OULUN YLIOPISTO
UNIVERSITY of OULU

Development of a model for classifying software based musical instruments using the instrument Seq1 as a testbed

University of Oulu
Department of Information Processing Science
Master's Thesis
Jussi Väisänen
8.1.2016

Abstract

Software based musical instruments are new kind of musical instruments that employ digital technology in the generation of sound, user interface or both. In this work I describe a new software based musical instrument called Seq1 that was developed at the RFMedia laboratory in Ylivieska. Design choices for Seq1 came from findings in literature into similar phenomena. Research literature also unveiled the fact that although there are frameworks for describing different aspects of software based musical instruments there is no general model for classifying software based musical instruments.

There are systems for classifying musical instruments such as the most prominent of these, the Hornbostel-Sachs system which is meant as a way to classify every type of musical instrument. It was originally created in 1914 and has since then been updated several times. The last update has been in 2011 which added tens of new subcategories for electrophones (electronic instruments). The subcategories meant for classifying software based musical instruments seem to be inadequate.

This work introduces a sketch of a new model for classifying specifically software based musical instruments and introduces Seq1 as an example for the usage of the new classification system.

Keywords
Software based musical instrument, Hyperinstrument

Supervisor
Professor Kari Kuutti

Foreword

Interest in the subject of this work grew from my personal experiences with music. I am a classically trained musician, my main instrument being the cello. The cello is not the easiest instrument to play and playing it very concretely shows the meaning of a steep learning curve. Learning different instruments gave insight into different characteristics of instruments – the same characteristics found in the frameworks discussed in this work like expressivity and efficiency.

Everything changed when I was introduced into electronic music. In 1989 there was a 16-bit computer with sequencer and a simple synthesizer at my junior high school. Although a simple synthesizer, Roland D-10 had a synthesis engine that could be programmed in various ways. This taught something about the relationship of the user interface and the synthesis engine running on a digital synthesizer bringing the concept of mapping into play. In a studio containing electronic musical instruments modularity and hence levels of encapsulation exists on various levels – synthesizers are interconnected and interchangeable via MIDI protocol, audio from each and every sound source is basically interchangeable and many kinds of feedback is possible within the system, from simple delay loops via an delay effects unit and an analogue mixing desk into the parameters generating the notes themselves – software for using e.g. genetic algorithms to generate notes.

Tod Machover's Hyperinstruments played a huge part in choosing the subject of this work and for the original process of designing and implementing the new sequencer, Seq1, introduced in this work. First Hyperinstrument was a cello for the master cellist Yo Yo Ma. It was a cello augmented with digital technology. Naturally as a cellist this got me interested in the subject.

This work has taken a lot of time. It has changed various times. The original title had to do with ambient sound generators in 3d-environments and has changed at least twice. This document contains texts written through different stages of understanding the phenomena at hand but I hope it gives and coherent and interesting view on the subject.

Jussi Väisänen

Oulu, January 8, 2016

Contents

1. Introduction

The purpose of this work is to define a new classification system for software based musical instruments. It is derived from attributes and frameworks addressed in research literature, frameworks concerning the building of software based musical instruments and the outcome and experiences of the design and construction of the software based musical instrument Seq1.

This thesis has evolved from a case study of building a single software based musical instrument into creating a classification system. Correspondingly, the thesis has two major results: the developed new software based musical instrument Seq1, and a new model to classify new digital musical instruments.

In previous research into this subject it became clear that there are attributes such as low latency and high modularity that are desirable. Many were adopted into the design of Seq1. These attributes might be universal in such a way that deviation from for example low latency requirements could indicate bad implementation in most cases.

Thus the research question of this thesis is: What are the attributes needed to separate essentially different new software based musical instruments from each other, and are these attributes related to each other, or put in a different way: do these attributes form a model ?

The initial base for the work is the electronic musical instrument that was designed and built by me and Juhana Jauhiainen at the RFMedia laboratory of Oulu Southern institute, Ylivieska in 2010. Named Seq1, it is a music sequencer based on a modular design and a multi touch screen. The outcome of this design is explained in chapter 4.

While going through literature of various similar kind of phenomena it became evident that these kind of new instruments were hard to classify by any of the current classification systems in use. Therefore second part of this work concerns itself with a new kind of model for classifying modern digital instruments of this kind by their essential attributes, making it easier to compare them with each other. Hopefully this will lead to new way of analyzing general classes to identify instruments by for example their archetype. This part of the work is presented in chapter 3.

Software based musical instruments span many fields from computer human interaction, software development, and digital signal processing to music theory and the history of music in general. So before the outcome of the design and the new way of classifying instruments a historical perspective for the work is given in chapter 2.

This work discusses the historical background that precedes modern software based musical instruments and related phenomena. Prior research on the subject will be discussed in chapter 5 along with specific frameworks and attributes used to analyze software based musical instruments. Models used for classifying musical instruments will also be discussed although the most prominent – Hornbostel Sachs-system – will be the main classification system discussed in this work. Examples of software based musical instruments will be given along with their relationship with aforementioned frameworks and attributes. Problems that have arisen and can be specified by the

attributes and frameworks will be discussed. Seq1, an aleatoric music sequencer designed with the knowledge acquired by investigating prior software based musical instruments and frameworks will be presented. The case of creating a unified model for analyzing software based musical instruments will be discussed and a sketch of one will be presented.

The new classification system has been constructed from the viewpoint of the phenomena at hand, software based musical instruments, discarding the larger question of classification of general musical instruments. It will be introduced in chapter 6.

In Chapter 7 the new classification is evaluated and verified by using it to classify a number of recent software based musical instruments.

Finally the chapter 8 reflects upon the research and makes some conclusions and points out directions for further research.

Explanations for specific terms, acronyms and devices are given in the Appendix.

Methods used in the work

Two methods have been used in the work. First, the design and construction of a new artifact (Seq1) belongs to the field of constructive research, where the development of an artifact is used to generate new information. The development of Seq1 has been essential both for the formulation of the research question and then to understand the value and relative importance of different features used for classification

The second method is the analysis of the literature related both to the classification of musical instruments in general and software based instruments in particular. The evaluation of the developed model is based on analysing both Seq1 and other artefacts presented in the papers.

According to Crnkovic “Constructive research method implies building of an artifact (practical, theoretical or both) that solves a domain specific problem in order to create knowledge about how the problem can be solved (or understood, explained or modeled) in principle.” (Crnkovic, 2010, p. 4)

The design and construction of Seq1 have been based both on practical and theoretical considerations. Performing preliminary studies into the field of software based musical instruments brought forward problems that were alike with the design of several such instruments. These problems were closely related to practical considerations and addressed in the design. Understanding the problem gave the possibility to both incorporate possible solutions into the design but also to create a synthesis of the whole knowledge attained in the process of preliminary studies, design and construction of the artefact. This resulted in the model described at the end of this work.

In the context of constructive research it can be said that this work produces results both on the practical and theoretical side. Improvement on the general good practices of building software based musical instruments are given. Feasibility of the design of Seq1 was ensured after the instrument was constructed. Improvements to previous designs found in literature were made by identifying problems found in many designs and improving these aspects. Theoretical results are already discussed, practical aspects of the design were found sound after the instrument was constructed. No previous instruments like Seq1 were found in literature so this should address the aspect of novelty.

Theoretically central and valuable aspects of this work will be the aspects found in the model that enables to classify software based musical instruments. The model itself is the most condensed form of theory imaginable in this situation. Yet, improvements are made on practical aspects – such as modularity, minimizing latency, the importance of considering mapping in the design of these instruments.

The frameworks for evaluating Seq1 and previous similar software based musical instruments found in literature are multidisciplinary. These are introduced in chapter 5.2.

The most prominent classification system for musical instruments in the western world is Hornbostel–Sachs which was originally published in 1914, then revised several times (von Hornbostel & Sachs, 1961) and which has been revised by MIMO (Musical Instrument Museums Online) in 2011. There are older classification systems found in different cultures, some are based for example on the material of the instrument. These are not relevant in this study.

2. Prior research

There is general tendency to think that the innovation of musical instruments has somehow halted in the form of the orchestra of the romantic period of music with its large sections of classical musical instruments. (Otonkoski & Barrière, 1991, p. 164) In reality the invention of new musical instruments has never ceased. Throughout the history instrument builders have embraced the latest technology, whether it was the use of a new material, new molding technique or a new kind of an electronic oscillator that could be used in the audible range to make novel sounds.

The evolution of musical instruments has seen many failed instruments, merely side notes in museum but some have achieved prominence such as the electric guitar.

Composers have always sought wider tonal ranges, new kinds of sounds and new nuances for their music, a historical process that seems to have but accelerated when electronics and digital technology were brought into the picture. There has also been cross-pollination between different fields of inquiry. For example, Iannis Xenakis' compositional style based on probability fields has its roots in his work as an architect - after digital computers got fast enough his probability fields could first be practically computed with computers in no real time. Later this method evolved into a new sound synthesis method called granular synthesis computed in real time – even faster digital computer made this possible. (T. B. Holmes & Holmes, 2002, p. 101).

The rise in the processing power of computers in the last decades has democratized the art and science of creating new musical instruments. Instruments that required dedicated hardware to function two decades ago can nowadays be simulated virtually in real time with an off-the-shelf personal computer. (Schiettecatte & Vanderdonckt, 2008, p. 3)

In the last fifteen years or so plenty of commercial products have been introduced that intend to replicate the user interfaces and sounds of classic electronic instruments. There are also plenty of products that do not try to imitate older musical instruments but try to accomplish something totally new. Some of these are forgotten but some survive and are developed further. There also exist nowadays development kits and advanced software tools for the sole purpose of creating new software based musical instruments. This could be seen as the evolution of musical instruments accelerating.

The rapid development of new kinds of musical instrument poses a problem in the way they have been classified in the past. The most prominent classification system called Hornbostel-Sachs has incorporated into itself electronic musical instruments already in the 1940 and has been updated since but the up to date-version is insufficient in explaining these new phenomena, sufficient in explaining some of the physical characteristics of new kinds of software based musical instruments but failing to explain their working characteristics.

With new kinds of user interfaces new kinds of preliminary ways for evaluating software based musical instruments have been devised. There are frameworks referring to more abstract concepts such as expressivity and learning curve of collaborative musical instruments or the processes working inside algorithmic music systems. Moreover, there are simple attributes that can be used to describe single software based

musical instruments such as appearance, mapping, latency or learning curve. Although the need for a unifying model to bring these all these attributes and more abstract concepts together there is yet none.

This work discusses the historical background that precedes modern software based musical instruments and related phenomena. Prior research on the subject will be discussed along with specific frameworks and attributes used to analyze software based musical instruments. Models used for classifying musical instruments will also be discussed although the most prominent – Hornbostel Sachs-system – will be the main classification system discussed in this work. Examples of software based musical instruments will be given along with their relationship with aforementioned frameworks and attributes. Problems that have arisen and can be specified by the attributes and frameworks will be discussed. Seq1, an aleatoric music sequencer designed with the knowledge acquired by investigating prior software based musical instruments and frameworks will be presented. The case of creating a unified model for analyzing software based musical instruments will be discussed and a sketch of one will be presented.

History of software based musical instruments is closely tied to the history of electronic music in general.

Composers such as John Cage (September 5, 1912 – August 12, 1992) (T. Holmes, 1985a, p. 377) and Edgar Varèse (December 22, 1883 – November 6, 1965) lived at the threshold of the electronic revolution in music.

Varèse's Poème électronique was a turning point in the art of electronic music. It was a tape composition ordered for World's Fair in Brussels and was the first time electronic music came outside academia for larger audiences. It was partly a showcase for Philips to show off their engineering capabilities. From April to October 1958 estimated two million people attended the piece. Suddenly electronic music studios started to sprung up over the world (T. Holmes, 1985, p. 4).

Edgar Varèse had tried as early as 1930's to finance a electronic sound studio. Finally, when he was 74 years old in the fifties the technology caught up and made the dream possibility. (T. Holmes, 1985, p. 5) John Cage had been experimenting with turntables as early as the 30's (Prendergast, 2000, p. 45)

Point being, composers are eager to try new technology. Two of the main themes in Thom Holmes' book "Electronic and experimental music" are that "the marriage of technology and music is inescapable but sometimes imperfect, like any civil union". Other one is that the development of new technologies continually and sometime thwarts the creation of new music.

Lauri Otonkoski claims in "Klang - uusin musiikki" that "Composers, musicians, technicians, engineers and scientists have always tried to use latest of knowledge and know-how to apply them into musical objectives… So music has rarely been lacking behind latest technology". (Otonkoski & Barrière, 1991, p. 164)

The history of music has seen music periodically reinventing itself, time after time. New schools of thought and aesthetic have emerged to counter the old and stale with some of their aspects being assimilated to the more mainstream musical culture at a later stage.

In the history of western music medieval music gave first way to renaissance music which gave way to first baroque, then classical and romantic periods in music. Then avant-garde ideas started to appear and electronic musical instruments started to take hold. (Geiringer, 1978)

Music reinventing itself holds true also in the domain of electronic music. In the realm of electronic music new technological inventions such as the tape recorder (Geiringer, 1978, p. 210) and synthesizer have been increasing the rate of change, new inventions being essential in creating whole genres of music. For example, Musique Concréte in the 50's France was mainly based on splicing magnetic tape on tape recorders. (T. B. Holmes & Holmes, 2002b, p. 45) Still many musicians recognize that electronic music started only after the synthesizer was invented and inventions before that were mere preliminary stages in this development. (Geiringer, 1978, p. 271)

The history of electronic musical instruments is long. Divis Prokop might be called the first pioneer on the field but his Denis d'Or from circa 1730 utilized electricity only as a gimmick to give the unsuspecting player an electric shock. Jean-Baptiste Laborde's Clavecin électrique utilized electricity for sound generation for the first time (Otonkoski & Barrière, 1991, p. 165)

A notable early electronic musical instrument was Elisha Gray's Musical Telegraph which employed a two octave piano like keyboard and could send pitched notes over telegraph wires, even polyphonically (several notes at a time). Although the inventor demonstrated the device over a 200 mile distance he lost interest in its musical applications and concentrated on the possibility of sending several messages at the same time, predating communication multiplexers. (T. B. Holmes & Holmes, 2002, p. 6) The Musical Telegraph was way ahead of its time and is mostly forgotten.

In the turn of the 20th century before Lee De Forest invented electronic amplifying vacuum tubes that made easy audio amplification possible large machinery had to be utilized in order to generate higher sound volume.

A unique musical instrument called Telharmonium was designed by inventor named Thaddeus Cahill in 1897. Several versions were built. Each was huge, employing large carved tone wheels in contact with metal brushes, each part of a sound generating electrical circuit that required lots power to work. Rotating the wheels made signals capable of being fed to loudspeakers or transmitted to multitude of people via a city's telephone network. The principle was that of additive synthesis, playing several tone wheels at the same time added harmonics to the sound thus making possible to make at that time unique sounds. The principles were adopted later by electronic organ makers such as Hammond. Unfortunately the telharmonic music wasn't good business opportunity and little has been heard about the instrument since. (T. B. Holmes & Holmes, 2002a, p. 12) Given, the vision to transmit music over a telephone network to a large audience was greatly ahead of its time. To get the regular audience interested in the Telharmonium familiar musical compositions were adapted to be played on the instrument although there seems to be no mention of music specifically composed for the device. Tone wheel organs were reintroduced to the world forty years later.

Lee De Forest made a big contribution to electronic music - not to mention the whole field of electronics - when inventing the thermionic amplifying vacuum tube called Audion in 1907. The vacuum tube led directly to development of new musical instruments. De Forest adapted the concept of heterodyning (two supersonic radio frequencies mixed together to create a signal in lower audio range) for the Audion Piano

he designed in 1915. It is an interesting fact that the inventor of the amplifying vacuum tube himself made a musical instrument out of his revolutionary invention.

