shows, the note messages again consists of three parts: pitch (43 for G), velocity (a medium value of 54 for on, 0 for off) and MIDI-channel (1, standing for channel 2).

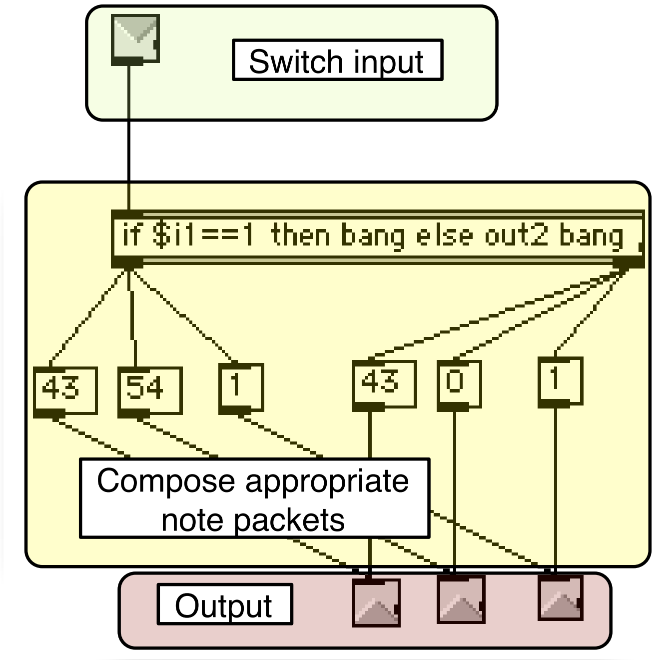

Figure 5.4: Composing the G-drone note message. Each MIDI note message consists of channel, pitch and velocity information.

5.1.3 Basic Drum Track

When switched on, a metronome sends trigger messages in regular time intervals. Whenever a trigger event is sent, the system creates and plays a note-on message (similar to 5.1.2—"Drone Strings") and a delayed note-off message.

5.1.4 Feedback

Generating Feedback consists of two steps:

1. Computing a score to rate the player's performance (see Figure 5.5)

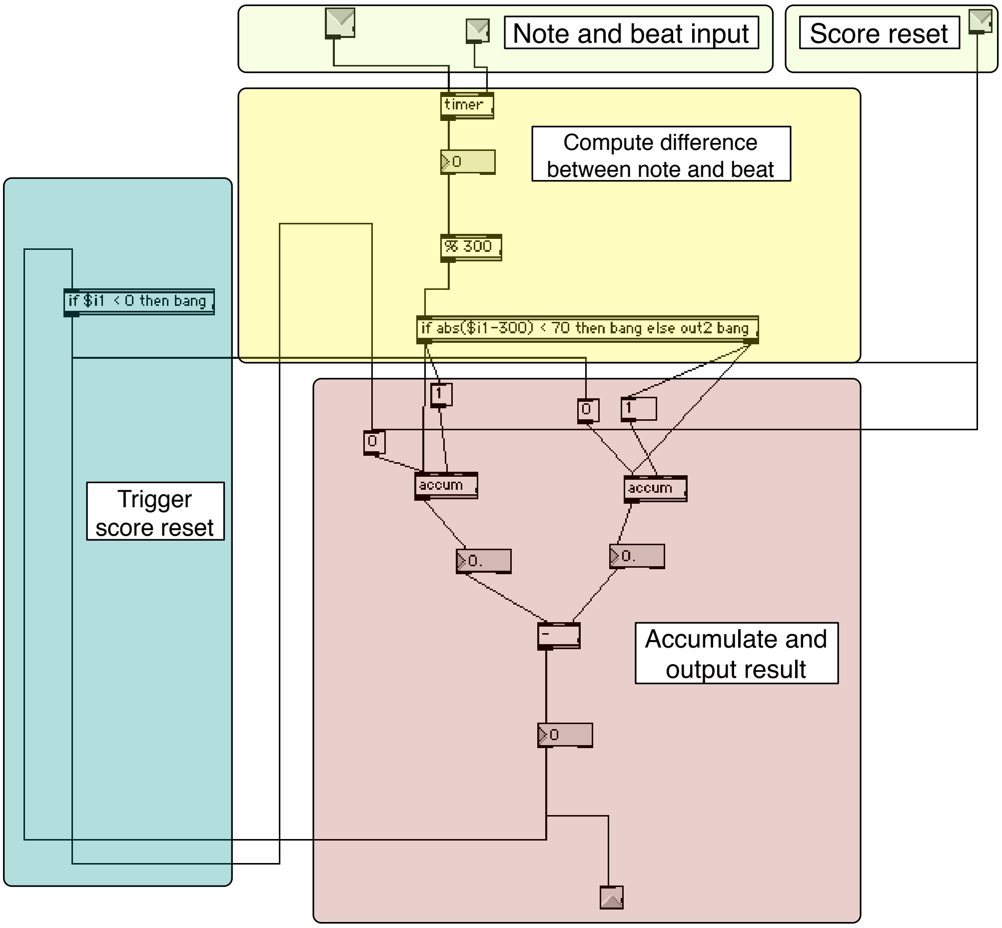

Figure 5.5: Sub-patch to rate player's performance

2. Playing rewarding feedback samples (clap, cheering) based on the score (see Figure 5.6)

Computing a score

The two inputs for this patch (see Figure 5.5) are the regular metronome beat and the note-on messages triggered by the player. A "timer"-patch computes the delay between the two events. If the time is within a fixed interval of an eighth note, the left accumulator is increased by one, otherwise the right accumulator is increased. The difference between the two accumulator values is the score. The "if"-patch on the very left checks if the score is below zero and resets it in that case. This reset can also be triggered through the rightmost

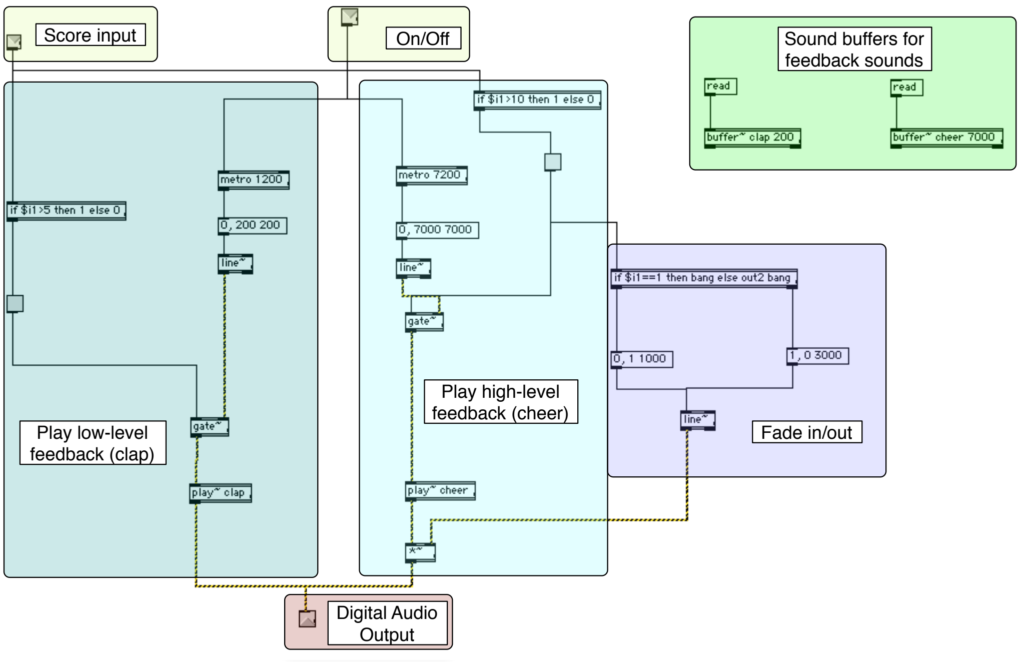

Figure 5.6: Sub-patch for playing feedback samples (yellow dotted lines indicate digital audio-related signals)

input manually.

Playing reward samples

Based on player's score, another patch (Figure 5.6) can play samples as rewards. If the player gets above a certain score level, a rhythmic clapping will be played. If he earns even more points, a cheering noise is played at regular time intervals. The samples have to be read into buffers first, which is done in the upper right corner. The inputs for this patch are score (left) and on/off (right). The "gates" work as switches to turn digital audio playback on and off based on the score. The right part is used to fade the cheering sample in and out to avoid making the sound start or end too abruptly.

5.2 Analysis: User Test

With the first prototype ready, we decided to run a preliminary qualitative set of user tests. Even though the prototype was still very rough and had some clear limitations (no proper modelling of hurdy gurdy characteristics in handling and sound, simple feedback mechanism, bad sound quality), we hoped that some data from users might help us in guiding further development. The system was set up with a standard MIDI keyboard and a set of stereo speakers connected to it.

Five people took part in this session. All of them were students from different fields and between 23 and 27 years old. Three of them had a classical musical education, the other two had no musical experience at all.

5.2.1 Procedure

We decided to do the first set of tests using a "quick and dirty" observation method as described by Preece et al. [2002] (chapter 12). Each participant was told that this was a first prototype for a future museum exhibit offering the possibility to play a hurdy gurdy and to experiment freely with it. We chose these very limited instructions because we felt this to be realistic for a museum setting where visitors will in most cases neither have deeper knowledge about the exhibit nor want to read lengthy instructions. The drone strings and rhythmic accompaniment were then activated. Users were free to experiment with the system as long as they wished. We took notes during that procedure and based a short non-formal interview at the end of the session about the user's experience on them.

5.2.2 Results

Even with the very limited scope of this test session, we were able to extract some interesting observations from user observation and interviews.

- After a short time of experimentation, all users played for 1-2 minutes. All users at least got the clapping feedback.

- Several users winced when the clapping started. They explained afterwards that it was too loud and started too abruptly.
- The piano-style keyboard showed to be a distraction: Those users who were familiar with playing a piano tried to play it accordingly (not like a hurdy gurdy); those who did not know how to play a piano found the classic piano keyboard layout deterring.
- The speed of the rhythm accompaniment was criticized as too fast in one case. After ad-hoc correction of this, the user seemed much more comfortable with the system.
- When asked about what caused the feedback, several users uttered the (wrong) assumption that it was triggered by certain keys or key combinations.
- Bad quality of instrument and feedback sounds were criticized by several users.
- None of the users had heard a hurdy gurdy before, only few had seen one. Some confused it with a barrel organ.
- Melodic correction was noticed and found confusing by users with musical education.

5.2.3 Design Implications

We discussed the occurred problems in the interviews at the end of the sessions. The users came up with some suggestions for improvement and we also discussed our own ideas with them.

- Even though there are hurdy gurdy models that have keyboard layouts similar to a piano, this would induce a wrong mental model for a lot of users. We communicated this result as a recommendation to project management to make sure a non-similar layout is picked.

- Feedback should not only consist of samples that are abruptly triggered at certain points in time. Instead, a "virtual audience" should be audible all the time to make feedback more natural.
- Good quality of all used audio content is essential (too low in first prototype).
- Balancing the different audio sources is of great importance for the effect of the exhibit. This has to be carefully adjusted in following user-tests.
- The offered user interface should look and feel more like the instrument it is supposed to mimic.

Chapter 6

Second Prototype: Preview System

"From error to error one discovers the entire truth."

—Sigmund Freud

The second prototype was the first prototype to be shown to the public. It was used during a two-week preview in July 2005 in which we presented early versions of our future REX-exhibits to potential sponsors, press, city officials and a limited number of Regensburg citizens. Due to these requirements, we decided to design a *vertical prototype* as defined by Nielsen [1993].

