RWTH AACHEN UNIVERSITY

DISCO

Disk-based Interface for Semantic Composition

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

by
Mei-Fang Liau

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Reinhard Oppermann

Registration date: June 03th, 2008
Submission date: Oct 06th, 2008

Contents

List of Figures

List of Tables

Abstract

Musical structure is defined by the biological constraint of being human and having absorbed cultural influences. The semantics of music is the interrelationship between its musical elements structured by its composer.

Creating music is an iterative process. The composer captures an idea, realizes it and evaluates the result. This process repeats and the musical material evolves. The composer might constantly shift his focus between considering the song structure as a whole and a detail such as the musical feature of an individual note. However, such an activity is not well supported when moved to the desktop composition environment. Typical track-based environments are regarded as the composition tools for all aspects of music, while they are initially aimed at assisting the composition of preconceived material. Previous research has given efforts to visualize the composers' musical structure but the musical semantics are still not well considered. This results in an ineffective creative process as well as the ineffective organization of the low-level musical elements.

Therefore, we have designed the system *DISCO* tailored to support the creative process of music composition. The musical material is organized based on the composers' musical concepts and visualized in different layers. The composers are able to work at the abstract level of their structural definition, but can also easily shift to edit specific detail and its conceptually related elements. With the vertical prototype as a rhythmic composition tool, we demonstrate how the iterative process of composition could be assisted. The composers are free to perform ideation and concept realization, while the low-level musical elements are semantically organized.

Before developing DISCO, we have gained many insights of the composers' workflow through field interviews. The development of the system is based on an iterative, user-centered design process, where our prototype evolves from a storyboard to a paper prototype and finally a working software prototype. The final evaluation of the system shows the proposed workflow is in tune with the majority of the composers we interviewed. To achieve a complete environment for creative workflow of musical composition, the system shall be able to incorporate other instruments and visualize the musical semantics they convey with the consideration of their distinct musical idiosyncrasies.

Überblick

Musik wird im Wesentlichen von allgemeinen menschlichen Eigenheiten und kulturellen Einflüssen bestimmt. Die semantische Struktur der Musik basiert auf den wechselseitigen Beziehungen ihrer musikalischen Elemente.

Musik zu komponieren ist ein iterativer Prozess: der Komponist hat eine Inspiration, hält sie fest und evaluiert das Ergebnis. Der Prozess wiederholt sich mehrmals und das Musikstück entwickelt sich. Im Laufe dieses Vorgangs wechselt der Komponist mehrmals die Ebene, auf der er arbeitet, zwischen grober Struktur und einzelnen Noten. In heutigen Desktop Kompositionssystemen wird dies jedoch nur unzulänglich unterstützt. Typische trackbasierte HD Recording Umgebungen werden als Kompositionstool für alle Ebenen angesehen. Dabei sind sie vorwiegend zum Arrangieren bereits existierendem Materials geeignet. Bislang wurde in erster Linie die Visualisierung der musikalischen Struktur erforscht, die Semantik jedoch vernachlässigt. Damit bleibt sowohl der kreative Prozess an sich als auch die Organisation der musikalischen Elemente auf kleinster Ebene ineffektiv.

Daher haben wir das System "*DISCO*" zur Unterstützung des kreativen Prozesses entworfen. Das musikalische Material wird entsprechend des konzeptuellen Modells des Komponisten organisiert und in verschiedenen Ebenen visualisiert. Komponisten können auf der abstrakten Ebene der Songstruktur arbeiten, aber jederzeit bequem spezifische Details und konzeptuell Verwandtes bearbeiten. Wir zeigen mit unserem vertikalen Prototypen, wie der iterative Prozess unterstützt werden kann. Komponisten können Ideen und Konzepte im Groben realisieren und gleichzeitig die einzelnen Elemente auch auf unteren Ebenen semantisch organisieren.

Vor der Implementierung von DISCO haben wir durch Feldinterviews die Arbeitsweise von Komponisten evaluiert. Der Entwicklungsprozess des Systems ist selbst iterativ und nutzerorientert. Storyboards, Paperprototypen und schließlich auch Softwareprototypen wurden in jedem Iterationsdurchlauf von Usern evaluiert und das Feedback ins Design integriert. Die finale Evaluierung des Systems zeigt, dass der zugrunde gelegte Workflow mit dem der meisten Komponisten, die wir interviewed hatten, übereinstimmt. Um eine komplette Kompositionsumgebung zur Unterstützung des kreativen Prozesses zu erhalten, muß das System zudem diverse Instrumente unterstützen und auch deren Semantik mit ihren jeweiligen Eigenheiten visualisieren können.

Acknowledgements

This thesis would have never been accomplished without the contribution of many people.