Most important electronic musical instrument using the heterodyning principle came from Russia. The instrument, Theremin, was built around 1920 and it was the first gesture controlled electronic musical instrument. (T. B. Holmes & Holmes, 2002, p. 19) The Theremin was played by waving hands in the vicinity of the instrument's antennas. Sound of the Theremin can be heard on many old movie soundtracks. Custom music was composed entirely for Theremin because of its unique tonal range and capabilities. It is one of the musical instruments that have survived evolution.

In the 60's Robert "Bob" Moog started his synthesizer business by building Theremin kits. (T. B. Holmes & Holmes, 2002a, p. 209) The kits are still as of 2016 available for anyone to buy.

Robert Moog was an engineer interested in the generation of sound. He was probably the most prominent character to create successful modular analog synthesizers. (T. B. Holmes & Holmes, 2002a, p. 209)

Modular synthesizers were made possible in the 60's when solid state technology made it possible to create voltage controlled modules. The idea is to have a set of audio generating and sculpting modules that are controlled by a defined voltage range. Modules were usually oscillators, low- or high pass filters, amplifiers, envelope generators and for example low frequency oscillators. By combining these modules into a keyboard capable of running control voltage signal into them the sonic range of a synthesizer could be extended almost without limit. (T. B. Holmes & Holmes, 2002b, p. 189) With voltage controlled modules paradigm was to have two main signals:

- the main audio signal going from oscillator through various sound sculpting modules to the output of the synthesizer
- the control signals that would control the voice creating and sculpting modules

This is important because the same model is still used with software based musical instruments, generally a differentiation between control and main audio signal is made although these can be combined to create various effects.

Tape and wire recorders were the first instrument that allowed musicians to precisely cut pieces of sound out of tape and rearrange them according to plan, a revolution happening circa 1940-1950. There was a rivalry between the 50's French magnetic tape cutting scene and the magnetic tape cutting scene in Germany (T. B. Holmes & Holmes, 2002a, p. 56)

From the German scene came Karlheinz Stockhausen whose early works Studien I and Studien II discovered many techniques that are nowadays standard procedures in the creation of sound and music. For example Stockhausen used for the first time in tape music additive synthesis of sound with sine waves. (T. Holmes, 2008, p. 65) Creating music by just recording simple sine wave oscillators and splicing and rearranging stretches of magnetic tape represented a huge undertaking. Modern technology and nonlinear digital editing has made these kind of operations incalculably faster although this fact says nothing about the quality of the compositions themselves as music.

This is another aspect in the evolution of electronic music: as the pioneering technology and techniques of manipulation of sound become easier and cheaper, they will become more common and used in mainstream music. For example, pioneers of tape music such as Edgar Varèse introduced surround sound in sound installations in the fifties – rock bands of the seventies such as Pink Floyd found the same techniques again. (Prendergast, 2000, p. 36)

The pioneers like Raymond Scott (1908 – 1994) (T. B. Holmes & Holmes, 2002a, p. 161), Vladimir Ussachevsky (1911 – 1990) (T. B. Holmes & Holmes, 2002a, p. 155) or Harry F. Olson (1901 – 1982) (T. B. Holmes & Holmes, 2002a, p. 143) were composers that were technologically knowledgeable enough about the nature of sound and electronics to build their own sound creating or altering devices. Some of these pioneers held degrees in engineering or physics.

History of computers, programming languages and information processing science itself cannot be written without the history of electronic music. Claude E. Shannon (1916-2001) wrote in his paper A Mathematical Theory of Communication that ”any stochastic process which produces a discrete sequence of symbols chosen from a finite set may be considered a discrete source” He gave examples of such signals; ”natural written languages, continuous information sources that have been rendered discrete by some quantization process – for example the quantized speech from a PCM transmitter, or a quantized television signal; Mathematical cases where we merely define abstractly a stochastic process which generates a sequence of symbols” (T. B. Holmes & Holmes, 2002a, p. 252) This definition includes sound and music. As the technology progressed first the notes of a composition were generated, later whole data streams needed to play natural sounds out of a human audible range digital-to-analog converter. Wooller et al. added ways to classify these stochastic processes in musical context (Wooller, Brown, Miranda, Diederich, & Berry, 2005a)

ILLIAC I (Illinois Automatic Computer) from 1952 at the University of Illinois was the first computer built and owned entirely by an educational institution in the United States. ILLIAC I was used for groundbreaking work in demonstrating how computers could be used to produce novel musical structures.

Lejaren Hiller and Leonard Isaacson thought that music was one form of information that could be managed by a computer. Hiller and Isaacson created a computer program capable of generating data that could be transformed into parameters of a musical score. By picking suitable results the data transformed into data used for composing the piece Illiac Suite for String Quartet in 1957. (T. B. Holmes & Holmes, 2002a, p. 263)

RCA Mark II Music Synthesizer from Columbia-Princeton Electronic Music Centre in the year 1957 was a computer dedicated to render music to analog magnetic tape based on a program inserted in the machine via perforated paper tape. It cost $500,000 dollars which compares nowadays to inflation adjusted amount of about 4,2 million dollars. (T. B. Holmes & Holmes, 2002a, p. 147)

When computers became available Iannis Xenakis used them to calculate complex parameters and probabilities for individual notes in his musical compositions rather than render PCM data to analog tape in the case of RCA Mark II.

Computers helped Xenakis' compositional process because before computers he had had to calculate data for his compositions by more simple means. For example Metastasis composed in 1964 was calculated by hand. It involved 61 musicians in a transforming unison. Fibonacci series and 12-tone series system (used by serialist composers) were utilized.

Xenakis had conceived the musical style based on probability theory and associated mathematical processes in the 1950's. This involved the generation of large masses of sound parameters that were very tedious to calculate by hand. Xenakis gained access in Paris to an IBM 7090 computer and started computing parameters for his compositions (T. B. Holmes & Holmes, 2002a, p. 155)

Max Mathews discovered how to use digital-to-analog converter with a computer to generate audible sound in 1957. (T. B. Holmes & Holmes, 2002a, p. 101)

We can see three ways in which the electronic music has evolved in the twentieth century:

1. New technological innovations have allowed composers to create new types of sound and music (e.g. musique concréte and the invention of rearranging pieces of magnetic tape)

2. Needs of the composer have led engineers to develop new technological innovations (e.g. voltage controlled sound modules specified by Robert Moog (T. B. Holmes & Holmes, 2002a, p. 188), computers dedicated solely for music (e.g. RCA Mark II)

3. Cross-pollination between different fields has brought new inventions for the use of composers. e.g. Xenakis' composing style based on probability fields coming from architecture, development of faster and faster computers, DA-converters (T. B. Holmes & Holmes, 2002a, p. 101) and such.

3. Examples of software based musical instruments

These are examples of software based musical instruments that have been found in research literature before designing and actualizing Seq1. There are many instruments using multitouch tables and tangible interfaces, many using fiducial tracking technology and objects such as pucks over the table.

In this chapter I also summarize different software based musical instruments described in this work by the attributes found in the previous chapters about frameworks and theory. Every aspect about the software based musical instruments presented is not necessarily directly written down in the original research but implied by the design of the instrument. Can the attributes be addressed one by one for each of the software based musical instruments presented? Seq1 is addressed here because it is a critical point of this work but is described in the next chapter.

3.1.1 Audiocubes

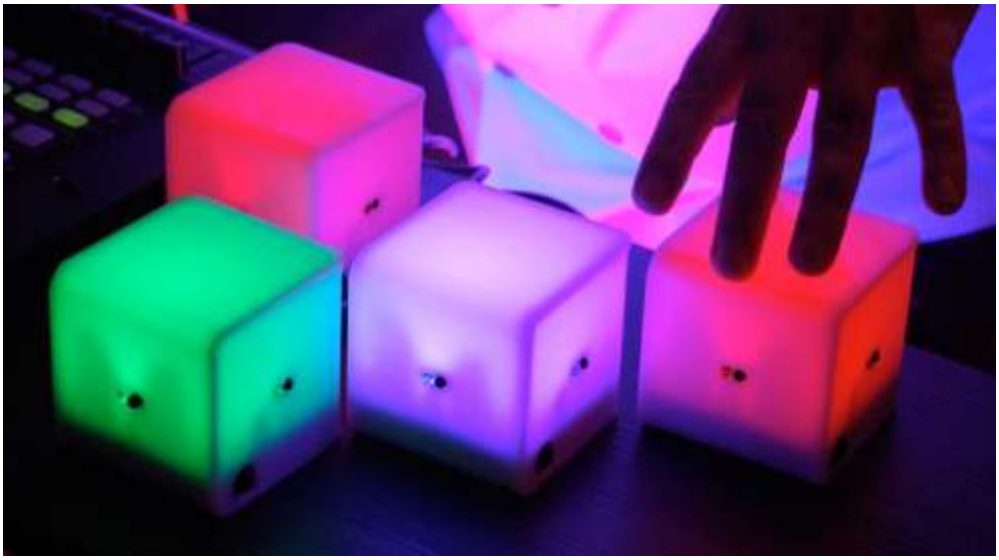

Figure 1. 2011 Close-up of 4 Audiocubes (Schiettecatte, 2011)

Audicubes are physical cubes containing a digital signal processor, electronics to make it work and sensors. They are self-containing, have a rechargeable battery and can be placed on any table surface near each other to create a signal processing network. Each cube contains its own software that can be updated via a cable. Cubes identify themselves in the network by emitting certain type of a sound. (Schiettecatte & Vanderdonckt, 2008b, p. 2). There are several different kinds of cubes that are identified by the user by their color. For example granular synthesis algorithms are implemented in a yellow cube. (Schiettecatte & Vanderdonckt, 2008b, p. 5) This is concept is very much like the reacTable but doesn't require a specialized backlit screen. Downside is that for reacTable you can print with a printer your own physical audio units if you wish – audiocubes are more complicated, they require more hardware: their own power supply, electronics, sensors and digital signal processor. Also for using audio to

communicate instead of infrared light the latency of the Audiocubes is inherently higher. Latency is dependent also on the network complexity; it may be vary between 10 milliseconds to 100 milliseconds.

Relation to frameworks and attributes

By appearance, Audiocubes can be classified as a tangible user interface. By mapping the system can be described as many-to-many as a single Audiocube can be placed towards another one in any way. As mentioned, latency is dependent on the total latency of all the Audiocubes in the current network and is above desired amount for real time performance, 10 to 100 milliseconds. Test subjects generally needed guidance on the principles of the cubes if given an opportunity to test them freely. So, learning curve is steep at the beginning. Authors suggest learning process for the usage of the system and its parameter control similar to that of an electric guitar. (Schiettecatte & Vanderdonckt, 2008a, p. 8) So, a similar learning curve as in the electric guitar is probable. By adding nodes to the system it becomes quickly more complex and expressive. At the same time the efficiency gets better i.e. less control parameter changes need to be performed to cause a multitude of changes to the output of the system. Expressivity increases as more nodes are added to the network – the system becomes more complex but unfortunately latency also gets higher.

3.1.2 Audiopad: A Tag-based Interface for Musical Performance

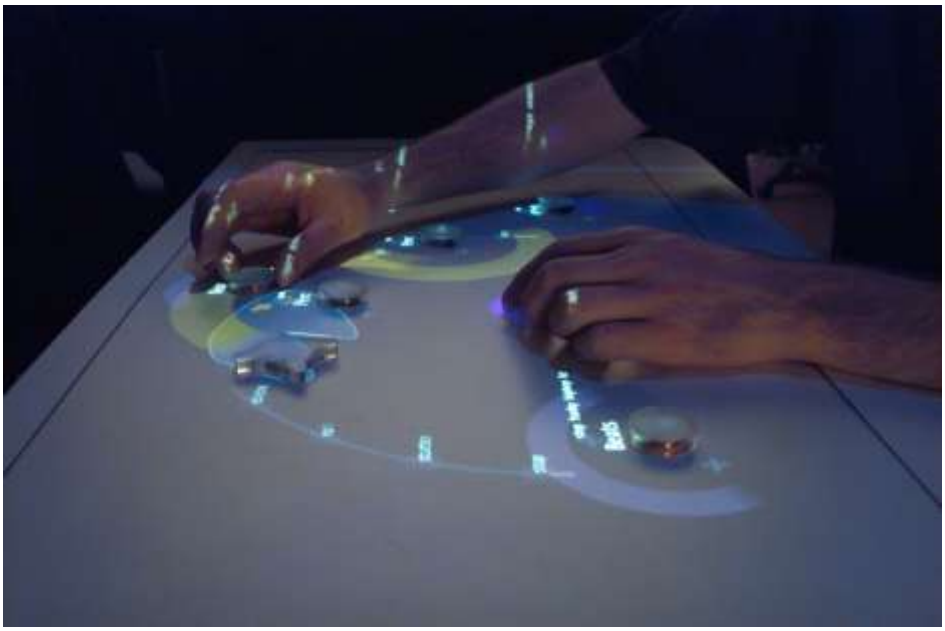

Figure 2. Audiopad in action. Press photo showing the instrument in use. (Patten, 2016)

The idea behind the Audiopad is to bring the expressive power of analogue synthesizers back to musicians. As personal computers in general got faster and laptops in particular performing musicians started to utilize them in their performances instead of synthesizers on stage. This created a drift between the audience and performer as the audience didn't see what was happening on the laptop screen. Laptops also reduced the expressivity, nuances and improvisational capabilities of performances. (Patten, Recht, & Ishii, 2002a, p. 1) Regular midi controllers are dismissed as non-expressive, they are basically bunch of potentiometers glued to a box. In a performance situation it is not necessarily easy to remember to what every potentiometer is assigned to.

Audiopad uses electromagnetically tracked pucks as input devices on a table. User interface is projected from above. Each puck includes an RF tag called LC tag, consisting of a coil and a capacitor which enables the Audiopad to track the pucks over the surface. There are precedents to using RF tags for such use as The Marimba Lumina (Patten et al., 2002a, p. 3). As with the Seq1 Audiopad chose to use MIDI as interface because of its popularity.

Relation to frameworks and attributes

The user interface consists of a horizontal table with an image projected from above. The user interface is controlled by pucks placed on it – pucks may have individual user interfaces projected beside them so appearance of the interface is in its way multimodal. As this instrument was developed in response to the poor expressivity of simple laptop and midi controller-type setup mapping capabilities of the instrument have been given thought – mapping can be seen as many-to-many and is easy to change on demand. Low latency is claimed although the actual numbers for the latency are omitted. (Patten, Recht, & Ishii, 2002b, p. 2) As the system is used with a back end software (Ableton Live), the latency is at least that of the back end software (usually 5 milliseconds) added to the whole latency of the system. In the evaluation period users claimed the system users found the system very satisfying to use. (Patten et al., 2002b, p. 5) Deeper insight into the learning curve of the system was not given. Efficiency is hard to discern for the instrument as the data going in and out of the system is basically changed by the user but the amount of data remains the same. With more complicated mappings and a performance requiring more real time changes the efficiency naturally drops. Resolution of position tracking within the system of a single puck is accurate within 4 mm. (Patten et al., 2002b, p. 2) As the system is not a musical instrument per se but a system for modifying incoming soundtracks it is hard to describe expressivity. The system is aimed at controlling performances; several tracks played in real time. Contextual breadth is limited to shaping incoming notes and parameters of the synthesis engine, not per se changing the composition. The tracking system and user interface send MIDI signals to back end which uses Ableton Live software as its back end. Because the use of MIDI, the back end could be connected to any other sequencer, DAW or a hardware synthesizer. Modularity and thus levels of encapsulation exist at least on the level that the main software and back end are compartmentalized. (Patten et al., 2002b, p. 3) The system gives graphical feedback about the synthesis process as well as audio feedback. Passive haptic feedback is also mentioned as the pucks are physical objects on a table. (Patten et al., 2002b, p. 3)

3.1.3 Drawsound

Figure 3. Whole performance created in Drawsound. (Jo, 2008a, p. 62)

Drawsound utilizes multi-touch table called Diamondtouch (Jo, 2008b, p. 1). Above the surface of the multi-touch table is placed a paper. Idea is to draw lines and pictures on the paper and transform this information into sound. It is intended for live musical performance. Three different kinds of conductive pens are used for different style of playing. One of the pens can be used for continuous drone sounds and two emulate normal brush painting and a Japanese style pen. The Diamondtouch works by detecting changes in electrical field of the table. Cycling 78 Max/MSP is used as the main backend software. Using patches created with MaxMSP Drawsound is able to record and playback both the visual and auditory representation of the performance (Jo, 2008b, p. 2).