6.1 Excursus: Horizontal vs. Vertical Prototypes

When designing a prototype of any kind (paper, screen, ...), an important decision are the features of the final system to be included in the prototype. The focus of the current design phase helps decide which features of the final system should be shown in the prototype and which should be left out. Two common approaches are horizontal and vertical prototypes, as well as combinations of these. Figure 6.1 shows how the two types differ.

Horizontal prototype

A *horizontal prototype* shows every feature of the system, but only simulates their functions. For example, in a desktop system a horizontal prototype could show all the menus and GUI elements, but the responses to each command would be fixed and not dependent on real data.

Vertical prototype

In contrast, a *vertical prototype* shows only one or a few features of the system, but implements these features in a fully functional way. In a desktop system, this could mean that the user can access certain, but not all functions to modify a text or query a database.

Combinations of these two types are possible to adjust the prototype for the designer's needs.

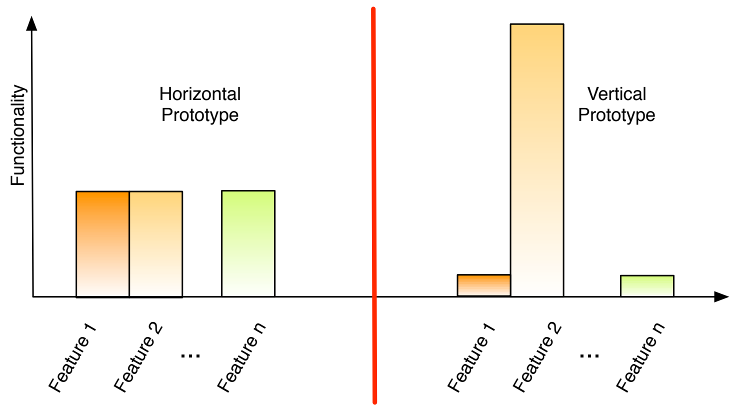

Figure 6.1: A horizontal prototype limits the depth of the interaction, a vertical prototype limits the breadth of the interaction.

6.2 Design

When we started working on this prototype, we already knew that this would not only be another experimental prototype, but that it needed to be a fully working version. It was both our next iteration in the development process leading towards the final version of the exhibit and a product in itself that was to be used for presentation purposes.

Due to lack of time and funds, we decided to follow a vertical prototyping approach. While the final version was supposed to be installed in a special room, we decided to do a semi-mobile solution with only one instrument for this prototype. Our idea was to modify a replica of a real hurdy gurdy and equip it with electronic components. The hurdy gurdy would then be mounted onto a stand that included the necessary computer and speakers. We chose a hurdy gurdy because it seemed relatively easy to build an electronic version of it.

Requirements

Building on our experiences from the first prototype, there were slightly different requirements for the second one:

- better sound quality for all used audio sources
- real instrument replica as user interface
- behavior of the real medieval instrument should be modeled in software
- computer and electronics need to be hidden and safe out of reach of the user
- system should create an audible, constantly present atmosphere
- feedback needs to be more varied and natural

Tools

The second prototype was developed under Apple's Mac OS X[1] , using Xcode[2] as development environment and incorporating Apple's Core Audio[3] and QuickTime[4] libraries. All code was written in C, Objective C and C++. While Max/MSP was a good choice for a first rough prototype, these different technologies allowed us more control over the behavior of the system.

Hurdy gurdy replica

We were able to find an instrument builder from Regens-

[1] http://www.apple.com/macosx/
[2] http://www.apple.com/macosx/features/xcode/
[3] http://developer.apple.com/audio/coreaudio.html
[4] http://www.apple.com/quicktime/

burg who could make an authentic replica of a medieval hurdy gurdy for our exhibit. This instrument was made to look completely realistic, but did not include any strings or other acoustic sound generators since we wanted to do the whole sound generation electronically. This allowed us greater control over the sound and made melodic correction possible. For more information on how we modified the instrument for our needs, see section 6.3.5—"Modifying the Hurdy Gurdy".

Accompaniment

The accompaniment track was recorded especially for our exhibit by members of the ensemble "...sed vivam!". They played a rather simple medieval dance piece with a harp, a hurdy gurdy, a fiddle and several percussion instruments. Each instrument was recorded on a separate audio track; the hurdy gurdy even was recorded as two separate tracks so that we could play the drone and dog strings independently from the melody strings.

Atmospheric background sounds

To create a constant audio-atmosphere, we decided to play the sounds of a medieval tavern or pub all the time while the exhibit is running. We used two looped pre-recorded audio-tracks with different lengths that were started simultaneously so that only very rarely both audio tracks would reach their end at the same time. We also recorded voice audio samples from one of our musicians that were triggered in random time intervals. These included humorous remarks to encourage visitors to use the exhibit.

6.3 Implementation

A rough structural overview over the second prototype is shown in Figure 6.2 . The modified hurdy gurdy can send two types of MIDI data: Note-messages which are sent when one of the keys is pressed or released and data packages that are sent when the crank is turned (sent as pitch wheel controller messages). The incoming MIDI packages are multiplexed to different components according to their type when they are received by the system.

Incoming notes are statically mapped to notes that melodi-

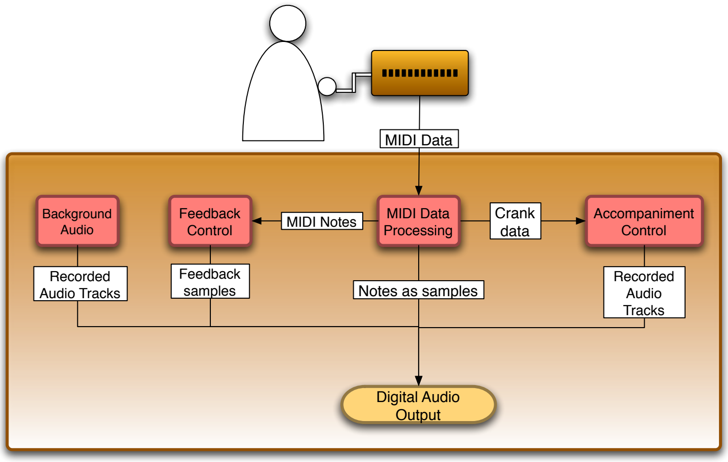

Figure 6.2: Functional Overview Over The Second Prototype.

cally fit to the accompaniment piece. These new notes were sent out via MIDI and received by a software sampler tool (VSamp[5]) that we prepared with a soundbank containing high-quality samples of single hurdy gurdy notes. We also used this tool to play all other shorter audio samples that only need to be triggered at certain points in time, such as encouragement samples.

6.3.1 The Exhibit in Idle-Mode

When the second prototype is started, it only shows a small status window, as shown in Figure 6.3 . This window allows to manually switch the accompaniment track on and off using a radio button. It also shows the progression of the playback of the accompaniment and background noise tracks, using the standard Apple QuickTime controls.

As long as no one has touched the exhibit or has not touched it for some time, the system is in this state. It will play the background tracks all the time and at random time intervals trigger an encouragement sample or the sound of a musician tuning his instrument. The system will not

[5] http://www.vsamp.com/

start the accompaniment track until a visitor starts playing the hurdy gurdy or the accompaniment radio button is switched to “on”.

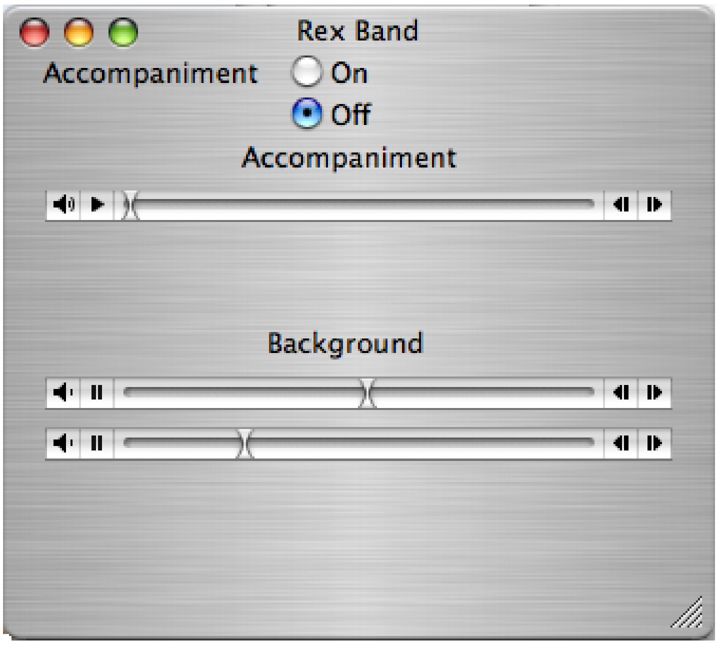

Figure 6.3: Status display of the second prototype

6.3.2 Playing the Exhibit

When a user starts turning the crank on the hurdy gurdy, the system starts playing the accompaniment track. This includes the recording of the piece as well as the drone and dog string tracks from the recording (not the hurdy gurdy melody tracks). The user can play to the piece using the buttons on the hurdy gurdy. When he stops turning the crank, the piece will continue playing for 10 seconds and an encouraging voice sample is triggered. No hurdy gurdy sound is played when the user does not turn the crank. If the user has not resumed playing during that time, the system assumes that he has left and returns to idle-mode.

6.3.3 Modeling a Hurdy Gurdy

We modeled the behavior of our electronic hurdy gurdy to behave mostly like a real hurdy gurdy. However, to ensure that even a novice user without any knowledge of medieval music or the instrument could use the exhibit successfully, we left out or simplified some features of a real hurdy gurdy. Except where noted otherwise, all features listed here are features that the original instrument also has or that at least exist in some models:

- Only one note can be played at a time. When two or more buttons on the hurdy gurdy are pressed, only the note corresponding to the highest key (i.e. the one closest to the crank) is played.
- The hurdy gurdy starts playing only when the crank is being turned. When the user stops turning the crank, he can not play notes any more.
- The melodic correction ensures that only certain notes can be played. Since not all hurdy gurdy models had a full twelve tone keyboard and different tunings existed, we can assume this to be mostly realistic.
- While the user turns the crank, the pre-recorded drone and dog strings play all the time.

We decided not to include a feature that would allow the user to set his own rhythmic accents using the dog string. Based on our user tests, we assumed that so few people knew about this subtle detail that it would only confuse people who are not already familiar with playing a hurdy gurdy.

6.3.4 Giving Feedback

Marking beats

Like the first prototype, the second prototype also plays reward samples when the user plays rhythmically exact. To measure this exactly, we first analyzed the accompaniment tracks and marked the regular rhythmic beats in them using a tool called BeatTapper (see Figure 6.4). BeatTapper was developed within our group and has been used in several other projects.

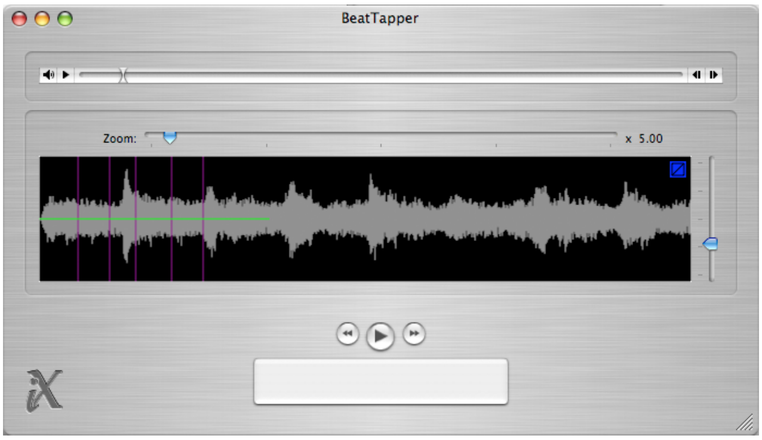

Figure 6.4: BeatTapper, a tool to mark beats in audio recordings.