First of all, I would like to thank Prof. Jan Borchers for having given the enlightening lectures that guided me into the area of HCI, and the creation of such a great department comprising many ingenious minds. Thanks to Prof. Reinhard Oppermann for always offering immediate support when I needed his opinion. Many thanks to Jonathan Diehl, for having offered me this challenging topic, sparing his time for the meetings, giving me so much guidance towards achieving the work effectively.

Thanks to all the participants in the user tests of DISCO. Their contributions were most helpful in providing a fresh look into the creative process of making music.

Thanks to Clio for being such a genuine friend and passing through the hard times with me.

Thanks to the creative power of i10, especially Yvonne, Nori, Leo, and Moriz, who are always very caring and offer constructive comments to my work.

Special thanks to Jonathan Ansell, who has always offered generous love and care. Thanks for spending time correcting my English grammar without any complaint.

Finally I would like to thank my family and friends for their understanding of my lack of attention due to the stress of writing this thesis.

Thank you!

Conventions

Throughout this thesis we use the following conventions.

The plural “we” will be used throughout this thesis instead of the singular “I”, even when referring to the work that was primarily or solely done by the author.

”He” is used to describe the unidentified third person merely for the purpose of readability.

The whole thesis is written in American English.

Chapter 1

Introduction

1.1 Track-Based Composition Environment

Multi-track environment is aiming at composing preconceived material.

About two decades ago, graphical tools for computer-aided composition were in their infancy. Due to the low availability of computers they were only accessible to a small group of people. In the 90s, the concept of virtual studios emerged. Aiming at supporting recording preconceived material, the software simulated the physical equipment in the studio, such as multi track recorders and mixers. Collins [2004] These devices are typically used for music production.

Creative composition is not well supported by the track-based environment.

Nowadays, computers are pervasive. More and more composers are willing to compose music in the software environment, as various kinds of software instruments and end devices can empower their creation. Nevertheless, most of the software composition tools still remain the metaphors from the studio equipment, while new requirements have emerged for the composition environment. Composers do not merely linearly record their song on the interface, but very often perform creative activities with their music software tools. As complicated as music composition is, where ideas need to be caught and organized, the multi-track environment is not competent as an assisting tool.

Tracks force users to group material physically but not in terms of their musical concepts.

Furthermore, track-based sequencers lead users to group material physically. Tracks convey the information of the material's sonic features, or their compatible input formats, such as audio, MIDI, etc.. On the contrary, the composer may conceptually relate several musical material as a group, which is sonically incompatible. In order to arrange them over the sequencer, he has to dispute the interrelationships among material and instead group them into tracks of sonically related material. As a result, the musical semantics are missing over the track-based interface. If the composer wants to repeat a motif composed of material over several tracks, for example, he has to move about the tracks to retrieve the related elements, and again group them according to their sonic features, rather than his musical concept. It makes structuring music less intuitive over the tracks.

1.2 Objectives

We will develop a system tailored to support creative workflow.

We would like to develop a system to overcome this inadequacy. Based on the findings of music cognition and previous research of composition workflow2.3—"Creative Workflow: How do Musicians Compose Music?", Schneiderman [2002], we design the system to allow the composers to be able to visualize their musical concepts. Meanwhile, with this system we want to explore the essential aspects of supporting composers to capture, organize, and evaluate their ideas.

Some significant questions and goals we would like to achieve are as follows:

1. How to visualize composers' musical concepts?

2. How to allow composers directly access to the content through their musical concept?

3. How to assist the tentative process of musical composition, with the consideration of musical semantics?

4. How to reduce the cost of idea input?

1.3 Rhythmic Composition as a Vertical Prototype

Users' musical concepts are considered in multiple levels by this vertical prototype.

Music could be viewed in different levels 2.1.4—"Three Levels of Perception: Macro/ Meso/ Micro-level". In order to demonstrate how the musical semantics could be considered by the system in different levels and how these levels relate and empower each other, we develop our prototype vertically with the focus on rhythmic composition. Furthermore, we incorporate the circular interface into the pattern editor, in the hope to expand the dimensions of creativity as well as make the looping concept easy to understand.

1.4 User-Centered design

To develop our system, we have adopted the iterative design approach, the design-implement-analysis cycle, as shown in figure 1.1. The cycle starts with an idea as the initial design, a user-survey as the first analysis, and a low-cost storyboard as an initial implementation. The implementation technique upgrades in the following iteration as the design has a clearer contour. The design of the system is refined iteratively to reflect the analysis of last iteration.

The design of DISCO has significant changes with each DIA iteration.

Initially, we conducted field interviews to gain more understanding of the composers and discuss our idea of the system with them. Then we performed several iterations of the DIA cycle. With each iteration we gained significant findings and better understanding of the actual requirement. Appendix A demonstrates how the final prototype looks dramatically different from the original idea of the storyboards.