Relation to frameworks and attributes

The system has a dedicated design environment to enable easier creation of mappings; mappings are not performed real time. Mappings from the Diamondtouch to the synthesis back end can be seen as many-to-many (Jo, 2008c, p. 60) although for performances some regular mappings have emerged (Jo, 2008c, p. 61) Moving the pen is scanned at an interval of 33 milliseconds so this should be the lower bound of the latency of the system. (Jo, 2008c, p. 60) Added latency comes from the synthesis back end which is with regular pro audio interface drivers in the five millisecond range with the use of Max/MSP. The Drawsound is custom made by The SINE WAVE QUARTET (SWQ) for the quartet itself so it has only the player of the Drawsound as an example; learning curve is not discussed further. (Jo, 2008c, p. 1) Efficiency of the system stays the same by any mappings used; the user draws a line and a sine wave of variable frequency, panning and amplitude comes out. As the instrument is highly dependent of the metaphor of drawing and drawing is part of the sound generation itself the authors liken the expressivity of the instrument with that of drawing with a pen itself. (Jo, 2008c, p. 4) Contextual breadth can be increased by using more and more creative mappings between the front end of the system and the sound engine (Max/MSP).

Changing from a simple sine wave to a more complex waveform or a sample based playback device could give numerous parameters that the sound engine could translate into sound and thus increase the contextual breadth. As with several other multitouch table-based instruments there is encapsulation in the sense of modularity, front end software talks to the back end via OSC. (Jo, 2008c, p. 60) Both the front end and back end are standalone modules and are in essence interchangeable. The system gives both auditory feedback and visual feedback as a real time drawing is created on a sheet of paper in front of the audience.

3.1.4 HERON: A Zournas Digital Virtual Musical Instrument

Figure 4. Vizualisation of the Virtual Zournas (Tzevelekos, Georgaki, & Kouroupetroglou, 2008a, p. 356)

HERON is a VMI (Virtual Musical Instrument) which authors define as a musical instrument that doesn't belong to physical domain but a virtual one – although by their definition interaction with a VMI usually happens in the physical domain while the sound production happens in the digital domain (Tzevelekos, Georgaki, & Kouroupetroglou, 2008b, p. 1)

HERON is based on physical modeling and is a model of a traditional Greek double-reed woodwind instrument, the Zournas (Tzevelekos et al., 2008b, p. 2). Physical modeling is a way to mathematically model acoustics of a real instruments and has been used also on modern commercial synthesizers such as Korg Prophecy and Yamaha VL1. (Böttcher, Gelineck, & Serafin, 2007, p. 32) HERON can be played only via a MIDI file or an external midi device and accepts only a limited set of parameters (pitch, duration, velocity) (Tzevelekos et al., 2008b, p. 6) which means the mapping of the

instrument is limited. One could argue that this little control over parameters of a instrument are inadequate. Aftertouch, polyphonic aftertouch or custom control parameters could be utilized to bring more control to the instrument. The authors claim this fact too. (Tzevelekos, Georgaki, & Kouroupetroglou, 2008c, p. 357)

Relation to frameworks and attributes

The appearance of the instrument is a 2D user interface on a computer screen – it shows an archetypical model of an actual physical instrument called Zournas. Mapping has not been given a lot thought, there exists no possibility of changing mapping of parameters. The HERON can be played only via a MIDI file or an external midi device and accepts only a limited set of MIDI parameters (pitch, duration, velocity) (Tzevelekos et al., 2008c, p. 357). Inside the software itself such characteristics of the physical model can be adjusted as bore diameter or bore length. The latency of the system is not mentioned. (Tzevelekos et al., 2008c, p. 356) As the Zournas currently resides behind a computer screen without dedicated physical controller and is a monophonic instrument with only three control parameters, little can be said about its learning curve. Not much can be said about the efficiency of the system as it outputs monophonic sounds with three parameters. Also, at the current state of affairs and lack of dedicated physical Zournas controller expressivity cannot be explained. For every single note the synthesis engine receives three parameters in a performance situation. In addition physical characteristics of the virtual Zournas can adjusted but not in real time – contextual breadth is limited at the time of the performance by these three inputs. There is virtually no levels of encapsulation as the The Zournas itself is a single piece of software running on a computer. Feedback of the system comes in two forms: the sound of the system played according to the parameters of the physical model and the three MIDI parameters. Visual feedback is given in the shape of the Zournas as different physical characteristics of the instrument are rendered on the computers screen.

3.1.5 The Hyperbow

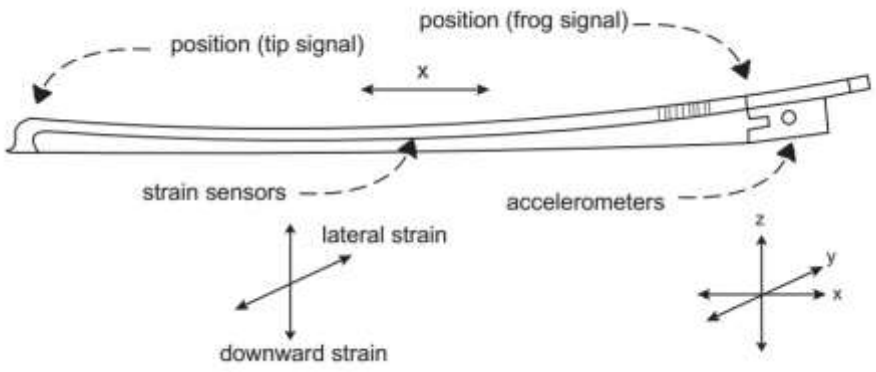

Figure 5. The various ways The Hyperbow can sense movement (Diana Young, 2002a, p. 5)

Hyperbow consists of sensors installed into a violin and bow; most of them are placed in a carbon fiber bow used to play the instrument. The instrument's main aim is to bring new nuances into violin playing for more experienced players. It is based on previous research by Tod Machover on Hypercello for Yo Yo Ma.

Foil strain gauges are used to measure downward and lateral strain of the bow stick. (Diana Young, 2002b, p. 1). The instrument also uses sensors capable of sensing

electromagnetic fields relative to the electrode antenna mounted behind the bridge of a Jensen electric violin (Diana Young, 2002b, p. 2). Subsystems relay information to each other. Latency of the system is not clearly mentioned but it has to be near real time as the system is developed for a experienced musician playing the violin in real time. Learning curve of the Hyperbow is not easiest thing to define; it is clearly meant for players who already have mastered their instruments and meant for giving more control over different nuances of the sound.

Relation to frameworks and attributes

Mapping of the Hyperbow can be defined as many-to-many. The Hyperbow mimics the bow of an original classical violin but adds sensors that can be mapped in different configurations to the performance. Latency of the system is not clearly mentioned but it has to be near real time as the system is developed for an experienced musician playing the violin in real time. Hyperbow is certainly not designed for beginners; it is clearly meant for players who already have mastered their instruments and meant for giving more control over different nuances of the sound. The learning curve thus would start from the level of advanced players and then it by making the bow technique probably more complicated and thus the learning curve steeper. As the Hyperbow increases the amount of parameters from a classical violin bow it can be assumed that more parameters need to be given though of when playing, thus increasing expressivity while decreasing theoretical efficiency. Expressivity is increased by giving the violin player more parameters to control. The design specifically means to enhance the abilities of an advanced player. Contextual breadth is increased if compared against a regular violin bow because of the increased amount of parameters to control. Only mention of the modularity or encapsulation of software translating the bow movements into usable data stream for musical applications is that the bow position is determined from the data in a computer connected to the output of a remotely mounted circuit board. (D. Young, 2002, p. 2) The system gives aural feedback and passive haptic feedback as the bow itself is a physical controller of the system.

3.1.6 The Hyper-Flute

Figure 6. Hyperflute played by Cléo Palacio-Quintin. Photograph by Carl Valiquet (Palacio-Quintin, 2008, p. 293)

The Hyper-Flute is a normal concert flute extended with electronics to enhance its performance capabilities with digital processing on a computer. Good mapping capabilities have been created. Purpose of the instrument is to enhance advanced flautist's expressivity with various sensors that can be mapped meaningfully to modify sound. All data from the sensors is converted into standard MIDI Continous Control messages which give a resolution of 128 steps to the messages (Palacio-Quintin, 2003, p. 1). In the context of classical musical instruments this might be too little resolution. MIDI protocol defines non registered parameter numbers (NRPNs) which can be used to raise the resolution to for example 16384.

The sensor data of the Hyper-Flute interacts with Max-MSP programming environment where the author has created different patches to try different kinds of mappings for the instrument. It is possible to create almost infinite amount of different mappings with the instrument (Palacio-Quintin, 2003, p. 2)

Relation to frameworks and attributes

Good mapping capabilities have been a design goal as the purpose of the instrument is to augment and enhance the performance capabilities of an advanced flute player. Latency of the system is near real-time as the architecture of the system depends on signal processing performed by Max/MSP. The original flute sound goes through signal processing, varied by different kinds of Max/MSP patches (programs for allowing different kinds of mappings). As with the Hyperbow the Hyperflute is meant as an aid for experienced instrument players to extend their instrument's expressivity. The learning curve is affected at the advanced stage and made a little more difficult, thus gaining more expressivity for the instrument. Here again, as with the Hyperbow, efficiency is sacrificed by bringing more parameters for the player to control. Efficiency

can be seen and lowering as the parameter numbers increase. Expressivity for an advanced flautist is increased because of increased number of parameters that can be controlled in real time. Contextual breadth is increased when compared to a regular concert flute because of increased number of parameters. Moreover, because programming Max/MSP patches is integral part of the system the amount of parameters controlled in real time can be increased even more than with a many-to-many mapping; mappings that could alter dynamically by the position the performance could be easily created. Main components of the system are the augmented electronics of the flute and the computer running signal processing software so there are two main components that could theoretically be interchangeable so there is a degree of modularity involved. The system gives auditory feedback and passive haptic feedback as the augmented concert flute is the controller device for the system.

3.1.7 The reacTable

Figure 7. A Reactable at the Altman Center in 2007 (Williams, 2007)

The reacTable is one of the most prominent performance oriented instrument in recent years. Its development lead it to become a commercial product. It is based on a previous design called FMOL that was a software synthesizer without a multitouch table or pucks like the reacTable (Sergi Jordà, Geiger, Alonso, & Kaltenbrunner, 2007, p. 3), (S. Jordà, 2003)

The reacTable consists of a round multitouch table and pucks placed on top of the instrument so the table has no lead position for any player. Internally it uses a radial coordinate system and radial symmetry instead of the usual cartesian two dimensional system. The pucks can represent for example modules of a modular synthesizers such as oscillator, filters or envelope generators. It can be thought of as a sophisticated, multiuser modular synthesizer. The system tracks pucks on the table with infrared light to mitigate occlusion and high latency of similar kind of systems based on ultrasound or electromagnetic field sensing. (Sergi Jordà, Geiger, Alonso, & Kaltenbrunner, 2007a, p. 4).

Relation to frameworks and attributes

The appearance of The reacTable is a round multitouch table with controller pucks on top, graphical representation of the synthesis process via a modular synthesizer metaphor projected on the table from below. The reacTable uses pucks that can be assigned to represent different kind of modules of a modular synthesizer. As the possible configurations of a modular synthesizer rise almost geometrically when new modules are added the mapping capabilities are impressive. Latency is rather low. Audio synthesis of the system works in near real-time and the system keeps tracking the pucks at the rate of 60 times per second. The learning curve is optimal in the sense that a beginner can use only a few pucks that represent the simplest modules and then start to build larger configurations. Efficiency varies in the sense that with a modular synthesizer one can create a patch that changes endlessly via random parameters or via changing just one parameter. Expressivity is dependent on the configuration of pucks created with The reacTable. Expressivity of a modular patch with dozens of modules and several people controlling them is most likely higher than that of made by an amateur. Contextual breadth as well as expressivity is dependent on the configuration of the modules interconnected with The reacTable. There are several components to The reacTable. ReacTIVision, the reacTable vision engine for tracking fiducial markers in real time, the visual synthesizer, a connection manager for sending data to different components and finally, the audio synthesizer (Sergi Jordà, Geiger, Alonso, & Kaltenbrunner, 2007b, p. 142). The system is highly modular. The reacTable gives both visual and auditory feedback. Pucks on the table give passive haptic feedback.

3.1.8 Instruments by Mäki-Patola et al.

Figure 8. A Virtual Air Guitar (Mäki-Patola, Laitinen, Kanerva, & Takala, 2004a, p. 15)

Mäki-Patola et al. created and studied four instruments of which three can be controlled with gestures in a virtual space. They are played in a Cave-like virtual space and data gloves (Mäki-Patola, Laitinen, Kanerva, & Takala, 2004b, p. 11). According to Mäki-Patola there has been relatively little research into using virtual reality user interface to control sound. Problems with virtual reality based musical instruments are the lack of tactile feedback, bad temporal and spatial resolution and latency inherent in the whole system when playing an instrument. These aspects mean that playing this kinds of virtual instruments is a rather different thing than playing classical instruments. For quick prototyping a dedicated instrument interface system was built. (Mäki-Patola et al., 2004b, p. 11).

Mäki-Patola understands the limitations of the current technology he is using to create instruments and works around them, finding new ways of playing the instruments that probably were not planned in the beginning of the research. I will address some of these problems with the instrument created based on these findings. Most of Mäki-Patola's instruments are played within a Cave-like virtual room called with shutter glasses. (Mäki-Patola et al., 2004b, p. 1)

Mäki-Patolas virtual Xylophone has a high latency from a virtual mallet being stricken to hearing the sound after 60 milliseconds. This is totally different from a normal xylophone which has a similar latency of less than 5 milliseconds (the time it takes sound to travel from the real physical mallet to player's ears) (Mäki-Patola et al., 2004b, p. 1). In a virtual Theremin based on FM synthesis the same latency doesn't affect playing that much because Theremin's sounds are sustained for a longer periods of time, not percussive such as in a xylophone. Users perceived the latency on when it became about 30 milliseconds.(Mäki-Patola et al., 2004b, p. 11)

One of Mäki-Patola's goals on the design of the instruments has been fast prototyping and a modular structure to test quickly different kinds of configurations. For example, a control parameter mapping module that is regarded very important by some (A. Hunt et al., 2000, p. 1) is implemented in Mäki-Patolas' designs. This module enables the user to create different kinds of mappings between the input parameters and the synthesis parameters of the instrument (one-to-one, one-to-many, many-to many).

The last two instruments realized by Mäki-Patola et al. in his article (Mäki-Patola, Laitinen, Kanerva, & Takala, 2004c, p. 14) are a drum membrane and a virtual electric guitar that has been shown in Finnish media. One point that became evident in the research was that facsimile copying of the original instruments to virtual reality was not purposeful but the creation of the new instrument's user interfaces should start from the constraints of the virtual reality platform itself. The new four instruments created had aspects that could not be found on the original archetypical instruments. (Mäki-Patola et al., 2004b, p. 11)

The Augmeded Djembe Drum Mäki-Patola created takes a different approach from the other three instruments; it does not work in aforementioned CAVE like system but utilizes machine vision. (Maki-Patola et al., 2006). The instrument works as follows: 1. a regular web camera is placed inside a Djembe drum. 2. From a light source above the drum, shadows of the players hands can be seen through the membrane of the drum. 3. Data is transmitted through OSC to a regular virtual software instrument, a sampler that works via ASIO drivers on Windows operating system. First PureData was considered but it was not flexible enough. (Maki-Patola et al., 2006, p. 365).