BeatTapper can visualize the waveform of a given audio or video file, allows setting and editing beat marks and exports them as a list of time values.

These time values can then be compared to the time values of user-triggered notes. Like in the first prototype, the user gains a point if he is within a given interval and loses a point if he is not. The score is reset whenever the system returns to idle mode.

Reward samples

The system can play reward samples in two levels that are also played by the software sampler tool VSamp. Another soundbank that is addressed by a certain MIDI channel number contains a wider variety of reward samples than we had available in the first prototype. The samples are picked at random. The list of samples contains both approving voice samples and sound effects, e.g. the sound of a falling coin.

When the user plays to the end of the piece, a rewarding applause is played if the user has played well enough. The intensity of the applause also depends on the user's performance. During the applause, the user can not play any notes and no accompaniment will be played. After a short timeout, the system will reset to idle-mode.

6.3.5 Modifying the Hurdy Gurdy

The hurdy gurdy replica for our exhibit was provided by Alois Biberger[6] , an instrument builder from Regensburg. His replica was inspired by early medieval hurdy gurdy models, having a simple box-shaped corpus and its eleven keys arranged linearly (see Figure 6.5). The buttons had no real function, but felt realistic through a guitar string that was used as a flexible barrier (the string itself does not produce any sound); the buttons could be pushed up to the point where a small hook inside the box reached the string, then some more with increasing pressure but not more from a certain point onwards.

We used a Doepfer CMT64[7] board to generate the MIDI signals that were sent to the computer. This board features connectors for simple switches as well as potentiometers. When a switch is closed, the board sends a note on-message, when it is opened, the board sends a note-off message on a previously selected MIDI channel. Potentiometer changes are sent as pitch-bend-, modulation-, volume- or aftertouch-messages. Figure 6.6 shows how the board is mounted on the inner part of the hurdy gurdy's bottom lid.

We equipped each of the wooden bars that are part of the buttons of the original hurdy gurdy with a standard off-the-shelf microswitch. When one of the buttons is pressed, the microswitch is pushed into the inner wall of the hurdy gurdy which will activate the switch and trigger a note on-message. The microswitches were mounted using hot glue and metal brackets (see Figure 6.7).

When looking for a sensor to measure the turning of the crank, we decided for a reflection light barrier consisting of an infrared LED and a photo sensor. This is also a commercially available electronic component and sold as one piece

[6] http://www.klangbaeckerei.de
[7] http://www.doepfer.de/ctm.htm

Figure 6.5: Our modified hurdy gurdy.

with the LED and the sensor facing each other in an obtuse angle. When the user turns the crank, he also turns a small wooden wheel inside the hurdy gurdy that we sticked a Siemens star upon. The light from the LED is reflected by this wheel and creates a different output from the photo sensor depending on if it is currently facing a black or a white segment of the Siemens star (see Figure 6.8). Using a small electric circuit, we were able to connect this output to one of the potentiometer inputs so that it is accepted as a potentiometer input. This enabled us to read the sensor data as pitch wheel controller-messages in MIDI format.

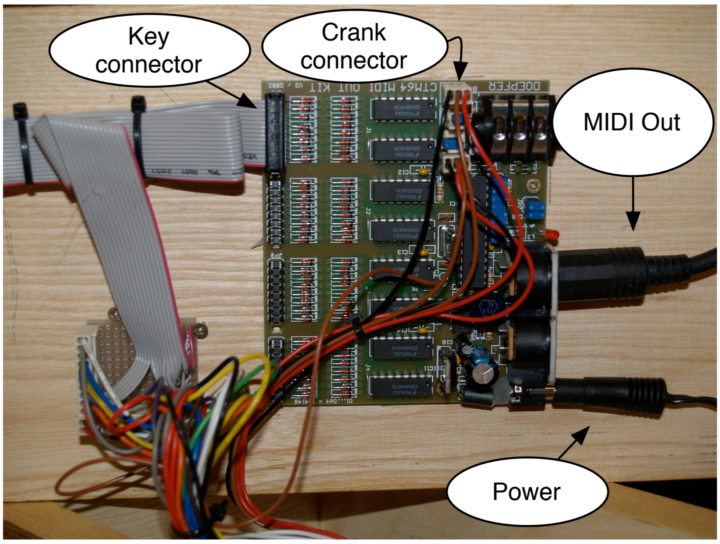

Figure 6.6: Doepfer MIDI board mounted in the hurdy gurdy.

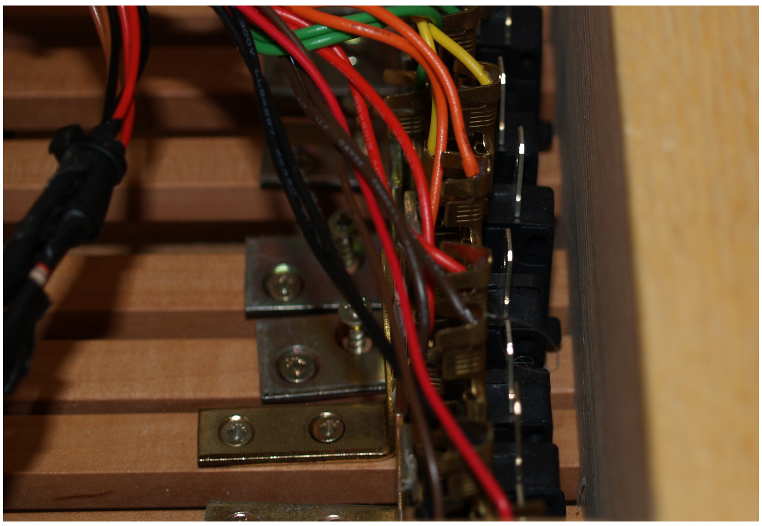

Figure 6.7: Microswitches inside our hurdy gurdy.

Figure 6.8: Measuring the turning of the crank using a reflection light barrier.

6.4 Analysis

This prototype was analyzed in multiple ways. We ran another set of ”quick and dirty” user tests about one week before setup in Regensburg. We were also able to get an experienced hurdy gurdy player from a local medieval music ensemble to try our exhibit and give comments. During the preview in Regensburg, we also had a couple of users try out the prototype and we encouraged comments and criticism.

6.4.1 User Test

With only little time before having to set up in Regensburg, we were able to do user tests with three users. All of them were students, between 23 and 25 years old, with different levels of musical education. The test setup included an Apple PowerMac G5, our modified MIDI hurdy gurdy and a set of stereo speakers. While the users seemed to enjoy using the exhibit, a couple of usability problems occurred.

Again, a big problem for all of the users was the lack of familiarity with the instrument. None of the users had seen a hurdy gurdy before, one of them confused it with a barrel organ. The buttons on the hurdy gurdy were not recognized as such at first; some users tried pressing the wrong ends of the little bars holding the buttons, which did not have any effect at all. One tried to use the pegs that hold the strings in a real hurdy gurdy but which do not have any effect in our modified version. However, after being pointed to the real position of the buttons, all users were able to play the system successfully.

To all of the users it was also unclear what exactly they could influence in the music. None of them had even heard a hurdy gurdy before, so they found it hard to identify the music played by themselves in the stereo signal.

The users perceived the accompaniment track as pleasant, though some comments about the mixing of the tracks were made.

None of the users got to the highest feedback level.

6.4.2 Expert Review

Before going to Regenbsurg, we were able to find a hurdy gurdy-player who has been playing in a local medieval music ensemble for a few years. He agreed to take a look at our prototype and help us with some comments:

- He liked the background sounds and said they sounded authentic compared to the medieval-style events he knew.
- He only got to the highest feedback level after several tries. He suggested that this should be made easier.
- The other instruments were too dominant compared to the hurdy gurdy sound. He suggested to turn them down and let the fiddle only play at the end of the piece, because otherwise the hurdy gurdy and the fiddle (both melody instruments) sounds would clash.
- The drone and dog strings were too silent compared to a real hurdy gurdy.
- It should be made clearer which sounds are produced by the hurdy gurdy.

- He suggested that the bars should be lubricated with graphite to increase durability.

He also helped us find some bugs, such as feedback samples that were triggered too often. These were also fixed before the setup.

6.5 Last Minute Modifications and On-site setup

With the experiences from the test and review sessions, we were able to make some modifications to the system before the preview exhibition. Though most of them are, from a technical point of view, minor modifications, our user tests confirmed that subtle details are important for making the exhibit a success.

Adjustments

Modifications to the software included re-balancing the different audio sources, lowering the score levels that were necessary to get positive feedback from the system and using the left and right channels of the output to split the hurdy gurdy sounds and all other audio output from the system; the intention behind this last change was to connect two speakers to the computer running the software and placing one very close to the hurdy gurdy and the other one farther away so that it is easier to tell the sounds of the hurdy gurdy and the other accompaniment instruments apart.

Physical setup

We already knew that the hurdy gurdy was to be mounted onto a stand, but the stand itself had not been built yet. In cooperation with a carpenter from Regensburg, we made sure the stand would fit to our requirements. In the end, we had a stand design where the hurdy gurdy was mounted on top with the MIDI and power cables led through a hole in the bottom into a lockable drawer in the stand; the drawer contained a laptop computer, MIDI interface and one speaker. Holes in the drawer lid that were placed directly under the hurdy gurdy ensured that the hurdy gurdy sound would be perceived as coming directly out of the instrument. A small, but relatively powerful speaker with a good sound quality (Fostex 6301 [8]) was used; it even created noticeable vibrations similar to a real instrument with a resonant body.

6.5.1 Feedback from Visitors

The exhibit was shown during a two-week preview during which selected groups (press, potential sponsors, city guides, school classes, etc.) were given guided tours through the exhibition. However, it was made clear that this was not the final REX exhibition and that the exhibits were still prototypes.

Even though the exhibit was not fully complete yet and the conditions under which it was shown were slightly different than in a complete exhibition, we encouraged visitors to give feedback and got some interesting responses.

Most people gave positive comments about the exhibit and while some (especially older) people seemed to prefer classic, non-interactive exhibits in a museum setting, the feedback was in general very encouraging.

Suggestions for improvement included the following:

- Offer more information about the hurdy gurdy and how it works.
- Explain functions of keys (e.g., notes becoming lower when user plays more on the left keys).
- Allow to switch off correction (or support on multiple levels).
- Make feedback more noticeable.
- Give a melody (in simplified notation) to play.
- Let users select more than one piece.
- Use surround sound.

[8] http://www.fostexinternational.com/docs/products/6301b.shtml

From our point of view, the requests for more information and documentation are certainly valid and we will have to provide posters, signs and explanations for the final exhibit. We also intend to use surround sound for the final exhibit and use a better sound system in general.

However, considering our experiences from user tests, we are somewhat hesitant to offer more control and less support from the system. A selection mechanism for the level of feedback would make the user interface more complex, and only very few people would benefit from such a feature as most people are not familiar with medieval music and its instruments.

Chapter 7

Experimental Feature: Rhythmic Correction

"You know how it is when you go to be the subject of a psychology experiment, and nobody else shows up, and you think maybe that's part of the experiment? I'm like that all the time."

—Steven Wright

As REXband is targeted towards a wide audience of people who do not necessarily have any musical education, we wanted to include features that can help people without or with only little musical experience to have a successful and enjoyable interaction experience. Our melodic correction helped with only one musical dimension (melody). We were curious if rhythmic correction would also help the user accomplish better and more satisfying results while making sure such a feature would not unnecessarily limit musical expression.