The same problem is present in The Augmented Djembe as with contraptions created in the CAVE like environment. With 30 Hertz sampling rate the web camera is able to get to a latency of 50 milliseconds between refreshing data. This made a professional percussionist rather annoyed because he has former experience with a real djembe drum that again has sound latency of about 5 milliseconds (the time it takes for sound to reach players ears). The most musically advanced test subject proposed combining The Augmented Djembe with a real one (Maki-Patola et al., 2006, p. 365) which supports Mäki-Patola's view that copying the original instrument to virtual space is not sensible but the creation process should start from the limitations of the virtual environment itself. Mäki-Patola claims that The Augmented Djembe helped inexperienced players learn to play a djembe drum with less exercise than a regular one which was a point of this experiment. (Maki-Patola et al., 2006, p. 365)

3.2 Current limiting factors in technology and design in software based musical instruments

Some limiting factors caused by current technology affect the whole experience of playing and the nature of software based musical instruments. Some of these can be avoided by careful design.

3.2.1 Latency revisited

Mäki-Patola discusses one aspect in depth: latency. The whole experience of playing Mäki-Patola's hyperinstruments is shaped by this aspect. For example, his virtual xylophone that is played in a CAVE environment has a latency of 60 milliseconds – way higher than in the traditional xylophone. This affects the whole nature of the instrument; it is not played the same way as its original archetype. One way, of course, is to explain the excess latency away and analyze the results – how did the test subjects react, in what way did the playing of the instrument differ from playing the original one? This is all important and has been done in in Mäki-Patola's work (Mäki-Patola et al., 2004b). As explained in the chapter 1.5 the ways in which authors describe the latency of their instruments is generally vague.

In the commercial side of things standards have been accomplished in order to guarantee both low and constant latency. It is only feasible but recommended to use a modern audio interface with a driver optimized for audio production such as ASIO (10.1).

3.2.2 Modularity

Using well established protocols and standards for intercommunication of different parts of a software based musical instrument gives certain advantages. As still with commercial products, MIDI (10.3) is de facto standard for transmitting low bitrate note and controller data between digital musical instruments and the software controlling them. In the process of designing The Reactable (1.5.7) a new protocol was defined called OSC (10.6). This has become the standard for transmitting data between the software running multitouch tables and their back end. It has also found its way into Plugin-instruments and for example mobile applications used as a remote controller for modern sequencers (10.7).

Unlike, for example, The Hyperbow (1.5.5.) and The Hyper-Flute (1.5.6) that use custom ways of communicationg with their software, using well known protocols as MIDI or OSC gives certain advantages. If the user interface of a software based musical instrument stays on one logical unit and note- and control data can be sent via MIDI or OSC out from the device into another unit the receiving unit is interchangeable. Without modification mappings of the system can be rearranged at the receiving end of the system without altering the user interface. Also, there is nothing to stop from switching from say, audio to video – controlling video streams instead of creating notes and music. Modularity can exist on different levels but using it at this basic stage is advantageous.

3.2.3 Non-physical?

This and other aspects like this have to be taken into account when discussing the properties of certain software based musical instruments. For example Heron Zournas (Tzevelekos et al., 2008c) which its creator defines as a VMI (Virtual Musical

Instrument) only appears as a simulation behind a computer screen. It accepts MIDI data and plays it back in non-real time. There is no way of playing it physically – expressivity and learning curve are thus hard to define but aspects like mapping still remain. Appearance can be defined but as the instrument stays behind a two dimensional display the description of the way it should be played in real life becomes rather moot. In the case of the Zournas though mapping has not been given much thought. (1.5.4)

Generally, this is a larger problem faced by any commercially available plugin instrument sold for use with a computer sequencer. Plugin instruments are in wide use; they are software designed for a single purpose, to emulate certain type of instrument inside a sequencer environment. They have their own user interface and adhere to certain standard that sequencers themselves accept (usually named VSTi, AU, DXi). As with the Heron Zournas, expressivity and learning curve of these commercial products are hard to define. These attributes also become case specific because the user may use one of the various kinds of commercial MIDI controllers to control the parameters of a plugin instruments. (Figure 2.) The Audiopad (1.5.2.) was specifically created to avoid the paradigm of performing music with just a laptop and a general MIDI controller.

Figure 9. Archetypical MIDI controller called X-Session, basically bunch of potentiometers with not much thought on expressivity. (Moore, 2012)

4. Seq1

Seq1 is a new software based musical instrument developed in co-operation by Oulu Southern institute and Centria Research and Development in 2010. The work was carried out in RFMedia research laboratory in Ylivieska, Finland. Team of the Oulu Southern institute was planning on technology demonstrations of various kinds varying from digital holography into usability. I pitched on an idea for a new kind of software based musical instrument for a demonstration as I had just done research into the subject and was confident about being able to carry out a working design. Surprisingly all the resources needed were available – time for creating specifications, programming and the hardware needed.

I designed the instrument, Juhana Jauhiainen programmed the user interface and the core system. As the goals were simple there was no formal documentation made for the software itself besides hand writte notes. In testing the whole system only one bug and one misunderstood feature was found. As the instrument was created using parts from different projects the whole system is no longer intact but the software remains.

4.1 Empirical background

This work is partly based on the design and creation of a new software-based musical instrument called Seq1. The design choices of the instrument are largely based on experiences in previous research into building such similar new instruments.

Aspects found important in research for software based musical instruments were thought carefully in this design including:

- Easy learning curve
- Modularity
- Easy capability of changing mappings of the user interface elements to the sound synthesis engine.

Attention has been paid to avoid drawbacks in previous musical instruments found in research literature.

Several factors led to the design and construction of Seq1. There was a need for a technology demonstration for the available technology at the time in RFMedia laboratory at Ylivieska. There was also a need to construct applications for a large multitouch display. At the same time I was researching software based musical instruments. It became apparent that it would be fruitful to test concepts found in this research and create a new artefact based on the findings as there were enough resources available to be allocated to the task.

One thing that became apparent throughout the process is that though there are a multitude of different kinds of software-based musical instruments introduced in research there is no definite classification system for them although it clearly seems that one is needed. (Kvifte & Jensenius, 2006)

Seq1 has already been presented in MindTrek 2011 conference. Rather than focusing solely on explaining the design it seems more useful to use the knowledge gathered from the process of creating and using it and push the envelope a bit further. Because of lack of proper classification system for software based musical instruments this work introduces a preliminary model that can be refined later. There is a need for a classification system like this (Kvifte & Jensenius, 2006, p. 1)

Seq1 is an aleatoric music sequencer aimed for soundscape creation and live performance. (Väisänen & Jauhiainen, 2010, p. 1) It intends to address problems with certain attributes of aforementioned instruments. Inspiration for the design came from such software based musical instruments as the reacTable which has been used by contemporary musical artists in their live performances. reacTable and recent musical instruments like it are based on large multitouch screens. Also they have in common "pucks" or objects that are moved across the screen in their user interface. In reacTable pucks are used to simulate components of a modular synthesizer, feeding on the archetype of early modular analog synthesizers. Patten's Audiopad uses a similar approach. Also there is an sequencer called Xenakis which uses same kind of multitouch table but Xenakis is designed for composing music with probability models. (Bischof et al., 2008, p. 123)

Seq1 consists of a 46" Full HD (1920 x 1080 pixel) multitouch screen that is placed on a horizontal position although the screen can be also positioned vertically. The structure of the system is modular and three part as follows:

1. Multitouch Cell's own software for recognizing movements on the surface of the table, sending commands to a..

2. ..custom software for the sequencer itself that sends midi messages via a virtual MIDI interface..

3. ..to a MIDI/ASIO host (Reaper) that can playback virtual sound sources (VST instruments) in near real time

Caveats presented by previous research addressed by Seq1 are:

1. Latency is minimized by using ASIO driver. The overall latency of the system is under 10 milliseconds which is enough for a sequencer of this kind (Mäki-Patola et al., 2004b)

2. Learning curve is gentle. Anyone can get sounds out of the system but when testing the system it took effort to create soundscapes – creating a certain kind of a soundscape requires analyzing by scientific or artistic means the ways of recreating the sounds and their frequencies played in the soundscape. Interactions between sounds can also be programmed but this includes programming suitable software synthesizers in the synthesis end of the system.

3. Mapping between the sequencer and its sound engine can be virtually anything – the modular structure of the system allows the sound engine to be changed or modified at will. The sound engine has to understand MIDI protocol, though.

4.1.1 Seq1 custom software

The custom software developed for the instrument runs between the Multitouch Cell software and the synthesis engine (MIDI host that receives data from the custom software.) It was programmed using C++ and Qt. The software listens to certain network port for TUIO (Tangible User Interface Objects) messages. The Multitouch Cell tracks hands touching the multitouch table by using rear diffuse infrared technology. The Multitouch Cell software translates these gestures into messages that our custom software reads. Because our custom software listens to TUIO messages the whole system is not restricted to using multitouch tables from a single vendor. TUIO is becoming de facto protocol of messaging for multitouch tables.

For sending MIDI messages from the custom software to MIDI host an open source library named MIDIIO is employed.

In the prototype of the instrument real time sound synthesis runs on the same computer that also runs at the same processes responsible for rendering the GUI and tracking gestures on the multitouch table. This was quite resource intensive for the computer that was used. Advances in processing power will negate this problem in a few years.

Driver that enables routing of MIDI data virtually inside the computer called MIDI Yoke is used to route the MIDI data from the GUI to a sequencer called Reaper.

Reaper is running various software instruments and virtual effects according to the needs of the performance played. Sound latency is minimized by using an free low latency sound driver called ASIO4ALL.

User's view of the system is based on the archetype of a game named Asteroids where objects or asteroids are seen moving on a two dimensional plane, idea is to shoot the asteroids into smaller chunks. The idea is modified to fit the idea of an aleatoric musical sequencer better. In the original game the asteroids would zoom past the screen and appear on the other side of the screen but in this version objects bounces off the screen.

There are five different colored balls that the user can drag with fingers on the multitouch display to make them move. After letting go of an object on the screen it continues on its trajectory until it hits a wall. Hitting a wall produces a MIDI "note on" command to be sent to the MIDI host playing software synthesizers. Pitch of the MIDI note on command corresponds to the height of the site of impact. Initially the whole vertical axis was divided into two octaves (24 half notes). Naturally this can be changed by changing the mapping. This could normally be accomplished easiest by changing the MIDI data by a transformative (Wooller, Brown, Miranda, Diederich, & Berry, 2005b) MIDI plugin at the MIDI host receiving the data.

Pitch of the note is calculated from how high the ball hits the wall of the screen, by vertical resolution. Each ball on the screen sends its own data on its own channel. To make things more complicated and give more expression to the system each of the five balls sends their vertical and horizontal coordinates to the MIDI host by the use of MIDI control change messages (0…127) all the time. These parameters can be used to modulate various parameters on the software synthesizers played on the MIDI host, such as amplitude of the sound or a parameter on a filter of a subtractive synthesis algorithm.

In the first working version of Seq1 gate on/off-selection was removed. This was a parameter that enabled the sending of MIDI “note on” message with or without MIDI “note off” message. Stop and delete-buttons were also removed, considered unnecessary. Random and Speed-sliders were present. Random-slider randomizes the angle after impact to the walls as needed. Speed-slider increases the speed of the balls, handy for raising the frequency of impacts to the walls and thus creating more MIDI notes.

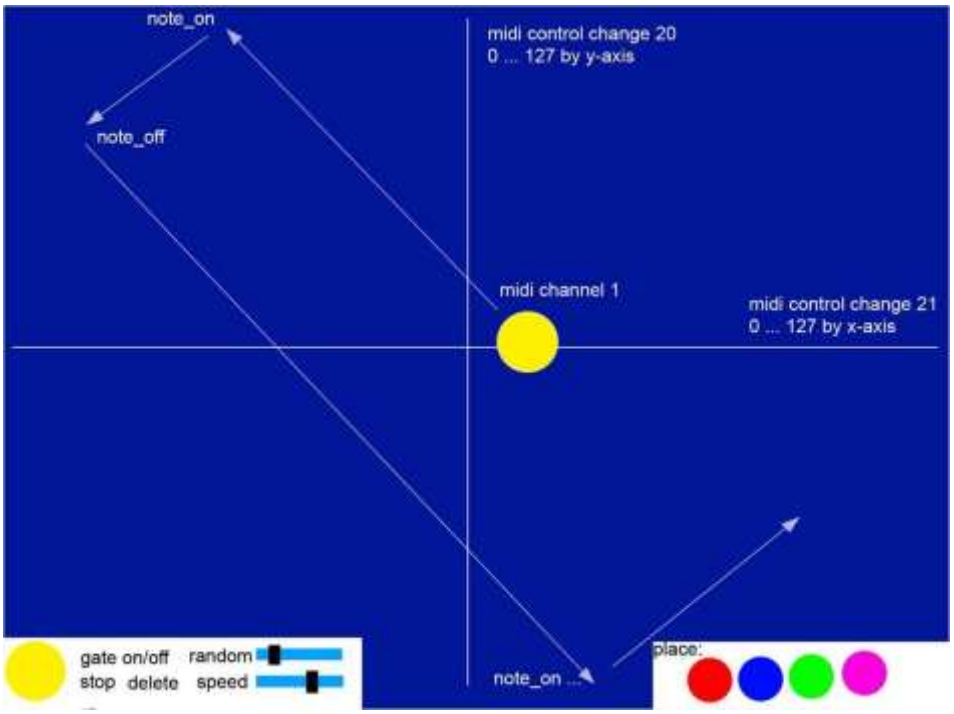

Figure 10. Descriptive picture of the instrument from a unpublished specifications document

4.1.2 Experiences

Only glitch found in the system at the beginning of its testing was the way the collisions of objects to walls sent MIDI data. The custom software sent as specified a MIDI “note on” command to the MIDI host but unfortunately it did not send a MIDI “note off” command after that. The MIDI notes were toggled on and off each time the objects hit walls. Here comes the keyboard metaphor to play: when a MIDI keyboard is played it sends a MIDI “note on” command when a key is pushed and a MIDI “note off” command when the said key is released. This fact had slipped my mind. Fortunately the problem was simple to fix and the system was ready for testing.

Because of the aleatoric nature of the instrument it was first tested with VST plugins that imitated the sound of nature. There was five instruments used in the first performances, made by Xoxos, winner of KVR developer’s challenge of 2007. These all used synthesizing instead of samples to produce sounds of nature. They are called Sounds of Nature VST Pack.

- Oscine tract VST emulates birdsong
- Synsect VST emulates insects chirping
- Wind, rain and thunder VSTs emulate the natural phenomena their names imply

Creating a soundscape of a forest with some wind and rain was easy to accomplish with the system. Each ball on the screen represented one MIDI channel on the REAPER sequencer. Each of the MIDI channels was connected to a Xoxos' VST plugin one ball controlled one instrument. Reverb was added for making the sounds more natural.

The velocity of the moving balls on the screen was directly proportional to the frequency of the instruments playing the MIDI notes as more velocity means more collisions with the borders. This allowed easy adjustments of the amount of birds, insects, wind, rain and thunder respectively.

It was early found out that one interesting thing to accomplish with the system was to record the MIDI data coming from the custom software into the Reaper sequencer. It was possible to record one performance and play it back at the same time as recording a new one. It was possible to play back one performance with a set of five instrument plugins, change the instruments to new ones, record a new performance consisting of the earlier instruments and the new instruments. This could probably prove to be a fast method of doing long and complex, never repeating soundscapes.

Because the design of the instrument is modular it is possible to create new kinds of mappings for the MIDI data generated by aforementioned custom software. MIDI Control Change-commands can be mapped to practically anything in the host that receives the MIDI data. Also MIDI note pitch (which is sent on five MIDI channels, one corresponding to each of the objects moving in the multitouch screen) can be mapped to different parameters altogether. It would be easy to make all data from the five MIDI channels to control parameters of just one monophonic voice of a software synthesizer. This would enable 20 singular parameters to be controlled as there are five MIDI channels, each outputting two continuous control parameters.

Ideas for further development of the Seq1 include the possibility of making the balls on the screen collide and trigger sounds from the collisions. Another possibility would be the addition of gravity to affect the balls on the screen. The gravity could draw the balls either downwards or around an object at the middle the same way as in the old Space War!-computer game.

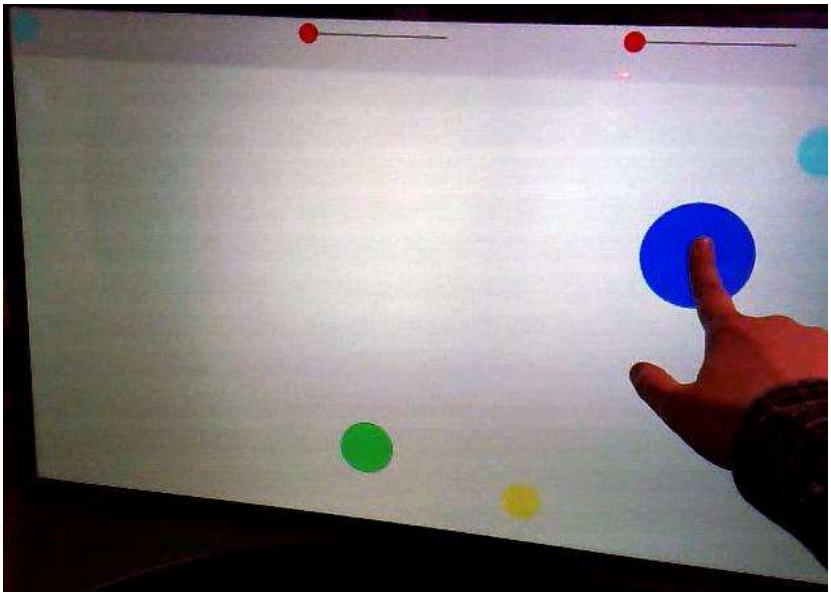

Figure 11. Working version of the Seq1 software based musical instrument from a unpublished video taken at Ylivieska circa 2010.

5. About classification

5.1 Models for classifying software based musical instruments

Here are some examples of ways of classifying musical instruments. The history of musical instruments employing electricity in its sound generation is long but electronic instruments started to become more common in the first half of the 20th century. Taken account the general length of the history of musical instruments it is not a surprise the classification systems are not necessarily conclusive.

5.1.1 Ancient

There have been numerous ways to classify musical instruments all over the world. The classifications may have been based for example by allegories to the four elements (air, water, fire, earth), by the way muscle power is transferred to the instrument or by the material of the instrument. One method was a classical Chinese way of classifying instruments named pa yin system. Pa yin referred to "eight sounds" made by different material. Included eight different materials were: metal, stone, silk, bamboo, gourd, pottery, leather and wood. It also referred allegorically to the eight seasons and eight winds presented in the Chinese culture. (Kartomi, 2001, p. 300)

These are old classifications do not fit very well to any kind of software based musical instruments although they might be considered of some use for examining user interfaces.

5.1.2 Modern

Elemental Organology

Concepts of four elements from ancient times have been used in Elemental Organology. It is a modern approach that divides musical instrument by the element of nature they represent:

- Earth (gaiaphones)

- Water (hydraulophones)

- Air (aerophones)

- Fire (plasmaphones)

- Idea (quintephones)

Electrophones are by definition musical instruments that produce sound primarily by electrical means. In Elemental Organology electrophones are a subcategory of the fifth category, that is, Quintephones. The category refers to "Idea" in contrast to natural elements; processes or procedures not limited by matter. While including electrophones this category also includes sound produced for example by neural or optical computations. (Mann, 2007, p. 6)

While software synthesizers most definitely can be categorized under quintephones the explaining power of elemental organology perhaps remains to be seen. It seems like a practice in simplifying musical instrument classification as much as possible.

Hornbostel Sachs-system of musical classification

Most influential scheme on classifying musical instruments is thought to be the Hornbostel Sachs-system. This scheme was co-developed by Erich Moritz von Hornbostel and Curt Sachs. First version of the system was introduced in Zeitschrift für Ethnologie in 1914. This German document can be found hosted online in various formats at https://archive.org/details/zeitschriftfre46berluoft (p.553), by University of Toronto - Robarts Library. (von Hornbostel & Sachs, 1914)

5.2 Frameworks and theory for evaluating software based musical instruments

Several different models and frameworks have been proposed for evaluating new digital musical instruments. These models are far and between in their field of science.

5.2.1 Levitin et Al.: Control parameters for musical instruments: a foundation for new mappings of gesture to sound

Daniel J. Levitin et al. introduce a new way of mapping gestures to sound (Levitin, McAdams, & Adams, 2002a) in the field of psychology and music theory. Their approach is based on the fact that electronically controlled musical instruments are too often based on the keyboard metaphor and it restricts their potential and expressiveness.

Computer-generated sound can far surpass the expressiveness of traditional keyboards; thus new metaphors for the user interface should be considered. The model proposed is based around the concept of a musical event and goes into great detail with different parts of the event and analysis on how different kinds of traditional instrument playing gestures fit the model (Levitin et al., 2002a, p. 184).

With this model different kinds of traditional and experimental musical instruments can be classified by the type of the energy source being controlled, by the physics of the instrument and by the gestures it is played with. For example: for violon cello, the type of energy (moving the bow) would be continuous, physics would be defined as the type “stringed instrument” and gestures used would be of type bowing and pressing the strings.

5.2.2 Tellef Kvifte and Alexander Refsum Jensenius: Towards a Coherent Terminology and Model of Instrument Description and Design

Tellef Kvifte and Alexander Refsum Jensenius from the department of musicology at University of Oslo try to create a framework for creating a coherent terminology describing, comparing and discussing different instrument designs with perspectives of the listener, performer and the conductor (Kvifte & Jensenius, 2006, p. 1). They claim that much of recent research into new kinds of musical instruments describe the construction of new musical instruments and their aspects but do not address more general principles of musical instrument construction. (p2) They claim that one part of

this problem is the lack of proper terminology and intend to address this part of the problem.

Kvifte and Jensenius advocate a new field of study called theoretical organology. In all three perspectives (of the listener, the performer and the conductor) they acknowledge two important aspects of instrument construction, that is feedback and mapping.

Also present is a model of a instrument constructor's perspective and a way of describing mapping in a chart form. Conclusion of the paper is that analysts and constructors would benefit from a unified model based on the various models proposed for analyzing new digital instruments which is the intent of the work at hand.

5.2.3 Blaine and Fels: Contexts of Collaborative Musical Experiences

Blaine and Fels focus on learning curve and expressivity of collaborative musical instruments designed (Blaine & Fels, 2003a). This is an important issue; if a novice player finds an musical instrument too hard to use he will lose interests soon. If the instrument is too easy and monotonous it will be also forgotten soon. A musical instrument should have a learning curve that is challenging so that it keeps the user interested. Blaine and Fels identify design considerations by investigating a series of contemporary collaborative software based musical instruments. Optimizing for a balance between simplicity and virtuosity is not easy. In the context of collaborative musical instruments design demand for example that a collaborative musical instrument installed in public space for anyone to use should have such interface that it can be learned quickly but which has the possibility to learn new, more complex features about the instrument. Contexts discussed are focus, location, media, scalability and player interaction.

5.2.4 Wooller et al.: A framework for comparison of process in algorithmic music systems

Wooller et al. (Wooller et al., 2005a) define a framework from a different perspective, meant for comparing processes in algorithmic music systems. The framework does not concern user interfaces but the way in which digital music and sound synthesis treats data. Four classes of algorithms are defined which seem to be capable of overlapping.

First class, linguistic/structural means music that is created with analytical theoretical constructs that are enough precise enough for generating music. Generative in this context means that the music is generated by mathematical recursion and the idea is derived from generative language studies – language or in this case music is seen as a mathematical construct that can be generated by picking correct grammar and syntax.

Second class, interactive/behavioral means music that has no input from outside the system. For example feedbacking the background noise of a analog mixing desk through delay loops can create music of this type (e.g. frippertronics)

Third class, creative/procedural is derived from the second class in that the composer "sets the ship in motion on the seas" as to say. This might mean in the example of the second class that the composer inserts a sound into the feedback loop. An example Wooler makes on this class is a composition called "Its gonna rain" by Steve Reich which is consists of a single phrase repeated all over again, copied from one to another

tape loop and back so that the texture of the sound constantly changes. Guessing what the phrase is left as an exercise for the reader.

Fourth class, biological/emergent refers to non-repeatable, non-deterministic music such as wind chimes. Wind chimes sound always the same but you can't easily make two identical longer recordings of these.

As said, these classes are not exclusionary and nothing prevents one from building a system consisting of more than one of them.

Wooller discusses several prior models of classifying algorithmical musical systems and notes that there are lots of different perspectives for these such as linguistic, musicological, sociological, user-centered, artist-centered and programmer-centered. Wooller concludes that his framework is best suited for the creators of interactive musical systems where there is a human involved in the "feedback loop" – the framework in this situation helps make the invisible processes happening in the synthesis module more visible.

The framework places algorithmic musical systems in a two-dimensional plane. The horizontal axis of the plane presents the system by a scale from analytical to transformational to generative. Horizontal axis represents the breadth of context of the system.

Analytical means a system that uses a lot of data but pick only parts of it. For example, a system making a polyphonic composition to a monophonic one would be a example of this type of system.

Transformational means systems that alter some parameter of the musical data but does not destroy it. For example a system transposing every note of a composition by some known ruleset would be transformational.

Generative is a system which produces greatly more data than is being fed to the system. Example of this would be a chaotic system generating lots of aleatoric music based on a few control parameters or maybe a cellular automata.

Breadth of context is best understood by comparing two similar sound generating software applications that are identical although one has four parameters that can be controlled and the other one only has one. The algorithm with four parameter has a larger breadth of context.

5.2.5 Oore: Learning Advanced Skills on New Instruments

Sageev Oore is a professional musician who has also learned to play new software based musical instruments. Because he is already an expert in traditional musical instruments he has different kind of insight into learning new kinds of software based musical instruments. There is a fundamental difference between learning to play a classical musical instrument or a new software based musical instrument. While learning to play a classical musical instrument the player gets lots of accumulated knowledge about playing the instrument whereas when learning to play a software based musical instrument there is no previous knowledge. (Oore, 2005a)

5.3 Attributes of software based musical instruments in works described

These are vital attributes derived from the above frameworks concerning analyzing musical instruments and research into them. They are ways which differentiate the instruments from each other thus making it possible to draw some conclusions. It is possible to categorize e.g. instruments by their appearance or the flexibility of their mapping.

5.3.1 Appearance

Appearance describes the physical characteristics of the instrument. Instrument may be a classical instrument augmented with digital technology or something completely different such as VMI – Virtual Musical Instrument that does not have a dedicated physical musical interface but is controlled via computer screen. Musical instruments based on virtual reality such as Mäki-Patola's virtual air guitar imitate their archetypes but may be quite different in the way they are played.

5.3.2 Mapping

Mapping describes how parameters on the user interface of the instrument are translated into the synthesis engine of the instrument. In certain cases such as augmented classical musical instruments they may partly mimic the archetypical instrument they are built on.(Kvifte & Jensenius, 2006, p. 1) claims that although mapping between gestures on the musical instrument and the sound produced is discussed widely more general principles are omitted. This directly involves the design of the new software based instruments created: what kind of sounds do the listener expect from a certain kind of a gesture? From the perspective of the performer aspects that should be studied are what kind of mappings prove to be most intuitive and creative?

5.3.3 Latency

Latency can mean the overall system latency measured from the user triggering a sound on the user interface in order to command the synthesis engine to start playback of its synthesis engine. Latency affects the way an instrument is played. Acoustic instruments have naturally low latency – digital processing raises the latency, sometimes so much that the way in which the digital reincarnations of instruments are played differs remarkably from the original one, for example with the augmented Djembe created by Mäki-Patola et al. (Mäki-Patola et al., 2004b)

In creating temporal media such as music the amount of latency is not the only important point but also the fact that the latency, low or high, should stay the same, synchronized, all the time so no drifting happens when recording or playing back recorded audio. Drifting and wobbling were problems in the era when analog tapes were used to record music.

5.3.4 Learning Curve

Classical definition of the learning curve is "A graph showing some measure of the cost of performing some action against the number of times it has been performed" (S. Jordà, 2004a, p. 2) and from a manufacturing point of view can be presented as a falling graph showing how much less time and resources it takes to produce one unit after learning occurs and the manufacturer gets more efficient at his job (falling time-metric).

From a psychological point of view learning curve is a raising graph, showing how much learning happens for a certain task (learning to play the flute, for example) versus time. Steep learning curve does not mean hard learning process – it means that more learning happens at the beginning of the learning process than later.

In musical context learning curve is not as easy to define. It is not possible to think it as amount of reduced time versus work done as learning an instrument involves more and more complex ways of controlling the instrument, not playing it in some way "more".

Levitin describes the learning curve in musical context "as the amount of time it takes to a novice to gain enough skill with an instrument so the experience of playing it becomes rewarding. (S. Jordà, 2004a, p. 3)

Wanderley and Orio define learnability "as the time needed to learn how to control a performance with a given controller" (S. Jordà, 2004a, p. 3). This definition came from research into usability of musical controllers. Learning the controller or an musical instrument enough to play a performance means to have a certain point in the learning curve that can be seen essential to mastering the instrument. According to Wanderley and Orio it takes about 10 years to reach this point for a traditional musical instrument. Will it take as long for new software based musical instruments?

Challenging learning curve for a musical instrument should be targeted in design; it keeps the player interested in learning the instrument. Expressivity is directly linked to learning curve – if there is not enough expressivity there is not enough to learn about the instrument and the instrument becomes boring. Example of this might be a triangle or a kazoo.

Probably because of problems in ways of measuring the musical learning curve no research that would display real learning curve graphs for classical musical instruments seem to be made. For really comparing different learning curves it would be essential to measure them but even without this information speculation about the differences between learning curves of various classical instruments can be intuitively grasped. Jorda (S. Jordà, 2004b, p. 2) suggests for this purpose a graph that has efficiency as x-axis and time as y-axis.

5.3.5 Efficiency

According to Jorda the engineering point of view on efficiency describes it as the ratio of useful energy output to energy input (S. Jordà, 2004a, p. 3). In HCI, efficiency is defined as the work output of a human interacting with a computer. (S. Jordà, 2004a, p. 3) In musical context, Jorda describes efficiency (musical instrument efficiency) as musical output complexity divided by control input complexity. Musical output complexity can be understood as the complexities inherent in different instruments such as microtonal capabilities of the violin. Other given names for the same variable are musical range and expressive range. Control input complexity has to do with the mapping, precision and degrees of freedom of control a musical controller can have. Musical output complexity increases as the performer learns the instrument or the controller and becomes able to control new nuances.