7.1 Algorithm Design

Our idea was to delay any notes that are played by the user when they are within a certain time interval before a marked rhythm beat. If a note is not within that interval, it should just be played without any delay. See Figure 7.1 for an overview of how our algorithm works.

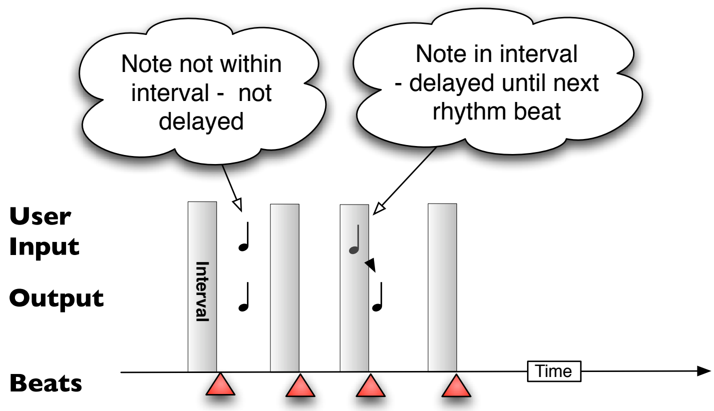

Figure 7.1: Rhythmic correction: user input is only delayed within a certain time interval around a rhythmic beat.

While the user plays to a given rhythm, the notes he plays are analyzed by our software. Each note is treated differently depending on its relative position the the rhythm beats:

1. If the user input comes in within a certain time interval before a rhythm beat, it is delayed until this beat.
2. If the user input is outside any of these intervals, it is just played without any delay.

7.2 Literature Review

Researchers are usually sceptic about introducing artificial latency, as it is usually noticed by users and interferes with responsiveness, which is considered important in interactive systems. For example, Borchers and Mühlhäuser [1998] express their skepticism like this:

"If the delay between musical user input and corrected system output becomes noticeable (more than about 150 ms), users will lose the feeling of playing an instrument and start thinking that they merely control some artificial music generator."

While this is certainly a valid statement, it still leaves open if user's could benefit from a delay that is less than 150 ms. It also does not answer for which scenarios this applies.

Rhythmic quantization in Jam-O-Drum

Blaine and Perkis [2000] experimented with a rhythmic quantization feature in the Jam-O-Drum system. Applying different quantization (correction) levels, they tried to quantize the input of the users to the "next incidence of the music's intended beat". However, they found this ineffective as this accentuated late responses for unskilled players and disconnected cause (hitting the drum) and effect (playing sound) in the user's perception.

Isolated events

DiFilippo and Greenebaum [2004] cite several studies about perceived timeshifts between different signal types. For touch leading audio, a difference of 66 ms is assumed as not noticeable. The same threshold is assumed for video leading audio. However, the referenced authors only tested this for single, isolated events, not for a sequence of audio signals such as music.

DJs

Beamish et al. [2004] developed a multimodal interface for DJs that would let them control digital audio with a turntable-style interface. They found that experienced DJs could even notice a delay of 10 ms or less.

7.3 Experiment Concept

These sources show that no clear answer to how much latency a user would typically notice in our scenario can be given. They also do not tell if people could actually benefit from a rhythmic correction in the way described above.

We decided to run our own study to answer these questions. Due to the apparent complexity of these issues, we decided not to try to give general answers to latency perception, but to conduct a set of experiments that are tailored towards our own design challenges. Our study consisted of two steps:

1. Determining a threshold below which a latency-based rhythmic correction is not noticeable for a given piece
2. Setting the correction interval to the value determined in step 1 and test for perceivable improvement

7.4 First Experiment: Determining Threshold

The first experiment was inspired by the classic methods of psychophysics, developed by Fechner [1889]. While more exact methods have been developed in the meantime, these are relatively easy to implement and we expected to get enough data for a reasonably exact estimate on a threshold value. Given the range of assumed perceivable latencies from literature review, we did not expect to get an exact value that would work for all users, so we decided to work with a method that would determine the threshold by decreasing from a relatively high value or increasing from a very low value. The actual correction events were logged to a file.

Experiment software

We wrote a software tool that implemented the correction algorithm as described above. The tool could play a piece from a digital audio file and correct the user input in an interval that could be set with an on-screen slider. The slider could also increase or decrease automatically between two values while the program is playing an audio track.

7.4.1 Setup and Tasks

The experiment was set up as shown in Figure 7.2 . The participants of the experiment were to sit in front of a standard MIDI keyboard that was connected to an Apple Power Mac Dual G5 that was running the experiment software. The screen was not visible to the participants so that they could not see the current latency value.

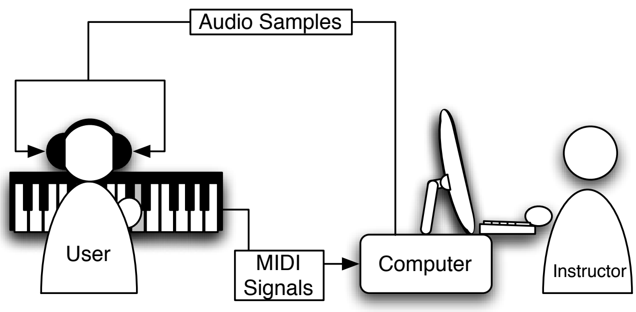

Figure 7.2: Setup for the threshold experiment.

Instructions

The participants were instructed to play a one-octave C-major-scale (only white keys) to a rhythm accompaniment that was played over a pair of headphones. The participants should only play this scale up- and downwards all the time while the music was playing and try to make it fit to the rhythm (it was left open how exactly this could be done).

Each participant had to do two tasks:

1. While the latency value was slowly increasing from 0 ms, the participants were told to stop playing immediately when they noticed a latency for the first time.
2. While the latency value was slowly decreasing from 200 ms, the participants were told to stop playing immediately when they *did not* notice any latency any more.

We found the system latency to be below 1 ms and therefor negligible.

The stop-values were recorded for each task. Also, we used two different rhythm accompaniments for each experiment session:

1. A medieval-style tambourine rhythm taken from our recording for the second prototype.

2. A straight, computer-generated bass drum-track at the same speed of the tambourine track.

Each participant had to do both tasks for both of the rhythm accompaniments. We varied both the order of the pieces and the order of the tasks to avoid training effects. Each experiment session took about 15 minutes. Participants were compensated for their time with chocolate.

7.4.2 Results

A total of 15 people participated in the study. 7 were hobby musicians with varying levels of musical experience, 8 participants claimed to have no musical education.

As expected, the variety of the results was quite high. Threshold values for the tambourine accompaniment ranged from 35 to 200 ms, with an average of 130.8 ms (standard deviation: 49.3). The results for this piece can be seen in Figure 7.3.

For the straight rhythm, we calculated an average of 144.8 (standard deviation: 65.8). Figure 7.4 shows the results.

7.4.3 Discussion

As expected, the results for subjectively perceivable latency varied greatly with 40 ms being the lowest value. However, to ensure that most users will not perceive any correction while still having a reasonably large time interval for correction, we picked 100 ms as our correction threshold. This corresponds to the delay that was not noticed by 2/3 of the users (see Figure 7.3). Although some very sensitive people might be able to detect an even smaller delay, we assumed that 100 ms will not be detected by the majority of users, especially if they are not actively trying to detect the delay.

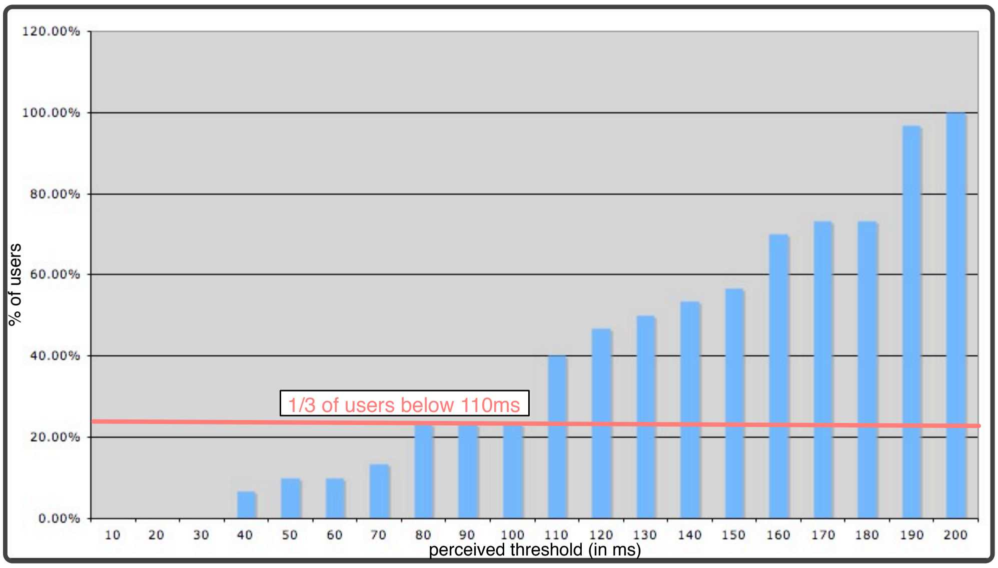

Figure 7.3: Cumulative results for medieval rhythm (in %).

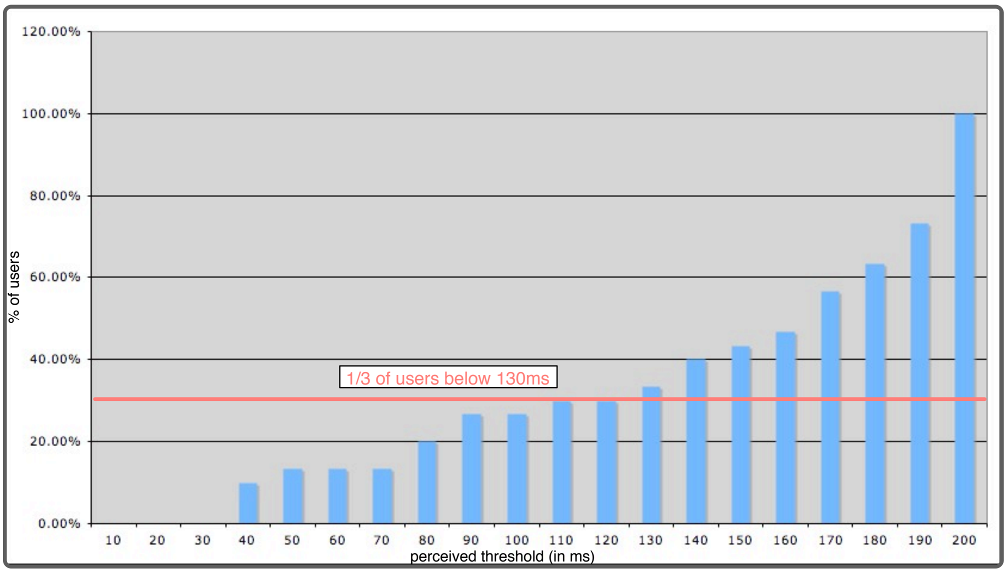

Figure 7.4: Cumulative results for straight rhythm (in %).

7.5 Second Experiment: Testing Usefulness

After determining 100 ms as a non-noticeable threshold for rhythmic correction, we wanted to find out if users benefit from that feature.

7.5.1 Setup and Tasks

To have both the perspective of a listening bystander as well as an active user of our system, we let each user do two tasks:

1. Let participants listen to two recordings of a previous user. One would be played with, the other without rhythmic correction. The participant should then decide which one was better and how big the difference was.

2. Let participants play twice to the medieval rhythm. One trial would be with, one without correction. Let participant rate which trial was better and how much.

We varied the order of the two pieces (with/without correction) to avoid training effects. The rating of the trial was done using questionnaires. Each participant had to decide for one trial or recording (forced choice) and then rate how much he thought it was better, on a scale ranging from 1 (hardly better) to 5 (much better). The basic setup was the same as shown in Figure 7.2, but the software was set to a fixed correction value (100 or 0 ms).

7.5.2 Results

A total of 10 people participated in the user study. 4 were hobby musicians with varying levels of musical experience, 6 participants claimed to have no musical education.

In the listening task, only 44% of participants were able to identify the corrected recording (one did not participate in the listening task). Of the users who were able to identify the corrected recording, the average improvement in comparison to the non-corrected recording was rated with a 3 on average, while it was rated with a 3.6 from the users that did not correctly identify the recording.

In the playing task, 40% of users correctly identified the trial in which they were supported by the system. Also, 40% of users stated that their own performance was better in the corrected trial.

7.5.3 Discussion

At first sight, the results look disappointing. While the correction interval of 100 ms that we determined in the first experiment seemed reasonable and could not be reliably distinguished from no correction, the users in the second experiment were generally not able to benefit from correction. Some users even were not able to tell in which trial they were supported by the system.

However, we do not see this experiment as a failure. Apart from learning about how much task and context influence the perception of latency, the informal post-session discussions with participants brought up some interesting ideas. It seems that playing to the rather complex medieval rhythm was considered a difficult task for a lot of users, even though no specific performance was expected. This shows that the original idea of making the system support users while playing can still be considered as valid. We will take this as a design lesson and experiment with other ways to ease this task, such as audible cues to emphasize the timing of the accompaniment piece. Concrete design ideas and test results can be found in the next chapter.

Chapter 8

Third Prototype: Final System

"Art is never finished, only abandoned."

—Leonardo da Vinci

The third iteration of the REXband system was designed to be the final exhibit to be installed in the REX building. Apart from using the hurdy gurdy from the second prototype, we modified another two instruments (harp and frame drum) to send MIDI signals.

When we started working on the third prototype, it became apparent relatively soon that the opening of REX would be delayed to after the deadline for this work. Nevertheless, the design of the software as well as the sensing hardware in the instruments can be considered as final. Changes can be expected on the setup side as it was not completely clear where and how exactly the exhibit is supposed to be installed at this point.

8.1 Design

As the third prototype was the first version to support multi-user interaction, both the design of the software and the exhibit in general as well as the used tools and hardware had to be significantly different from the second prototype.

The software for the exhibit was rebuilt from scratch as the architecture from the second prototype was too inflexible to control more than one instrument. However, we used the same software tools and libraries (Xcode, QuickTime, Core Audio, VSamp).

New instruments: Harp and frame drum

Two new instruments (harp and frame drum) were again custom built by Alois Biberger from Regensburg.

The frame drum consists of a wooden body with a closed back and a leather drumhead (see Figure 8.1). It is stuffed with styrofoam to minimize acoustic sound and has an output connector for the sensor on its back.

Figure 8.1: Our modified frame drum. To minimize acoustic sound, we stuffed it with styrofoam. The back is closed and has an output connector for the sensor.

The harp consists of a wooden frame and a resonant body that contains our sensors (see Figure 8.2). The body is also stuffed with styrofoam. The strings are nylon guitar strings.

Sensors

After evaluating several possible solutions, we decided to

Figure 8.2: Our modified harp. The resonant body contains vibration sensors for each string and is stuffed with styrofoam to minimize acoustic sound.

use the i-CubeX system from Infusion Systems[1] to modify the harp, and a Roland drum trigger module connected to a Roland TMC-6 trigger MIDI converter to modify the drum. Both systems output MIDI signals that can be easily processed by our software. See Section 8.2.2—"Hardware" for details.

After our experiences with rhythmic improvisation aids

Rhythmic support

[1]http://www.infusionsystems.com

(see Chapter 7—"Experimental Feature: Rhythmic Correction"), we decided to support users in playing to the rather complex rhythm of our medieval dance piece differently. We added a separate audio track to the accompaniment piece that emphasized the regular beat with the sound of stomping feet. We consider this as a very natural, unintrusive way of support that also adds to the impression of a virtual audience in the exhibit space.

8.2 Implementation

8.2.1 Software

The software for the final exhibit consists of four main components:

- A *MIDI signal chain*, handling the processing of the data coming in from the instruments,
- a *state machine* that models the system behavior in different stages of the interaction,
- a *state controller* that controls the state switching, and
- a *main controller* that is responsible for creating all the modules, organizing communication and managing central data.

The MIDI signal chain and the state machine can communicate using internal system messages ("notifications") so that the components do not have to know each other's interface.

An overview over the system architecture is given in Figure 8.3.

MIDI Signal Chain

The MIDI signal chain consists of one MIDI in-module handling all the MIDI input, a seperate module for each instru-

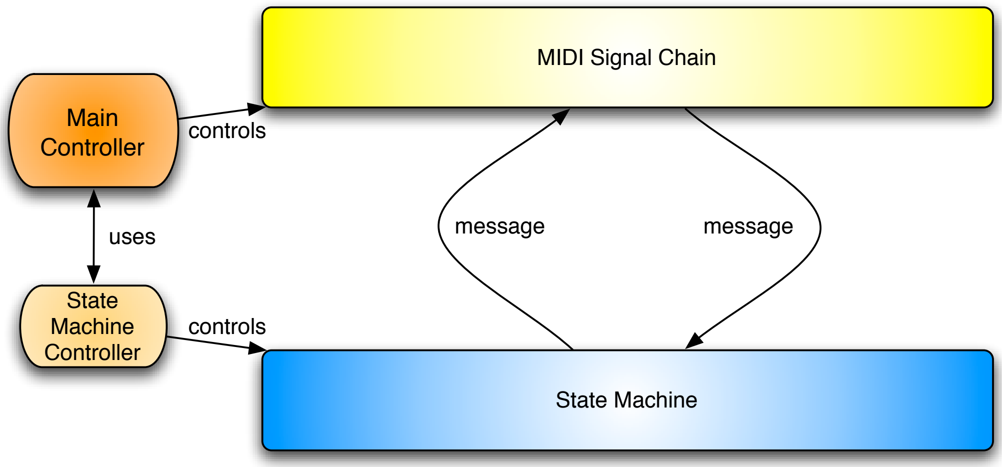

Figure 8.3: Final System Architecture: State machine and MIDI signal chain can communicate with messages and do not need to know each other's interface.

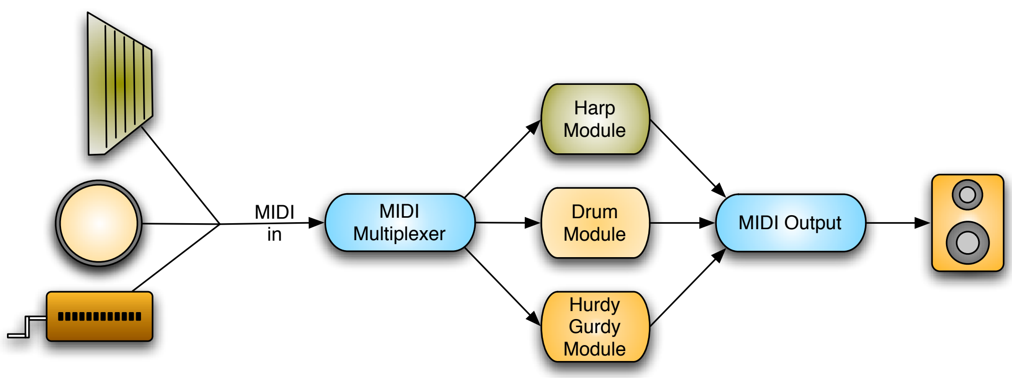

Figure 8.4: MIDI Signal Chain: Each instrument's data is processed by a seperate module.

ment that models the instrument's behavior and an output module. The MIDI signal chain can be seen in Figure 8.4.

Each instrument module can also send activity- and inactivity-messages that indicate when a user is playing the instrument or when it is not being used. For the harp and the drum, any kind of input will trigger these messages. For the hurdy gurdy, only the turning of the crank will trigger an activity-message. These messages are used by the state machine to decide when to switch between states.

State Machine

The system's runtime behavior is determined by a state machine wherein only one state is active at any given time. The possible states are *idle*, *play*, and *reward* which are activated under certain conditions.

- After startup, the system is in its *idle*-state. Only background noise and occasional attracting sounds (encouraging shouts, instrument tuning sounds) are played.
- When a user starts to play an instrument, the system switches to the *play*-state. An accompaniment track plays for around one minute and the user can improvise to it. The user's performance is rated based on a similar rhythmic criterion as in the second prototype and rewarded by playing various samples (cheers, claps, coins falling, etc.).
- After playing until the end of the accompaniment piece, the system switches into the *reward*-state. Any audio output from instruments is blocked while this state is active, and an applause sample will be played. The intensity of the applause depends on the player's performance. After the applause has ended, the systems returns to idle-state.

When all of the users lose interest in playing while the exhibit is in its play-state, the system will return to its idle-state after a timeout of 10 seconds. An overview of states and switch-conditions can be seen in Figure 8.5.

Activating a state

All states are initialized at startup time and are activated and deactivated as necessary. The declaration of each state follows a defined protocol, a concept in Objective C that is similar to subclassing from an abstract class in other programming languages. This protocol defines several methods that each state has to implement. A state is activated by calling its `enter` method which can do necessary initializations and then starts the state's run loop.

Deactivating a state

To deactivate a state, its `stop` method has to be called. The

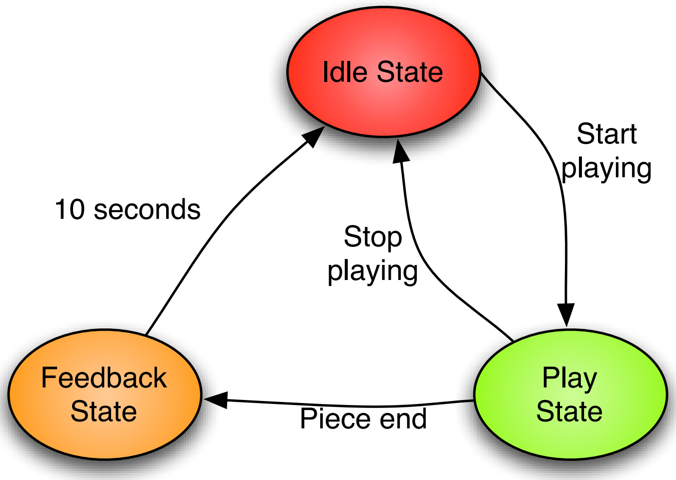

Figure 8.5: States and switch conditions.

run loop will then exit after its currently running iteration and call the `exit` method.

State Controller

The state controller is responsible for initializing the state machine, taking care of proper switching between states and direct communication with the controller. To avoid too much interference, all other classes can only communicate with the controller indirectly through messages.

State switching

When the state controller receives a relevant event (e.g., a user has started playing one of the instruments), it checks the currently active state. If the event should trigger a state switch, the state is made inactive and the appropriate state is activated.

Main Controller

The main controller module initializes all classes after startup, sets up messaging functions, controls playback of the accompaniment track and computes the player's score. The main controller is the only object that can communicate with most objects directly.

8.2.2 Hardware

Before the other two instruments could be built, we had to assess different technical solutions for modifying the instruments to send MIDI data. After considering several possible solution including designing own sensor hardware, we decided for using off-the-shelf hardware. Even though we could not influence the exact behavior of the sensors then, the lower effort to modify the instruments and the possibility to easily replace broken pieces in the running exhibits justified this decision.