5.3.6 Expressivity

Blaine and Fels describe expressivity through the design process of new musical instruments; what kind of requirements would the designs have to have to be both

intuitively easy to use and expressive enough to allow “an upgrade path to virtuosity”. (Blaine & Fels, 2003a, p. 1)This is closely related to the concept of learning curve. The upgrade path to virtuosity would consist of a learning curve that allows the novice to quickly learn basics of the instrument but at the same time includes subtleties that allow continuous learning so that the instrument doesn’t become boring after a while. These subtleties imply well thought mappings from the user interface into the synthesis engine and rich feedback for the player of the instrument, both audio and visual. (Blaine & Fels, 2003a, p. 2)

5.3.7 Contextual Breadth

Contextual Breadth is best understood by the amount of input a computational algorithm creating sound or music takes. An algorithm that takes data from less inputs has narrower contextual breadth than an algorithm that has more inputs. (Wooller et al., 2005a, p. 10) According to Wooller contextual breadth affects more than just for example temporal and textural dimensions. In the future, information theory may be used to quantitatively measure the contextual breadth of an algorithm.

5.3.8 Levels of encapsulation

Wooller et al (Wooller et al., 2005a, p. 6) borrow encapsulation from OO languages as one aspect which can be discussed in analyzing algorithmic musical systems and probably instruments. Encapsulation can be used to hide potential complexity inside a component so that it has its input parameters and outputs but can be mentally perceived more easily. Encapsulating too big parts of the system may be problematic as it may filter out too much details of a component encapsulated, too little encapsulation on the other hand may not reduce the perceived complexity of the system as a whole.

5.3.9 Feedback

Feedback is a phenomenon in the context of software based musical instruments that can be addressed on several levels. (Kvifte & Jensenius, 2006, p. 4) find that it is important to show feedback from different parts of the system for the player. It is also important to understand the implications of feedback. For example: a Theremin can only be played if the player receives feedback from its sound output - as there is no keyboard or similar metaphor present and the player can only see her hands move in air.

For the player, feedback can come from three sources: tactile feedback from the user interface, visual feedback from the user interface or sound from the instrument itself.

As a curiosity a fourth kind of a feedback system for the player can be identified. For example (Erkut, Jylhä, & Disçioglu, 2011) have developed an real-time application that recognizes the sound of clapping hands and augments it with the synthesized sound of large group of people clapping hands. Here the feedback goes in two directions, the application adapts itself to the sound of clapping hands and the player adapts to the sound of the application. Maybe this will finally end the debate about the sound of one hand clapping.

Synthesis engine of the software based musical instrument gets control data from the user interface and may feedback parameters of the synthesis procedure back to the user interface. Good example of this is the FMOL(Sergi Jordà et al., 2007a, p. 3) which is a

software based synthesizer heavily based on the use of its GUI. It consists of oscilloscope-like lines which represent most parts of its synthesis engine (voice generator, effects processor, low frequency oscillator). Each line can be dragged with a mouse to make variations to the sound, also each line gives dynamic visual feedback on the inner workings of the system.

Synthesis engine itself may have internal feedback systems that produce sound. Algorithms producing sound can vary in this greatly, some generate large amounts of sound from a very little data, very little sound from a large amounts of data or anything in between. For more precise description of this please check the chapter on (Wooller et al.) in this work.

5.4 Problems of classification

The Hornbostel-Sachs instrument classification system is thought to be based strongly on the evolutionary thinking in sciences and humanities at the end of the centuries that preceded it but that it has remedied the chaotic state of musical instrument collections in museums. (Gétreau, 2009, p. 303)

The fact that the classification system gained prominence in the context of museums should be understood. This must have influenced its development. By large, it seems to be able to classify instruments so that they can be catalogued and identified in museums but this emphasizes the physical, external qualities of the instruments, omitting characteristics needed for their real operation.

Before the 40's the Hornbostel-Sachs system did not include electronic musical instruments. In 1940 Curt Sachs added a fifth category to the classification system. The fifth category contains electrophones (electronically driven musical instruments). This category was further expanded into three subcategories in:

> 51 Electrically actuated acoustic instruments
>
> 52 Electrically amplified acoustic instruments
>
> 53 Instruments which make sound primarily by way of electrically driven oscillators

Several people have since contributed to the model. In 2011 MIMO (Musical Instrument Museums Online) made several changes to this version of the classification system based on a previous work by Jeremy Montagu. One goal of this project was to simplify the Hornbostel-Sachs model for non-specialists (non-specialists in the context of museums). The fifth Hornbostel-Sachs category was expanded extensively. As of 2011, it now consists now of six subcategories which are still divided into total 53 more specific classes. The main subcategories of the fifth class from Hornbostel-Sachs system are:

> 51 Electro-acoustic instruments and devices
>
> 52 Electromechanical instruments and devices
>
> 53 Analogue electronic instruments, modules and components
>
> 54 Digital instruments, modules and components

55 Hybrid analogue/digital configurations

56 Software

The new fifth category is based on the work of Maarten Quanten from the Musical Instrument Museum in Brussels. It is based on the premise that it covers both modular designs that may be used in creating electrophones and basically, standalone synthesizers. The Hornbostel-Sachs system is used so that a single instrument can be defined as a combination of more than one class.

The custom built instrument presented in this thesis (6.1) would thus be defined as "542+56+546" meaning "Touchpad/touch screen/digital sequencer+Software+Digital playback/Recording device" which would, without further explanation, explain exceptionally little about Seq1. As sound for this instrument comes out of loudspeakers it would also have to also include class 515 Transducers (Microphones, pick-ups, loudspeakers) which would complicate the matter further.

Problem with the 56. category (Software) is that there are no subcategories for it. It could be expanded in various ways. One way to think about this problem is that software can nowadays emulate in real time many instruments in the 2011 Hornbostel-Sachs classification model. Combining category number 56 with another category from the taxonomy gives the impression of what it says, combining software with some different instrument. Maybe for defining emulation purposes category number 56 could be used recursively so category 56 would be a category that could have as its subcategories other valid categories from the model.

Virtual analogue synthesizers are also a problem with the category 56. They are usually self-contained keyboard instruments or sound modules which emulate the sound of analog synthesizers with DSP technology. Analog synthesizers usually employ subtractive synthesis (T. B. Holmes & Holmes, 2002b, p. 32) which virtual analog synthesizers usually emulate. In the 2011 Hornbostel-Sachs model there are subcategories for digital synthesizers with frequency modulation synthesis (541.1.*), additive synthesis (541.2.*), phase distortion synthesis (541.3.*), and direct sampling (544). Why no category for subtractive synthesis in digital domain?

Furthermore, in category 56 there are rather esoteric synthesis methods such as phase distortion synthesis which are generally rare. Why not create categories for example for much more available granular synthesis (T. Holmes, 2008, p. 310), wavetable (T. Holmes, 2008, p. 305) synthesis or proprietary synthesis methods used by manufacturers such as Ai2 by Korg or V.A.S.T (variable architecture synthesis method) by Kurzweil Music Systems?

One problem with the 2011 version fifth category is that it does not contain a class for analogue to digital or digital to analogue-converters. There is only class 546 which refers to Digital playback/Recording. There is a class for analog sound reproduction and amplifying though, class 515: Transducers Microphones, pick-ups, loudspeakers.

There is also class 547: Digital modules communicating between devices/signal convertors but as it is under class 541: Digital synthesizers it does not seem to include aforementioned converters. Audio interfaces or sound cards containing analog to digital and digital to analog converters used in modern music and sound production studio are omitted as such. Modern audio interfaces or sound cards usually employ several pairs of

analog to digital and digital to analog converters useful up to 192KHz/24 bit resolution. (T. B. Holmes & Holmes, 2002a, p. 279)

First usage of audio interfaces or sound cards is to record any analog sound and playback the sound through loudspeakers or headphones. Second usage of them is to integrate old analog equipment into digital domain via control voltage (CV) / gate interfaces (11.10). There are specific products for the integration of modern digital music production suites on computers into old analog synthesizers and studio modules such as Mark of The Unicorn Volta.

There definitely should be a subcategory for audio interfaces in the 2011 Hornbostel-Sachs system and at least two classes: direct coupled and non-direct coupled audio interfaces as the latter cannot be used as control sources of old analogue equipment. Moreover, using the Hornbostel-Sachs system in its current state with software based musical instrument does not give much of a theoretical or practical benefit.

6. Sketch of a new model

This model is a synthesis of frameworks presented previously and the experiences of designing and constructing Seq1. It consists of five main classes. Three of the classes that come after the first one have different kinds of feedback to the previous classes. Classes are:

- Player
- UI/Controller
- Mapping
- Synthesis
- Sound output

Different aspects of these feedback systems can be defined such as latency over feedback from class 5. (Sound) output into class 1. (Player). While acoustic instruments do not have inherent latency digital processing may cause it and this may affect the way the whole instrument can be played.

Frameworks described in 3.3 can be used to explain different classes when applicable to the instrument analyzed.

Attributes described in 3.4 are directly interconnected to different aspects of this model and can be described, as needed, to the instrument being analyzed.

There are no previous models of this kind for analyzing software based musical instruments. Thus, comparisons with previous models cannot be made.

Below is a rough draft of the model. After that, different classes are explained.

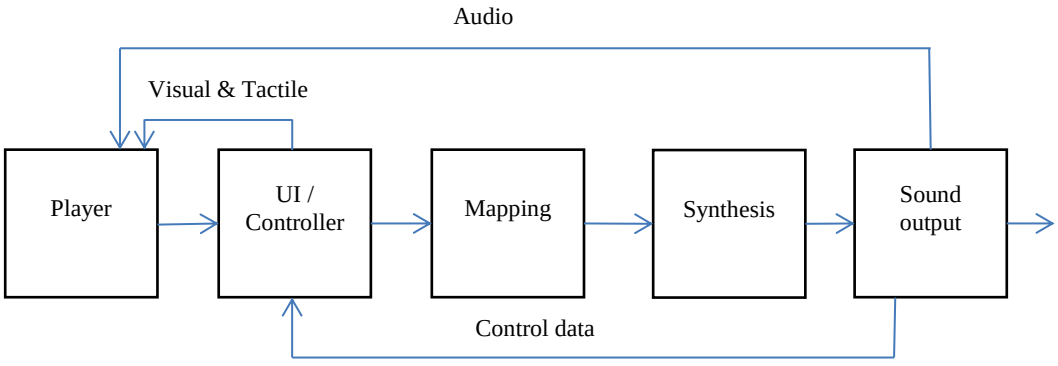

Figure 5. Proposed new preliminary model for classification showing classes and different kinds of feedback

6.1 Player

The player can get tactile feedback from UI/Controller and will get feedback from the sound output.

6.2 UI/Controller

User interface is integral part of a software based musical instrument. It may give visual or tactile feedback for the player. This part can be classified by for example Hornbostel-Sachs classification system as it is mainly concerned about the physical characteristics of the instrument at hand.

6.3 Mapping

Mapping is important but quite abstract part of software based musical instruments. It means the way the user interface is connected to the synthesis part of the instrument. Mapping is all that makes the difference between an unusable instrument and a more expressive one.

As Blaine and Fels put it, optimizing for a balance between simplicity and virtuosity is not easy. In this optimization mapping is one key element to pay attention to (Blaine & Fels, 2003b)

Mapping does not have direct feedback to any of the previous classes although mapping describes how the synthesis engine will give feedback to the user interface/controller-class.

6.4 Synthesis

The synthesis engine of a software based musical instrument is controlled from the UI/Controller class and has feedback to the UI/Controller class. It may give tactile or visual feedback on its inner workings to the UI/Controller class. For example, a cellular automata based synthesis algorithm could show the player at UI/Controller the graphical representation of its inner workings.

The synthesis engine itself can use different techniques for generating sound; subtractive, sample playback, wavetable, wavesequencing, granular synthesis etc. In the world of commercial software based musical instruments the synthesis method used is usually a main selling point.

6.5 Sound output

The sound output has a direct output to the class Player. Any distractions between Player and synthesis engine will distract the feedback to the UI/Controller.

7. Examples of classification

7.1.1 Seq1's relationship to frameworks described

Levitin et al.'s work (Levitin et al., 2002a)) does not approach very well the nature of Seq1 although Seq1 is not based on keyboard metaphor mentioned. Type of energy source does not translate to this instrument as gestures do not directly translate into sound. Physics do not translate for the same reason; gesture types can be defined but they are unique to this contraption. All in all, Levitin's work does not have enough scope to explain this contraption.

Tellef Kvifte and Alexander Refsum (Kvifte & Jensenius, 2006) concentrate on the listener, performer and conductor. Their new study field, theoretical organology has more to say about the subject. Feedback between the instrument and the player is important and other aspects of instruments modify this aspect. Higher latency delays the possible time resolution of the whole system – different mappings give various different kinds of feedback depending on the instrument. In Seq1 the horizontal position of an object moving in the screen might control the some aspect of the synthesis engine of say, a granular synthesizer or perhaps the amount of decay in a reverb effect for some virtual synthesizer played. These would both give totally different kinds of feedbacks in the system.

Blaine and Fels (Blaine & Fels, 2003a) focus on learning curve and expressivity have been addressed by Seq1. Learning curve is designed to be linear – for a novice it is easy to say, play pentatonic music with five objects on the screen without going into harder ways of mapping the system into the synthesis back end. As the skills progress, the player can program multitude of more complex mappings that affect different aspects of the sound synthesis. Expressivity of Seq1 can also be raised by more complex mapping and mapping parameters to various aspects of the sound synthesis. Being a sequencer rather than a traditional classical instrument expressivity comes from complex mappings rather than from adding more sensors to the instrument.

Wooller et al. (Wooller et al., 2005a) take a different view meant for studying systems that generate music algorithmically. Focus is on the movement and transformation of the data through the musical system and not on the interfaces used by the systems. Seq1 most closely resembles the third class of Wooller's classification creative/procedural as the results of the instrument resemble in a way the results of a complex feedback loop with five varying delay loops. Setting the five different objects bouncing from the walls of the screen is not wholly predictable as are not delay loops. Wooller's classes are overlapping and one could argue different musical systems belong partly to more than one class.

On the other dimension of Wooller's framework Seq1 is most closely generative. Seq1 generates lots of data from its input parameters – after defining the direction and mapping of the five different objects on the screen data will be generated until the software stops or something is altered sufficiently.

Breadth of context discussed by Wooller is varying on Seq1. Different compositions on the systems may have different amount of mappings and thus synthesis parameters. Control parameters do not alter between compositions, though.

Levels of encapsulation refer to the way different components are encapsulated from each other. In Seq1 this approach has been taken. Sound synthesis-part of the system can been seen encapsulated from the main program running the sequencer. Communication with these two components is one way; the sequencer sends sound synthesis part of the system MIDI data. Furthermore, sound synthesis by itself is run by software synthesizers inside a secondary sequencer (Reaper) and the sequencer itself is responsible for creating different kinds of mapping schemes for the whole system. Even further, the primary sound driver responsible for real-time audio can be changed according to hardware available – different components are interchangeable and new kinds of instruments not resembling the original Seq1 can be constructed.

As example, here are a few of the aforementioned software based musical instruments explained by the new model created.

7.2 Seq1 with Xoxos instruments according to the new model

This example shows Seq1 fitted into this model in its early incarnations, with Xoxos' virtual studio technology (VST) instruments. For general view about Seq1 see chapter 4. There may be several players using the Seq1 at the same time. Everyone will get aural and visual feedback from their actions. Playing the part of UI/Controller in this configuration is a large multitouch table. It will give feedback to the user or users from the synthesis engine in visual form: the user will intuitively know when the objects on the screen will hit the walls. As an object will hit the wall of the screen a sound is also triggered according to the parameters derived from the coordinates of the impact. In a more detailed view the core application sends a MIDI note to the MIDI port configured for use with the computer rendering the GUI – this data might be used for any other purposes than sound but MIDI data is usually dedicated for triggering sounds. The possibilities of Mapping on a modular design like this is limited only by software used in conjunction with the core application. In this case Xoxos instruments were ran inside a secondary sequencer called Reaper which took the MIDI output of Seq1 as MIDI input. Reaper has diverse mapping capabilities so any imaginable mapping scheme could be created. On synthesis side there were five different singular VST instruments producing sound: Oscine Tract VST that emulates birdsong, Synsect VST that emulates insects chirping and three VST-plugins for different forces of nature: wind, rain and thunder. All of them were part of a package created by Xoxos that was winner of KVR developer's challenge in 2007. Each one of the instruments is allocated one ball moving on the screen that sends data on a singular MIDI channel, five in all. The sound output works technically through an ASIO driver that enables low (sub 5 millisecond) latency for the audio system.