Hurdy gurdy update

For the hurdy gurdy, we did not change the sensor configuration, but only replaced the original switches with more robust versions. The old switches also produced a clearly audible click noise when pressed, a problem which we could fix with the new switches. For more information on the original hurdy gurdy modifications, see Section 6.3.5—"Modifying the Hurdy Gurdy".

The following sections explain how we modified the harp and frame drum.

Harp

After deciding for the i-CubeX Digitizer controller and matching vibration sensors (see Figure 8.6), the instrument could be built. We discussed several options with the instrument builder Alois Biberger, and he then built the instrument with a relatively big resonant body that could contain both the sensors and mounting points for the strings. We filled the resonant body with styrofoam so that the instrument itself would produce only very little sound.

Figure 8.6: i-CubeX Vibe Sensor.

Processing the harp's output

Each of the 12 instrument strings was led through the little tube at the end of each sensor. Using a custom editor software provided by its manufacturer, we programmed the controller to output the vibration measurements MIDI note values. Each sensor's information was sent as a different MIDI note number while the intensity of each string's vibration was encoded as the note's velocity value with a range of 0-127.

A typical velocity curve is shown in Figure 8.7. We implemented an algorithm to detect sudden increases in velocity values, similar to a rising edge trigger. This algorithm takes the first velocity value of a peak and sends out a note on-message with this velocity value. This algorithm can not detect if the velocity curve rises above its trigger value, but a difference-based peak detection algorithm would introduce a significant delay while waiting for the next value.

Frame Drum

We decided to use a drum trigger from electronic drum systems for modifying our frame drum to send MIDI signals. Our solution contained a Roland drum trigger sensor, a rubber foam cone that sits between sensor and drumhead to protect the sensor from direct hits and a Roland TMC 6

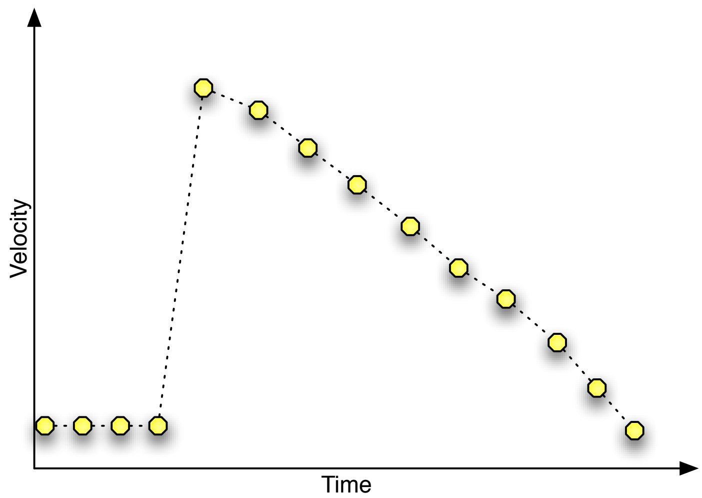

Figure 8.7: Typical value curve of velocity values in note on-packets after plucking a string on the harp.

trigger MIDI converter. See Figure 8.8 for a schematic view on our drum modifications.

The TMC 6 does not produce any sound on its own but only generates MIDI messages that can then be processed by our software. When the drum is hit, it sends a note on-

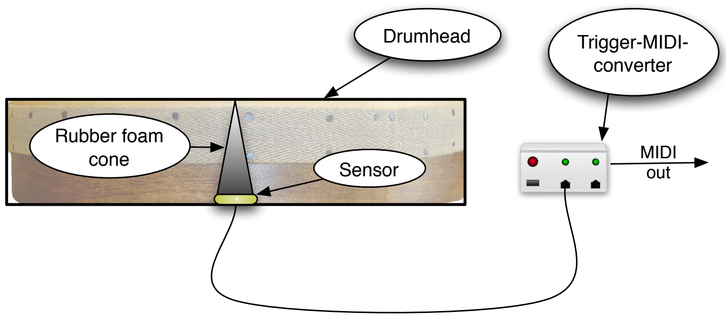

Figure 8.8: The drum trigger system

message with the intensity of the hit encoded as velocity value. Velocity values also differ between hits close to the sensor in the center of the drum and hits closer to the rim of the drum. Unfortunately this makes it hard to distinguish a soft hit in the center of the drum from a hard hit on the rim that would allow to trigger different sounds in different areas. We decided to leave such a feature out for now, even though a real drum would make this possible.

8.3 Evaluation

Setup

Even though most of the decoration and hardware for the final exhibit had not been built yet, we decided to run a set of user tests using an improvised setup (see Figure 8.9). All the instruments were fixed onto a table, with speakers standing in front of each instrument. To preserve the concept of invisible hardware, we hid the speakers, wiring and most of the electronics under a piece of cloth.

Hardware

Our software was running on a standard Apple PowerMac G5. We used a multichannel audio interface from MOTU[2] to route the output to different speakers. Apart from speakers in front of each instrument, we used the pre-installed audio system in our lab for accompaniment and background sounds.

Our aim was to test if REXband reaches three main goals:

1. REXband provides an enjoyable experience.
2. REXband conveys information about medieval music.
3. Users find it easier to improvise with the added rhythm track.

We decided to test the system in a controlled environment in our labs as well as in a more public setting. The latter was done in cooperation with a local museum. We also did an additional feedback session with members from our group who used a cognitive walkthrough technique.

[2]http://www.motu.com/

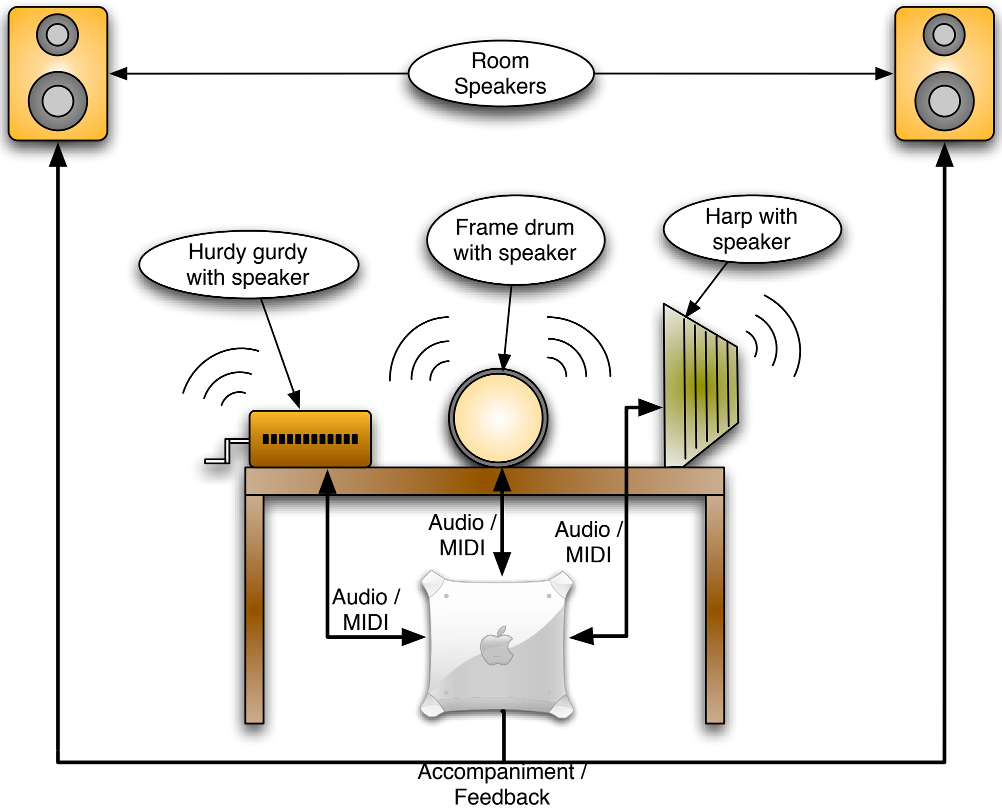

Figure 8.9: Setup for user tests: The instruments were mounted onto a table, and speakers were put in front of them. All non-instrument audio was played using the room speakers.

8.3.1 Controlled User Test

Looking for general experiences and feedback concerning our system, we decided to gather qualitative data rather than quantitative data in this test session. The tests were done in our labs with students as participants. Test sessions were done in eight groups of two or three people to test the multi-user functionality of the system.

Procedure

Instructions and procedure

We decided to do an observation study with a retrospec-

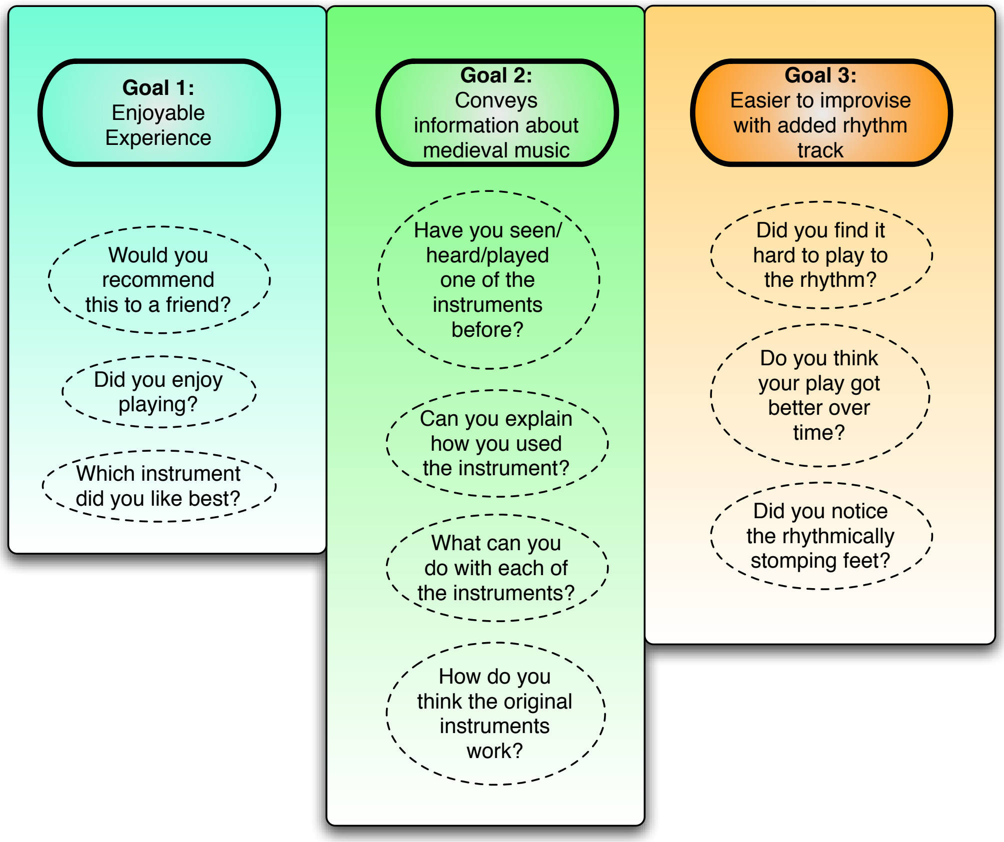

Figure 8.10: Our three main goals and associated interview questions to test if they were met.

tive interview at the end of each session. Instructions for the participants were kept simple: We asked the members of each test group to pick one instrument and play it until the end of the accompaniment piece. The users should then move to another instrument. We repeated this four times with every group, so that each user would play each instrument at least once.

The interview included questions to check if we met our goals as well as more general questions about participants' musical background and overall impression of the system. Figure 8.10 shows some of the questions we used and the hypotheses they are associated with. We also asked some more general, open questions to encourage users to make suggestions for improving the exhibit. As mentioned in

8.1—"Design", we added an extra audio track to the accompaniment that consisted of the sound of rhythmically stomping feet. This track was aligned with the beat of the piece to facilitate playing to the rather complex medieval rhythm. For our tests, we switched this track off for two trials to test if users benefit from this.

Results

Observations

Our observations allow only very limited conclusions on how the system was used and perceived. We did not interrupt the users and kept answers to questions fairly short to not disturb the interaction. However, we were able to observe some reoccuring actions and behaviors.

Most users playing the harp or the hurdy gurdy for the first time approached them curiously, but carefully. Some users were not sure on how to hold and play the hurdy gurdy properly, and some even did not find the buttons on the backside of the instrument at first. Fortunately these effects were corrected by the other users who tended to help users with fundamental problems with the hurdy gurdy.

For the harp, we observed a learning effect when users had the chance to play it more than once during the four trials. One user only plucked single strings in his first trial and tried multiple strings in his second one. Another user started with glissandos (i.e., quick successions of neighboring notes played in an ascending or descending order with one hand), but played more single notes later, which apparently sounded better to her.

Different playing techniques were tried with the frame drum. Users experimented with single- or double-handed playing styles as well as drumming with their fingers only.

Interview results

When asked afterwards, only three of our 18 participants claimed to have seen a hurdy gurdy before. None of the users have ever played a hurdy gurdy. Harp and drum were more familiar instruments: All of our test users had seen or heard at least one version of these instruments before. Only one user has played a harp before, but did not see himself as an experienced harp player.

Hurdy gurdy knowledge

When asked for characteristics of the hurdy gurdy, responses varied greatly. Two users confused it with a barrel organ. No user was able to fully explain the hurdy gurdy, but every group identified at least some characteristics successfully (e.g., connection between crank and sound production, drone sound, can play single notes only).

For the harp, playing glissandos and plucking single strings were the most common playing techniques. Only four users experimented with plucking two or more strings simultaneously.

8.3.2 Public Test

We installed our system as a temporary exhibit for one evening in the Couven Museum[3] in Aachen, Germany. This museum focusses on furniture from the 18th and 19th century and also shows reconstructed rooms from that time. Although most exhibits there referred to a different time period than our exhibit, they invited us to show our exhibit during a special event on July 15th 2006. Many visitors came to the museum on that evening, and being allowed to set up our exhibit close to the entrance, we had approximately a few hundred test users. Contrary to our previous test sessions, we had planned to stay more passive and only observe people use the system.

Observations

We soon found out that this was difficult due to the surroundings. Most other exhibits in the museum were old and valuable and visitors were not allowed to touch them; several security guards were present in the museum to enforce this rule. Our exhibit was apparently the only one that was meant to be touched and used. After realizing this problem, we started approaching visitors directly and invited them to try out our exhibit. Most visitors followed

[3] http://www.couven-museum.de

Figure 8.11: Setup for the Couven Museum. All instruments were mounted on a table, with the wiring and electronics hidden under decorative cloth.

this invitation, and in many cases other visitors tried out the exhibit without explicit encouragement after seeing it in use.

Feedback from the visitors was in general very positive. A lot of people told us how much they liked the idea and that it was fun to play on the instruments. One visitor told us that she had always wanted to play the harp, but never actually took lessons and was very happy to have the chance to try our harp.

Many visitors asked questions about the hurdy gurdy, both about the original instrument as well as our modified version. No visitor seemed to be familiar with this instrument. This also led to users standing on the side of the hurdy gurdy while playing it; while this is possible, the most appropriate position would be behind the instrument (turning the crank with the right hand, pressing the keys with the left hand).

Before trying out the system, some visitors were hesitant about playing themselves. Lack of musical ability was an excuse we often heard. However, when we told these visitors that our instruments were easier to play than the original versions and even supported users, some could be convinced to try out the exhibit.

Several users put their ears close to the instrument speakers to hear what sound they were producing themselves. This seemed especially hard to distinguish from the other sound sources when all instruments were used at the same time and when the room was very crowded.

Hurdy gurdy and harp were the most popular instruments, but some users tried the frame drum first. We assume that this instrument was attractive especially for visitors with low confidence in their musical abilities.

8.3.3 Cognitive Walkthrough

We did a feedback session with members of our group who did a cognitive walkthrough to evaluate our system. All attendees had a background in HCI with varying levels of experience. The session was led by Prof. Dr. Jan Borchers.

Suggestions

All participants were given the opportunity to try out the exhibit themselves as well as observe others interact with it and ask questions. Some comments were made about responsiveness problems of the harp and the frame drum. A fine tuning of the sensors seemed appropriate to the attendees.

A longer discussion arose when we mentioned the problem of rhythmic support. Several suggestions were made:

- Before the music starts, there should be a voice counting in. This would be relatively easy to implement as we have an appropriate voice sample that we currently use in the attract mode.
- The rhythm should be shown on a screen or a static note display using a spotlight. Some comments were made on if this would distract users' attention from the instruments.

- An added drum track could help emphasize the rhythm better. Although it would mean some effort to record an additional, authentic drum track, this would certainly be possible and might help users understand the rhythm better.

To avoid confusion on how to hold the instruments and where to stand, footprints should be painted on the floor to mark the correct standing positions.

8.3.4 Discussion

We consider the feedback from these three evaluation sessions as very valuable. It showed us that REXband is attractive as an exhibit and can reach its two main goals: Being fun and enjoyable to use and teach users some basics about medieval music in a novel and unique way.

However, we have also identified some aspects that are important to consider when designing the system setup for Regensburg Experience:

- Visitors need to know that the exhibit is meant to be touched and used. This must be made clear by placing it into an appropriate context.
- Most people know little about medieval music and its instruments. We have to provide on-site information to answer frequently asked questions (e.g, on hurdy gurdy mechanics).
- Balancing the different audio sources is important for how the exhibit is perceived by visitors.
- Instruments should be placed neither too close nor too far away from each other; users should feel neither disconnected from the other players nor overwhelmed with audio that is not coming from their instrument.

Unsolved problem: Rhythmic support

The only real disappointment is the still unsolved problem of rhythmic support. While the added rhythm track does not seem to distract users, they could benefit only little from this feature. We assume that a different, simpler rhythm track would make playing easier for musical novices. Unfortunately it is still unclear what could be a reasonable tradeoff between historic authenticity and rhythmic simplicity. The considerable effort to create a new multitrack recording is also a barrier to answer this question with scientific methodology. However, we are considering adding an extra drum track to the accompaniment.

Chapter 9

From Prototype to Exhibit

"You don't need eyes to see, you need vision."

—Maxi Jazz from Faithless

With the third prototype working and tested, we would like to sketch some ideas for the final exhibit. The ideas presented in this chapter show what we envision based on our own design ideas and experience with users.

9.1 Physical Setup

REXband is an exhibit that relies on audio for the interaction. This is both a good thing and a challenge: the exhibit can attract users without visual contact, but can also be distracting for museum visitors who are currently looking at or interacting with other exhibits.

To avoid distraction, we plan to install the exhibit in a semi-secluded area in the exhibit space; an additional wall could block audio from the inside while still allowing easy access to the exhibit. See Figure 9.1 for a sketch.

Wall-mounted speakers should be used to play the accompaniment and feedback sounds, while the instrument sounds should come from speakers close to the instruments. The instruments should be mounted on stands to

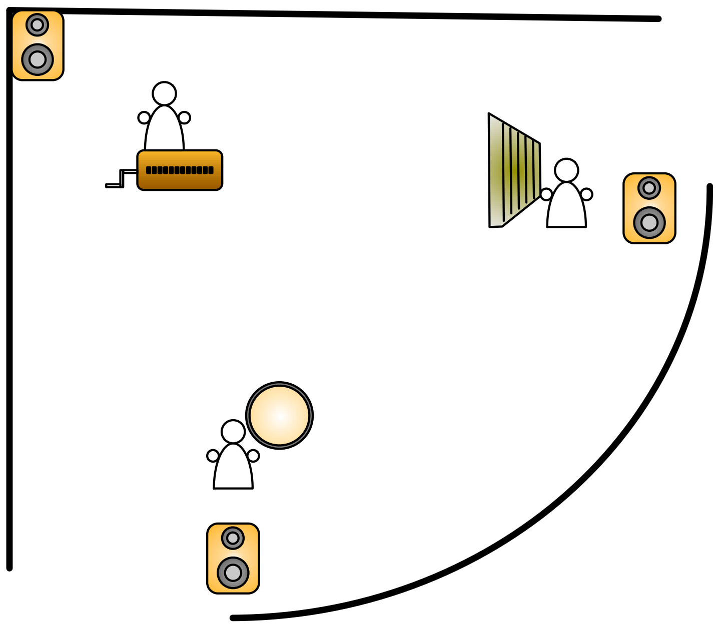

Figure 9.1: Top view on the exhibit setup. The exhibit could be installed in a semi-secluded corner to limit the audio spread. The instruments are placed so that users can see each other while playing.

avoid visitors taking them away or damaging them; this would also avoid problems with wiring, as the necessary wires (e.g., MIDI cables) could be effectively hidden. The stands could also be used to contain electronic components, such as audio interfaces, speakers, or MIDI interfaces.

Hurdy Gurdy

A stand for the hurdy gurdy has alread been built for the preview exhibition in summer 2005. Figure 9.2 shows a schematic sketch.

Frame drum

In its original form, the frame drum is meant to be held in one hand and hit with the other. To simulate this, we would like to mount the drum at a 45° angle with the user facing its side. Figure 9.3 shows how the drum should be mounted.

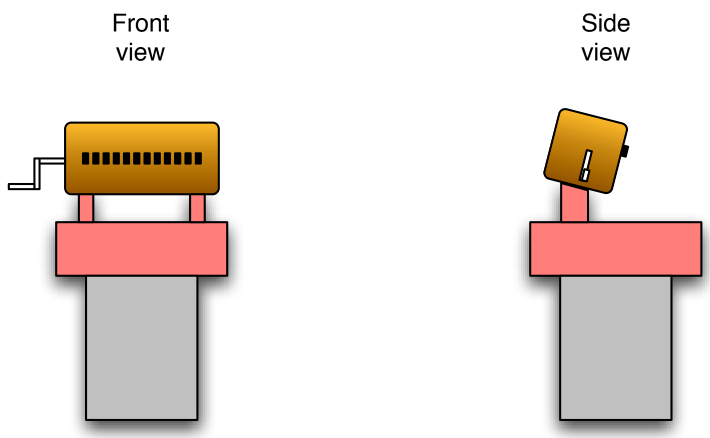

Figure 9.2: Hurdy gurdy and stand. The stand provides enough space for electronic components, such as a speaker mounted under the instrument, a MIDI interface etc..

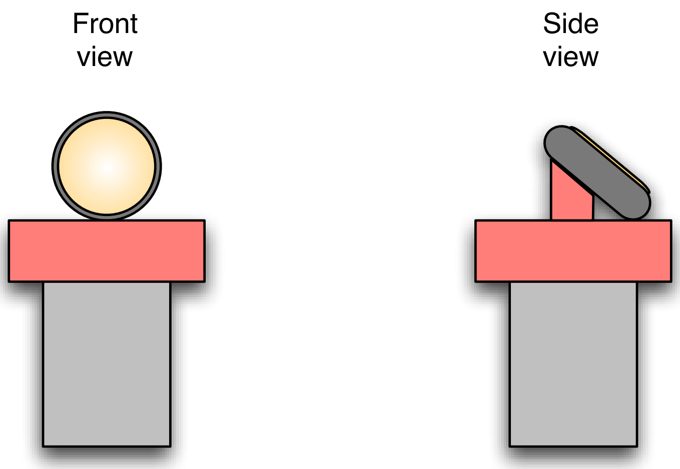

Figure 9.3: Frame drum and stand. The drum is mounted so that it encourages a similar playing style as the original handheld frame drum.