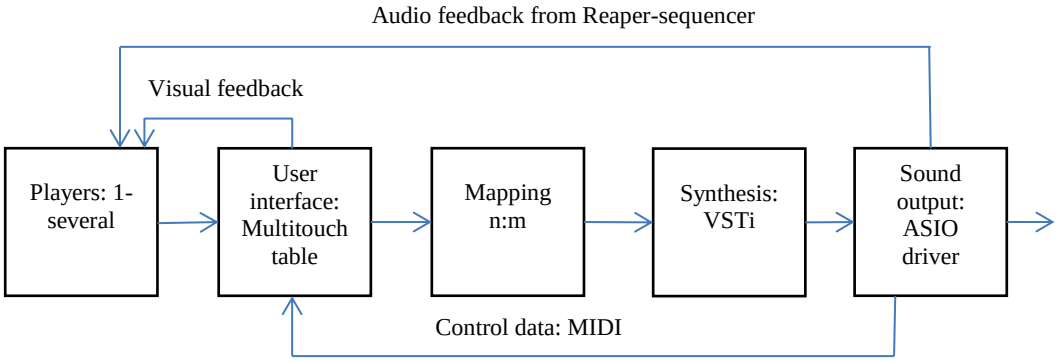

Figure 12. Proposed new preliminary model for classification showing classes and different kinds of feedback for Seq1 with Xoxos' instruments

7.3 The reacTable according to the new model

This example shows The reacTable in the context of the new model. The reacTable is one of the most important software based musical instruments coming from a research background that has been invented in the recent years and successfully turned into a commercial product. The reacTable is designed to be used by one or more persons at the same time. The player gets both visual and audio feedback from the system. Moreover, the visual feedback is sophisticated; each connection made between virtual modules shows the signal flow from one module to another. The user interface consists of a round multitouch table. This design decision was made because it was thought that at a round table everyone using the system would be equal. On top of the table are placed pucks with fiducial markers on bottom for recognition. Each marker corresponds to a certain sound generating or shaping module. The user interface is based on the metaphor of a modular analog synthesizer. Mapping of The reacTable can be defined as many-to-many. As the reacTable uses metaphor of analog modular synthesizers almost any kind of mapping is possible. As control signals and audio signals are interchangeable it is possible to make unpredictable and very complex arrangements for mapping. Mixing control- and audio signals naturally takes a heavier toll on the computer hardware running the synthesis. The synthesis engine uses what Sergi Jorda calls modular synthesis. (viite) The established term for the synthesis method used is subtractive synthesis. The synthesis engine has six kind of different types of modules to work with: audio generators, filters, controllers, control filters, mixing devices and global objects that affect every module within the area on the table it affects. Each of the different types of module has its own dedicated shape on the table so as to differentiate each other easily. The inner workings of the synthesis engine are displayed in real time on the table. The player gets visual and audio feedback from the sound output. The reacTable has low latency according to Sergi Jorda although the amount of latency is not defined. (Sergi Jordà et al., 2007b, p. 143)

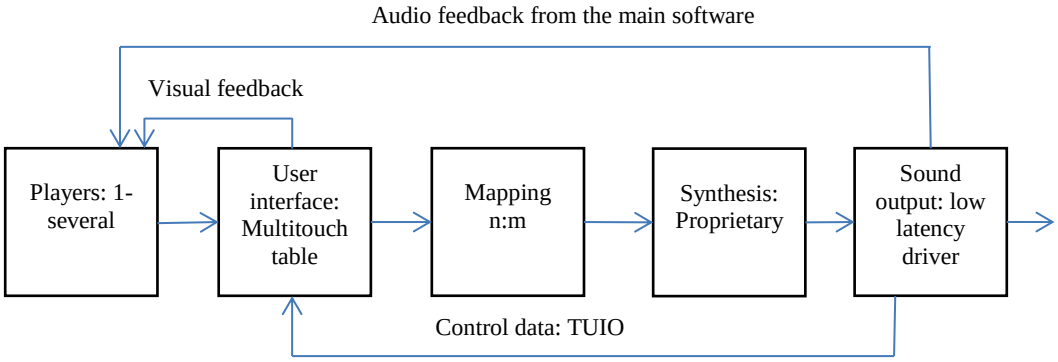

Figure 13. Proposed new preliminary model for classification showing classes and different kinds of feedback for the reacTable

7.4 Virtual Air Guitar by Mäki-Patola et al. according to the new model

Most of the instruments created by Mäki-Patola et al. were created in a CAVE-like virtual space named EVE. Six magnetic motion trackers are used. Shutter glasses are used for visual input for the user. Data gloves are used as input device for the user. 15 loudspeakers are placed to the room, behind the screen walls. (Mäki-Patola et al., 2004c, p. 11). There were three different approaches for the virtual air guitar (CAVE environment, small handheld control devices, hand tracking by video analysis) but here I analyze the version tested in CAVE environment. The player gets audio feedback from the sound output and visual feedback from the shutter glasses used in the CAVE environment. The user interface is entirely based in 3d virtual reality. Player manipulates the virtual air guitar via data gloves. The distance between user's hands determine the pitch and moving the right hand in a strumming motion triggers the sound. (Mäki-Patola et al., 2004c, p. 15) The virtual air guitar also uses foot controllers. (Karjalainen, Mäki-Patola, Kanerva, & Huovilainen, 2006, p. 964) Importance of parameter mapping is acknowledged. There is a dedicated control parameter mapping module. Mapping is specified by a visual mapping editor. (Mäki-Patola et al., 2004c, p. 12). The guitar synthesis engine uses an adapted Karplus-Strong synthesis algorithm to create the electric guitar sound. As with a real electric guitar, after the audio synthesis phase, further audio effects are applied to the signal including distortion, delay and reverberation. These would correlate to guitar stomp boxes used with a physical electric guitar. The synthesis engine was developed in a DSP platform called Mustajuuri . (Mäki-Patola et al., 2004c, p. 12). The direction of sound output relative to the CAVE environment can be specified by Vector Based Amplitude Panning. (Mäki-Patola et al., 2004c, p. 11) There are 15 speakers mounted to the walls of the space occupied by the user which gives the ability for 360 degrees panning of the sound. Another Mäki-Patola's virtual instrument, the Virtual Xylophone is created by the same components and has an overall latency of 60 milliseconds from playing a sound and hearing it. (Mäki-Patola et al., 2004c, p. 13) Nowhere the latency of the Virtual Air Guitar is explained but it is safe to assume that 60 milliseconds is the same.

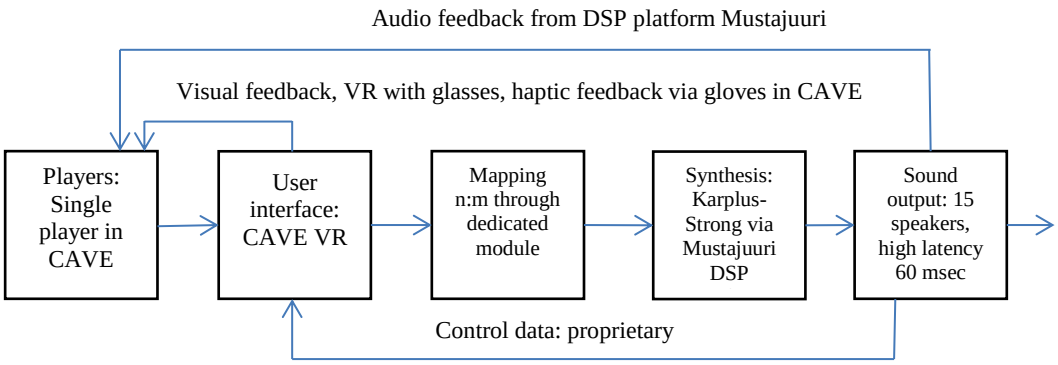

Figure 14. Proposed new preliminary model for classification showing classes and different kinds of feedback for the Virtual Air Guitar

8. Software based musical instruments in the context of frameworks

8.1 Discussion and implications

Previous exercises found in literature about designing software based musical instruments gave good foundation for designing a new one. Attributes and frameworks presented for evaluating software based musical instruments gave an overview of the subject; where to start the building of a new model.

The design and creation of the software based musical instrument Seq1 presented in this work gave ideas and a rough overview creating the model. From the previous literature certain important aspects for the design of this style of instrument were found – and found again and again. It became evident that these should be taken in the account for the design of the new instrument and hence the model.

First thing was modularity; containing different logical parts of the system into sections or modules gave improvements against a singular monolithic system. In the case of Seq1 the sound synthesis modules were interchangeable; any virtual on hardware equipment understanding MIDI could be used. Same with the MultiTouch Cell display. Another vendor using TUIO protocol could be used. Also, in case of bugs found in the system it was easier to pinpoint the source of the problem with a modular design.

It became evident that almost every previous software based musical instrument could be thought as a modular system on a more abstract level. This is the basis of the model – five different sections that interact with each other and contribute to the working of the system. From feedback loops within the model aspects such as type of visual or tactile feedback, visualization of the processes of the synthesis engine or the audio itself can be defined.

As the creation of the model has been based on the design and creation of Seq1 and Seq1 is based on previous attempts to construct such instruments the Seq1 fits to the model rather well.

Previous classification systems investigated in this work either did not either cover the aspects in software based musical instruments or did have different kind of aspect in explaining the phenomena. On the other hand, work related to software based musical instruments had many good perspectives on the system, varying from the different ways of creating generative synthesis engines into the relation between different kinds of parameter mapping relating to expressiveness and learning curve.

8.2 Conclusions

The field is new; no previous models for analyzing this kind of instruments have been created although there is evidence in previous literature for the need of one. There was similar kind research done before but not quite on the subject itself.

Hornbostel-Sachs classification system for musical instruments can be claimed the most important of its kind right now. It is omitting means for describing aspects of software based musical instrument although its recent amendments have added tens of new subclasses which should make it capable describe software based musical instruments.

Main sources for information in creating the sketch for the new model were the Hornbostel-Sachs classification system, other classification systems described in this work, theories and frameworks described in this work, and the design and construction of the Seq1.

The base for creating Seq1 itself was in previous literature about creating similar kinds of software based musical instruments. This gave insight into important aspects of such designs.

One limitation of this work is the long time it took to work out. The original work was concerned about generative music in games which took the investigation to for example Wooller's work in comparing different algorithmic music systems. (Wooller et al., 2005b) Although not in the core of the subject at hand now it sheds interesting light on the future of the current topic. Wooller's work would point into the synthesis engine-part of the current model created. This is something to consider in the future.

Although mapping is discussed with the model, Levitin's (Levitin, McAdams, & Adams, 2002b) thoughts should be addressed further when considering the future of the model. As mentioned before, Levitin goes into ways of making software based musical instruments more expressive than traditional keyboards. Different kind of physics and energy sources could be assimilated in the model.

Learning curve could be further analyzed in the model. Many people have discussed about learning curve concerning software based musical instruments but no one has come forth with a real way of materializing one, for example (Oore, 2005b), (Blaine & Fels, 2003b) and (S. Jordà, 2004b).

As the experience of designing and realizing the Seq1 was in another time and another place (2010, Ylivieska) so the design can work as a reference for further work like it but it realistically cannot be developed further. Seq1 can be considered as a success. It was developed within resources and time allocated for it. In ways it surpassed its design as new ways of using it were learned; recording whole performances made by it with a secondary sequencer, overdubbing previous recordings. Unfortunately the system was never used in a production environment. Different ways of using it were discussed; it could be used as a background sound or music generator that the public could adjust for playing sounds of the nature or something more experimental. Preliminary tests on this kind of this kind of applications were carried out in the laboratory and there are even midi sequences left of the experimental performances. The static midi sequences are quite irrelevant as the main point of the system was being able to create sound in real time.

In the future main task is first the refine the model and test it against different kinds of software based musical instruments. Secondary to this task would be designing a new instrument based on the research, the refined model and the Seq1.

References

Bischof, M., Conradi, B., Lachenmaier, P., Linde, K., Meier, M., Pötzl, P., & André, E. (2008). Xenakis: combining tangible interaction with probability-based musical composition (pp. 121–124). Bonn, Germany: ACM. http://doi.org/10.1145/1347390.1347416

Blaine, T., & Fels, S. (2003a). Contexts of collaborative musical experiences (pp. 129–134). National University of Singapore. Retrieved from http://dl.acm.org/citation.cfm?id=1085745

Blaine, T., & Fels, S. (2003b). Contexts of collaborative musical experiences (pp. 129–134). National University of Singapore. Retrieved from http://dl.acm.org/citation.cfm?id=1085745

Böttcher, N., Gelineck, S., & Serafin, S. (2007). PHYSMISM: a control interface for creative exploration of physical models. In *Proceedings of the 7th international conference on New interfaces for musical expression* (pp. 31–36). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1279743

Crnkovic, G. D. (2010). Constructive research and info-computational knowledge generation. In *Model-Based Reasoning in Science and Technology* (pp. 359–380). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-15223-8_20

Erkut, C., Jylhä, A., & Disçioglu, R. (2011). A structured design and evaluation model with application to rhythmic interaction displays (pp. 477–480). Retrieved from http://www.nime2011.org/proceedings/papers/M16-Erkut.pdf

Geiringer, K. (1978). *Instruments in the History of Western music*. Oxford University Press.

Gétreau, F. (2009). Curt Sachs as a theorist for music museology. *Music's Intellectual History*, 303–313.

Holmes, T. (1985a). *Electronic and experimental music*. New York: Scribner's.

Holmes, T. (1985b). *Electronic and experimental music*. New York: Scribner's.

Holmes, T. (2008). *Electronic and experimental music: technology, music, and culture*. Routledge. Retrieved from http://www.google.com/books?hl=fi&lr=&id=aT5nAQAAQBAJ&oi=fnd&pg=PP1&dq=electronic+and+experimental+musci+2008&ots=D8uw1XNso8&sig=OVQ9MGe_xJEjjUK_Canu9H9lsQQ

Holmes, T. B., & Holmes, T. (2002a). *Electronic and experimental music: pioneers in technology and composition*. Psychology Press. Retrieved from http://www.google.com/books?hl=fi&lr=&id=RsvdGY9VFLUC&oi=fnd&pg=PP13&dq=Electronic+and+experimental+music&ots=LbMVLT-dPh&sig=VTnx6NctsMGmO5ZI2oLmArsU8qg

Holmes, T. B., & Holmes, T. (2002b). *Electronic and experimental music: pioneers in technology and composition*. Psychology Press. Retrieved from http://www.google.com/books?hl=fi&lr=&id=RsvdGY9VFLUC&oi=fnd&pg=PP13&dq=Electronic+and+experimental+music&ots=LbMVLU0dNm&sig=zFsOCv-hRmFTxHThVhuseNjlRn8

Jo, K. (2008a). DrawSound: a drawing instrument for sound performance (pp. 59–62). Bonn, Germany: ACM. http://doi.org/10.1145/1347390.1347405

Jo, K. (2008b). DrawSound: a drawing instrument for sound performance (pp. 59–62). Bonn, Germany: ACM. http://doi.org/10.1145/1347390.1347405

Jo, K. (2008c). DrawSound: a drawing instrument for sound performance (pp. 59–62). Bonn, Germany: ACM. http://doi.org/10.1145/1347390.1347405

Jordà, S. (2004a). Digital instruments and players: part I—efficiency and apprenticeship (pp. 59–63).