Harp

The harp is meant to be played with the player facing its diagonal part, reaching at the strings from both sides. Figure 9.4 shows the harp and its stand.

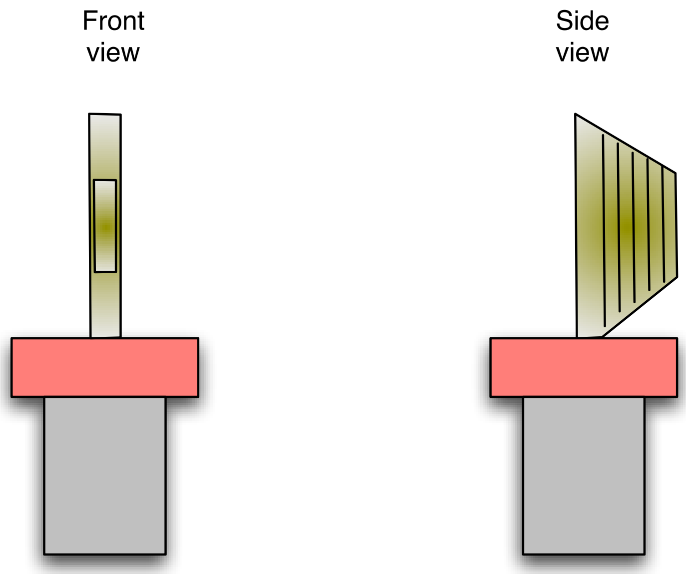

Figure 9.4: Harp and stand. The player should stand in front of the harp, facing its diagonal part.

Exhibit space

The exhibit space should also contain information on the instruments, e.g. as posters. Painted footprints could show where visitors should stand while playing the instruments, and pictures could help with holding the instruments properly.

As our user tests have shown, balancing the different audio tracks is important for users' perception of the exhibit. No general solution can be given to that problem, but we believe that careful adjustment will minimize it. It is also an open question what a good distance between the instruments is; users playing them should be able to tell their instrument's sound apart from the other instruments' sounds without feeling disconnected from the other palyers. On-site user tests will help us to adjust this.

Chapter 10

Summary and Future Work

"The best way to predict the future is to invent it."

—Alan Kay

The work at hand explains how REXband was developed in three iterations. While we believe to have developed an interesting and unique system, our work is not complete yet. The REX exhibition will open after the deadline for this work, and while we consider the basic system architecture as reasonable and robust, some changes in appearance and functionality can be expected.

This last chapter summarizes our work until today and gives an outlook on what further research and development is possible.

10.1 Summary and contributions

The system described in this work is a collaborative computer music system. It is designed as a museum exhibit that allows visitors to experience medieval music by playing on authentic replicas of medieval music instruments (hurdy gurdy, harp, frame drum). A computer system in the background receives the players' input, applies a melodic correction algorithm, simulates the playing behavior of the instrument, plays an accompaniment track and gives acoustic feedback on the player's performance.

The whole development process followed an iterative design approach that includes alternating design, implementation and testing phases. REXband was developed in three iterations:

- The *first prototype* was developed in Max/MSP. It was created to test the concept and run a first preliminary user study.
- The *second prototype* featured an authentic replica of a medieval hurdy gurdy. It was presented and tested during a two-week preview exhibition in summer 2005.
- The final exhibit will be based on the *third prototype*. Offering replicas of a hurdy gurdy, a harp and frame drum, it allows collaborative interaction with several medieval instruments.

The design process was influenced and evaluated by experts in medieval music, user tests and an extensive study on latency perception that tested an experimental algorithm for rhythmic correction.

10.2 Future work

Creating the exhibit

While a lot of design decisions have already been made, some problems concerning the exhibit are still open. Most of the hardware has not been bought yet, and the setup of the exhibit (decoration, instrument stands, documentation) is still incomplete. Our user tests and literature review indicate that the perception of the system strongly depends on the context, so we will work with project management to support the decision process and influence it to maximize the exhibit's impact. We are also working on making our instrument replicas (especially the frame drum) more responsive and behave more realistically.

Our specific focus left out some interesting research aspects that could be pursued in the future:

Artistic expression

Both the instruments as well as the system was created for robustness and ease of use, not for depth in artistic expression. A system with a focus on artistic expression could allow to switch off melodic correction and change the mapping from sensory input to sound generation.

Connecting different concepts

The melodic correction algorithm used for REXband is fairly simple and relies on a static mapping of incoming to outgoing notes. Systems like coJIVE and BoB (see Chapter 3—"Related work") follow a more complex approach that could provide interesting results when coupled with modified instruments such as the ones used for REXband.

Rhythm and latency perception

Our experiments with rhythmic correction and our literature review in this context showed that the perception of rhythm and latency is not yet fully understood. More experiments in that direction could show how people with varying level of musical experience perceive rhythm and latency and how a computer music system could provide support.

Instrument modification

Modifying each of the instruments used in REXband was a unique challenge. Examining the possibilities of modifying other less common instruments to send MIDI signals offers endless possibilities for creative solutions.

Collaboration

While REXband is a collaborative system, the psychology of collaboratively playing music has only been touched in this work. Further research could provide knowledge about interaction patterns in this context and show how a system can support players in collaboratively playing music.

Bibliography

Christopher Alexander. Keynote speech. OOPSLA'96 11th Annual ACM Conference on Object-Oriented Programming Systems, Languages and Applications, October 6-10 1996.

Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Contruction.* Oxford University Press, 1977.

Timothy Beamish, Karon Maclean, and Sidney Fels. Manipulating music: multimodal interaction for djs. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 327–334, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-702-8. doi: http://doi.acm.org/10.1145/985692.985734.

Tina Blaine. The convergence of alternate controllers and musical interfaces in interactive entertainment. In *Proceeding of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Paris, France*, 2005. URL `http://www.nime.org`.

Tina Blaine and Tim Perkis. The jam-o-drum interactive music system: A study in interaction design. *Proceedings of the ACM DIS 2000 Conference*, 2000. URL `http://www.jamodrum.net/`.

Jan Borchers. *A Pattern Approach to Interaction Design*. John Wiley & Sons, Ltd, 2001.

Jan Borchers. Worldbeat: Designing a baton-based interface for an interactive music exhibit. In *Proceedings of the ACM CHI'97 International Conference on Human Factors in Computing Systems*, pages 131–138, 1997.

Jan Borchers and Max Mühlhäuser. The design of interactive musical systems. *IEEE Multimedia*, 5(3):36–46, July-September 1998. ISSN 1070-986X.

Jan Buchholz. A software system for computer aided jazz improvisation. Master's thesis, RWTH Aachen, 2005. URL `http://media.informatik.rwth-aachen.de`.

Andrea R. da Silva, Marcelo M. Wanderley, and Gary Scavone. On the use of flute air jet as a musical control variable. In *Proceeding of the 2005 International Conference on New Interfaces for Musical Expression (NIME05), Paris, France*, 2005. URL `http://www.nime.org`.

Derek DiFilippo and Ken Greenebaum. *Audio Anecdotes*, chapter Perceivable Auditory Latencies, pages 65–92. A K Peters, 2004.

Stuart Favilla and Joanne Cannon. Children of grainger: Leather instruments for free music. In *Proceeding of the 2006 International Conference on New Interfaces for Musical Expression (NIME06), Paris, France*, 2006. URL `http://www.nime.org`.

Gustav Theodor Fechner. *Elemente der Psychophysik*. Breikopf & Härtel, 2. edition, 1889. URL `http://gutenberg.spiegel.de/fechner/epsypht1/epsyphi1/epsyphi1.htm`.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley, 1995.

James Jerome Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

Michael Kerres. *Multimediale und telemediale Lernumgebungen*. Oldenbourg, 2001.

Jonathan Klein. A pattern-based software framework for computer aided jazz improvisation. Master's thesis, RWTH Aachen, 2005. URL `http://media.informatik.rwth-aachen.de`.

Teemu Maki-Patola, Perttu Hämäläinen, and Aki Kanerva. The augmented djembe drum - sculpting rhythms. In

Proceeding of the 2006 International Conference on New Interfaces for Musical Expression (NIME06), Paris, France, 2006. URL `http://www.nime.org`.

Jakob Nielsen. *Usability Engineering*. Morgen Kaufmann, 1993.

Donald A. Norman. *The Design of Everyday Things*. Basic Books, 2002.

Piet Pollack. Helmut gotschy – einer der bekanntesten drehleierbauer. *Folker!*, 6, 1999. URL `http://www.folker.de/9906/gotschy.htm`.

Jenny Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design*. Wiley, 2002.

Byron Reeves and Clifford Nass. *The Media Equation*. CSLI, 1996, 2002.

Sébastian Schiesser and Caroline Traube. On making and playing an electronically-augmented saxophone. In *Proceeding of the 2006 International Conference on New Interfaces for Musical Expression (NIME06), Paris, France*, 2006. URL `http://www.nime.org`.

Belinda Thom. Bob: an interactive improvisational music companion. *Fourth International Conference on Autonomous Agents*, 2000.

Jenifer Tidwell. *Designing Interfaces : Patterns for Effective Interaction Design*. O'Reilly Media, Inc, 2005. URL `http://designinginterfaces.com/`.

Jenifer Tidwell. Common ground, May 1999. URL `http://www.mit.edu/~jtidwell/common_ground.html`.

Joseph Smits van Waesberghe. *Musikerziehung – Lehre und Theorie der Musik im Mittelalter*, volume III of *Musikgeschichte in Bildern*. Heinrich Besseler und Max Schneider, 1969.

Index

affordances, 30
anchored learning, 30

Band-out-of-the-Box, 13
barrel organ, 44, 58, 87
behaviorism, 26, 28, 30

cognitive apprenticeship, 30
cognitive approaches, 28
cognitive walkthrough, 89
constructivism, 26, 30
Core Audio, 49
criticism, 29

decoration, 83
design patterns, 3, 22
DJ, 65
dog string, 33, 50
drone string, 32, 36, 39, 43, 50
drum trigger, 81

exhibit space, 96

Fechner, 66
feedback, 29
frame drum, 6, 31, 94
future work, 98–99

glissando, 86, 87
goals, 83, 85

harp, 6, 31, 96
HCI, 8, 24, 89
horizontal prototype, 48
hurdy gurdy, 6, 31, 35, 49, 60, 73, 94, 98

interview, 85
invisible hardware, 26, 83
iterative design, 8

jazz, 13

laser harp, 18
latency, 64, 65, 67, 68, 71
light harp, 18

Max/MSP, 3, 35, 37, 49
mechanical violin, 33
mechanics, 32
medieval music, 31
melody string, 32
metric, 37
MIDI, 35, 36, 73
MIDI Keys, 37
MIDI Monitor, 37

note message, 38

patch, 35, 37
pattern language, 22, 23
praise, 29
prototype, 35, 47, 48
psychophysics, 66

qualitative, 84
quantitative, 84
quantization, 65
QuickTime, 49, 51

Regensburg, 2, 3, 47, 50, 55, 58, 60, 74, 90
Regensburg Experience, 90
replica, 5, 49, 50, 55
responsiveness, 89
reward samples, 54
rising edge trigger, 81

saxophone, 35
scale, 6, 38
sensor, 55, 56, 74, 81, 89
SimpleSynth, 37
situated learning, 26, 30
Skinner, 26
strings, 32
sub-patch, 37
synthesizer, 37

threshold, 65
trigger MIDI converter, 75, 82

University of Regensburg, 2
usage scenario, 8

velocity, 38, 81
vertical prototype, 48
video games, 17
virtual audience, 45, 76

wavetable synthesis, 15

Xcode, 49, 74

Typeset August 1, 2006