Jordà, S. (2004b). Digital instruments and players: part I—efficiency and apprenticeship (pp. 59–63).

Jordà, S., Geiger, G., Alonso, M., & Kaltenbrunner, M. (2007a). The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces (pp. 139–146). Baton Rouge, Louisiana: ACM. http://doi.org/10.1145/1226969.1226998

Jordà, S., Geiger, G., Alonso, M., & Kaltenbrunner, M. (2007b). The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces (pp. 139–146). Baton Rouge, Louisiana: ACM. http://doi.org/10.1145/1226969.1226998

Karjalainen, M., Mäki-Patola, T., Kanerva, A., & Huovilainen, A. (2006). Virtual air guitar. *Journal of the Audio Engineering Society*, *54*(10), 964–980.

Kartomi, M. (2001). The classification of musical instruments: Changing trends in research from the late nineteenth century, with special reference to the 1990s. *Ethnomusicology*, 283–314.

Kvifte, T., & Jensenius, A. R. (2006). Towards a coherent terminology and model of instrument description and design (pp. 220–225). IRCAM—Centre Pompidou. Retrieved from http://dl.acm.org/citation.cfm?id=1142271

Levitin, D. J., McAdams, S., & Adams, R. L. (2002a). Control parameters for musical instruments: a foundation for new mappings of gesture to sound. *Organised Sound*, *7*(02), 171–189.

Levitin, D. J., McAdams, S., & Adams, R. L. (2002b). Control parameters for musical instruments: a foundation for new mappings of gesture to sound. *Organised Sound*, *7*(02), 171–189.

Maki-Patola, T., Hämäläinen, P., & Kanerva, A. (2006). The augmented djembe drum: sculpting rhythms (pp. 364–369). Paris, France: IRCAM — Centre Pompidou. Retrieved from http://portal.acm.org/citation.cfm?id=1142215.1142304&coll=GUIDE&dl=GUIDE&CFID=14786698&CFTOKEN=62085871

Mäki-Patola, T., Laitinen, J., Kanerva, A., & Takala, T. (2004a). Experiments with virtual reality instruments (pp. 11–16). Vancouver, Canada: National University of Singapore. Retrieved from http://portal.acm.org/citation.cfm?id=1085946&dl=GUIDE&coll=GUIDE&CFID=14786698&CFTOKEN=62085871

Mäki-Patola, T., Laitinen, J., Kanerva, A., & Takala, T. (2004b). Experiments with virtual reality instruments (pp. 11–16). Vancouver, Canada: National University of Singapore. Retrieved from http://portal.acm.org/citation.cfm?id=1085946&dl=GUIDE&coll=GUIDE&CFID=14786698&CFTOKEN=62085871

Mäki-Patola, T., Laitinen, J., Kanerva, A., & Takala, T. (2004c). Experiments with virtual reality instruments (pp. 11–16). Vancouver, Canada: National University of Singapore. Retrieved from http://portal.acm.org/citation.cfm?id=1085946&dl=GUIDE&coll=GUIDE&CFID=14786698&CFTOKEN=62085871

Mann, S. (2007). Natural Interfaces for Musical Expression: Physiphones and a physics-based organology. In *Proceedings of the 7th international conference on New interfaces for musical expression* (pp. 118–123). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1279761

Moog synthesizer. (2014, December 17). In *Wikipedia, the free encyclopedia*. Retrieved from http://en.wikipedia.org/w/index.php?title=Moog_synthesizer&oldid=638277116

Moore, G. (2012). *Evolution X-Session MIDI controller*. Retrieved from http://commons.wikimedia.org/wiki/File:Evolution_X-Session_MIDI_controller.jpg

Oore, S. (2005a). Learning advanced skills on new instruments (pp. 60–64).

Oore, S. (2005b). Learning advanced skills on new instruments (pp. 60–64).

Otonkoski, L., & Barrière, J.-B. (1991). *Klang: uusin musiikki*. Gaudeamus.

Palacio-Quintin, C. (2003). The hyper-flute (pp. 206–207). Montreal, Quebec, Canada: National University of Singapore. Retrieved from http://portal.acm.org/citation.cfm?id=1085714.1085764

Palacio-Quintin, C. (2008). Eight Years of Practice on the Hyper-Flute: Technological and Musical Perspectives. In *NIME* (pp. 293–298). Retrieved from http://www.academia.edu/download/30875679/10.1.1.140.3830.pdf#page=311

Patten, J. (2016). *Audiopad*. Retrieved from http://tangible.media.mit.edu/project/audiopad/?item=2

Patten, J., Recht, B., & Ishii, H. (2002a). Audiopad: a tag-based interface for musical performance (pp. 1–6). Dublin, Ireland: National University of Singapore. Retrieved from http://portal.acm.org/citation.cfm?id=1085171.1085175

Patten, J., Recht, B., & Ishii, H. (2002b). Audiopad: a tag-based interface for musical performance (pp. 1–6). Dublin, Ireland: National University of Singapore. Retrieved from http://portal.acm.org/citation.cfm?id=1085171.1085175

Prendergast, M. J. (2000). *The Ambient Century*. Bloomsbury.

RTP MIDI. (2014, December 1). In *Wikipedia, the free encyclopedia*. Retrieved from http://en.wikipedia.org/w/index.php?title=RTP_MIDI&oldid=636176614

Schiettecatte, B. (2011). *English: Close-up of 4 Audiocubes*. Retrieved from https://commons.wikimedia.org/wiki/File:Detailed_view_of_Audiocubes.jpg

Schiettecatte, B., & Vanderdonckt, J. (2008a). AudioCubes: a distributed cube tangible interface based on interaction range for sound design (pp. 3–10). Bonn, Germany: ACM. http://doi.org/10.1145/1347390.1347394

Schiettecatte, B., & Vanderdonckt, J. (2008b). AudioCubes: a distributed cube tangible interface based on interaction range for sound design (pp. 3–10). Bonn, Germany: ACM. http://doi.org/10.1145/1347390.1347394

Tzevelekos, P., Georgaki, A., & Kouroupetroglou, G. (2008a). HERON: a zournas digital virtual musical instrument (pp. 352–359). Athens, Greece: ACM. http://doi.org/10.1145/1413634.1413698

Tzevelekos, P., Georgaki, A., & Kouroupetroglou, G. (2008b). HERON: a zournas digital virtual musical instrument (pp. 352–359). Athens, Greece: ACM. http://doi.org/10.1145/1413634.1413698

Tzevelekos, P., Georgaki, A., & Kouroupetroglou, G. (2008c). HERON: a zournas digital virtual musical instrument (pp. 352–359). Athens, Greece: ACM. http://doi.org/10.1145/1413634.1413698

Väisänen, J., & Jauhiainen, J. (2010). Seq1: an alaetoric music sequencer for the realtime creation of soundscapes (pp. 127–129). New York, NY, USA: ACM. http://doi.org/10.1145/1930488.1930514

von Hornbostel, E. M., & Sachs, C. (1914). Systematik der musikinstrumente. Ein versuch. *Zeitschrift Für Ethnologie*, 553–590.

von Hornbostel, E. M., & Sachs, C. (1961). Classification of musical instruments: Translated from the original german by anthony baines and klaus p. wachsmann. *The Galpin Society Journal*, 3–29.

Williams, D. (2007). *English: A Reactable at the Altman Center in 2007*. Retrieved from https://commons.wikimedia.org/wiki/File:Reactable_Multitouch.jpg

Wooller, R., Brown, A. R., Miranda, E., Diederich, J., & Berry, R. (2005a). *A framework for comparison of process in algorithmic music systems*. (B. David & E. Ernest, Eds.). Creativity and Cognition Studios. Retrieved from http://eprints.qut.edu.au/6544/

Wooller, R., Brown, A. R., Miranda, E., Diederich, J., & Berry, R. (2005b). *A framework for comparison of process in algorithmic music systems*. (B. David & E. Ernest, Eds.). Creativity and Cognition Studios. Retrieved from http://eprints.qut.edu.au/6544/

Young, D. (2002). The Hyperbow: a precision violin interface.

Young, D. (2002a). The Hyperbow controller: real-time dynamics measurement of violin performance (pp. 1–6). Singapore, Singapore: National University of Singapore. Retrieved from http://dl.acm.org/citation.cfm?id=1085171.1085188

Young, D. (2002b). The Hyperbow controller: real-time dynamics measurement of violin performance (pp. 1–6). Singapore, Singapore: National University of Singapore. Retrieved from http://dl.acm.org/citation.cfm?id=1085171.1085188

9. Appendix

These are all important areas to comprehend in order to understand this work. Modular synthesizers and voltage control are included as these concepts are archetypes for how for example the reacTable works. It uses virtual interconnected modules that employ control signals just as its archetype, analog modular synthesizers work.

9.1 ASIO

ASIO is abbreviation for Audio Stream Input/Output. It is a protocol for soundcards, delivering low audio latency between soundcard and the application supporting ASIO protocol. ASIO works on Windows platforms and is vital in allowing software based musical instruments to be played in real time.

9.2 Fiducial marker

In the context of multitouch tables a fiducial marker means for example a cube with high contrast symbol on one of its side. This allows the image tracking technology of the table to recognize the object and its position on the table. This makes it possible to assign different purposes for different objects. In the most quoted example, The reacTable, this means to assign different symbols for different signal processing or generating modules that can be intuitively placed on the table and then connected as wanted.

9.3 MIDI

Musical Instrument Digital Interface. Before the invention of MIDI audio and synthesizer equipment from different companies either had a custom digital interface for controlling their products or used CV/Gate signals for controlling equipment. As there were many versions of CV/Gate signaling, equipment from different manufacturers were not always compatible. There was a need for a simple, inexpensive digital interface that every manufacturer could implement into their products in the late seventies when microcomputers and personal computers became more common. In 1983 several competitors from the industry such as Roland, Yamaha, Korg, Kawai and Sequential Circuits managed to get out the first specification of MIDI. (T. B. Holmes & Holmes, 2002b, p. 227). Purpose of MIDI was specified as:

1) Connecting and controlling synthesizers

Synthesizer and audio equipment equipped with MIDI can talk to themselves via MIDI without the need of a specific central computer controlling them.

2) Linking computers to synthesizers

Microcomputers or personal computers equipped with a MIDI interface can control any MIDI equipped synthesizers or equipment.

MIDI does not send audio data. It sends control signals which trigger sounds or other functionality in MIDI equipped gear. For example when pushing a key on a synthesizer equipped with MIDI following data are usually sent:

- MIDI channel on which the note is being triggered (1-16)
- Pitch of the note
- Velocity (amplitude) of the note
- Duration of the note
- Aftertouch (while keeping the key down, if one pushes it a bit after the start aftertouch will message how much it was pushed)
- Pitch bender data (wheel usually sitting on the left of a synthesizer, usually modulating pitch of the note up and down.

Control change data such as aftertouch or pitch bender have a resolution of 7 bits (128 steps) which has become a problem as many applications need finer resolution. This drawback can be avoided by using NRPN (Non-Registered Parameter Number) part of the MIDI specification that allows 14 bit resolution (16,384 steps) which is enough for anybody. Moreover, there is a way to send large packets of any data to MIDI capable devices with System Exclusive messages (SysEx) (T. Holmes, 1985b, p. 233) This feature is usually used to send new firmware versions to synthesizers or PCM audio to samplers. As MIDI uses 31.25 kilo baud transmission serially with 8 bit words SysEx operations usually take long time.

Physically MIDI interfaces use a standard 5-pin DIN connector. Computer midi interfaces have MIDI IN and MIDI OUT jacks and synthesizers usually have a MIDI THRU jack too. The purpose of MIDI THRU (through) is to allow daisy chaining of synthesizers so that one MIDI output from a computer MIDI interface can control many synthesizers with the following configuration: MIDI OUT (computer) → MIDI IN (synth1) MIDI THRU (synth1) → MIDI IN (synth2) MIDI THRU (synth2) → etc. (T. Holmes, 2008, p. 228). As MIDI can send on up to 16 different MIDI channels at the same time and each channel can play several notes at the same time theoretically one MIDI interface could control 16 synthesizers at one time. Practically the latencies and throughput of the interface limit this.

Besides physical MIDI interfaces virtual MIDI interfaces can be used to route MIDI inside a computer from a program to another program or over Ethernet. MIDIOX is one program capable of routing virtual midi inside a computer http://www.midiox.com/

9.4 mLAN

For routing MIDI data streams over Ethernet Yamaha developed a multipurpose transport level protocol for transmitting video, audio and midi data streams called mLAN. mLAN was never very successful and seemingly has reached its end of life

9.5 RTP-MIDI

RTP-MIDI is a protocol that can send MIDI messages over RTP packets on Ethernet or WIFI networks. It has become increasingly popular. (“RTP MIDI,” 2014)

9.6 OSC

OSC stands for Open Sound Control. OSC is basically a modern equivalent of MIDI. It uses modern networking capabilities of computers and mobile devices, enabling for example a table computer to control a software synthesizer on a desktop computer over WIFI.

9.7 Sequencer

Sequencer in the context of music and audio is traditionally a hardware unit or software application that allows sequences of MIDI data to be recorded, edited and played back. Modern sequencers usually include capabilities of recording, editing and playing back audio data recorded through a sound card or an external audio to digital-converter. Furthermore, modern sequencers usually support VST plugins and a protocol such as ASIO in order to allow low latency playback of software based musical instruments, audio effects and audio.

9.8 Software based musical instrument

Software based musical instruments are musical instruments that employ digital technology in the generation of sound, user interface or both. Commercial products are usually based on VST technology and thus can be run on different applications that support the technology. In a broader sense all synthesizers employing digital technology are software based musical instruments.

9.9 TUIO

TUIO stands for Tangible User Interface Objects. It is “an open framework that defines a common protocol and API for tangible multitouch surfaces” TUIO was first made public when the software based synthesizer reacTable was released. After this the protocol has gained prominence and is considered to be a community standard for applications beyond musical instruments. Another example of how technology and music go hand in hand. TUIO is based on OSC so any system capable of using OSC is easier to turn TUIO capable.

9.10 Voltage Control

Bob Moog was the first to make voltage controlled synthesizer modules a commercial success although instruments before Moog have been using the same principles. (T. Holmes, 1985b, p. 208) The groundbreaking idea was to create individual modules that could be combined into a modular synthesizer. Each module would talk to each other with voltage control and there were almost unlimited possibilities in how to route and

connect modules on a big modular synthesizer. A minimal modular synthesizer setup would consist of the following modules:

- Keyboard for controlling the synthesizer
- Voltage controlled oscillator (VCO) producing primitive waveforms (sine, saw tooth, square, triangle) in the audible range (Moog oscillators go from 0.01 Hz to 40,000 Hz)
- Voltage controlled amplifier (VCA) to amplify the signal
- Voltage controlled filter (VCF), usually 4-pole low pass transistor ladder filter with resonance (Q)
- Envelope generator can be used to modulate for example the VCA in order to make different characteristics to the sound. Parameters usually controlled with an envelope generator are Attack, Decay, Sustain and Release, usually referred as ADSR envelopes. Other configurations possible. (T. Holmes, 2008, pp. 201–202, 213–214)
- Patch cords to link the modules together

Way to connect these modules with voltage control would be keyboard → VCO → VCA → VCF → Audio out. Whereas MIDI can send polyphonic playing on 16 channels at the same time voltage control can only send one monophonic note. Below is a image of first commercial Moog Modular synthesizer.

Figure 15. (“Moog synthesizer,” 2014)

9.11 VST

VST is an abbreviation of Virtual Studio Technology. It is conceived by Steinberg Media Technologies GmbH. It allows computer applications supporting it to run VST plugins in real time. VST plugins can be effect processors that manipulate audio, MIDI plugins that manipulate MIDI data or VST musical instruments that are either simulations of hardware units or instruments built solely for and by the limitations of current technology.