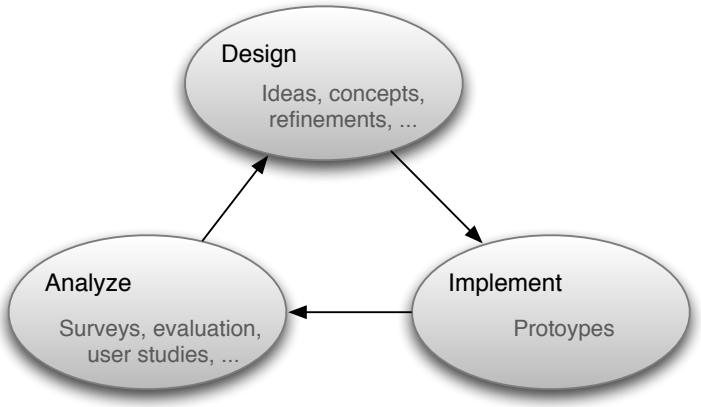

Figure 1.1: The DIA cycle

1.5 Thesis Structure

Chapter 2. *Background* covers how humans perceive sounds as music from cognitive science's point of view and elaborates the creative workflow in the music area.

Chapter 3. *Related Work*. The first part describes the systems that integrate a circular interface as a music composition tool. The second part shows the systems that share similar goals with us. We explain what we have learned from their work and how our system differs from theirs.

Chapter 4. *Design* elaborates the scope of our system as well as how creative workflow is supported by it against the typical approach.

Chapter 5. *Field Interview* describes the findings we have acquired by observing the composers at work.

Chapter 6. *Storyboards* demonstrates how we envisioned the creative workflow could be supported by our system and shows the many revisions after analyzing the prototype.

Chapter 7. *Paper prototype* We focused the design on showing the required interaction for making patterns.

Chapter 8. *Software Prototype* describes the design and implementation detail of the working prototype as well as its evolution after the evaluation phase.

Chapter 9. *Final Evaluation* shows the applicability of the system in different perspectives based on the feedback gained from the composers.

Chapter 10. *Summary and Future Work* summarizes the achievement of the work so far and the outlook based on the current implementation.

Chapter 2

Background

"Music is humanly organized sound"

—John Blacking

2.1 Semantics of Music

In order to design a musical interface, which is effectual for the composers to create music, the way people think about music should be considered.

The term musical semantics refers to the interrelationships among the musical elements we perceive; as stated in Abrams et al. [2002], they connect objects that relate by a certain functions, such as an expressive curve and a musical phrase. The ground concepts that formulate the musical semantics are explained as follows.

Musical semantics connect objects that relate by a certain functions.

2.1.1 Biological and Cultural Constraints

According to Kühl [2007a], Tulving [1985] and Bregman [1991], music is merely an audio stream, but structured in human fashion. The structure is based on the biological nature of human being. Some examples are as follows:

The biological and cultural constraints frame how we think about music.

1. Perception of simple pitch ratios (octaves, fourths and fifths)

2. Regularity extraction

3. Categorization of notes in scales of 5 to 7 steps

4. Perception of melodic contour

5. Group formation

6. Meter as evoked response

Music is created based on the inherent constraints of the creators and the social influences they absorb.

Furthermore, based on the viewpoints of cognitive science Kühl [2007c], the conveyed meaning of music may not always be shared across different cultures in that the perceived information is further influenced by the social context. People of the same culture have many common social conventions and therefore tend to react to sounds in similar ways. The creators of the music, similarly, compose their music based on the structure.

2.1.2 Musical Chunks

When our brain receives audio streams, in order to make sense of them effectively, it organizes the input in chunks. These chunks convey non-concrete information and are further processed by the human mind as gestalts.

2.1.3 Musical Gestures

Musical Gestures are further blended to make a song narrative.

Gestures are a rich form of gestalts. They combine auditory information with implied visual information, somatosensory information, and emotional information. When composing music, rhythmic information, notes, and sound patterns are organized in gestalts. These gestures are further organized in segments in order to make a song narrative.

2.1.4 Three Levels of Perception: Macro/ Meso/ Micro-level

The musical chunk or gesture represents a mesolevel of cognitive organization Kühl [2007b]. It is referred to be the generic level we interact with the world. From here we can look into the micro level to examine the detailed attributes of music, such as volume, or timbre. Or step back to have the macro view of music, where chunks are grouped in segments and convey high-level definitions of musical material. Most music software focuses mainly on supporting the activities on the micro level, losing sight of where the creation activities actually occur.

Meso level is where most creativity occurs.

2.2 Cognitive Styles and their Correlation to Composition Styles

Each Human brain is wired differently; some prefer images while the others might prefer words. The tendency of accessing information with some particular strategies reflects the distinct cognitive styles found in different individuals Riding and Cheema [1991]. For example, verbalizers prefer to work with things in written form, while imagers tend to perceive things visually. This tendency also reflects on how music is created differently:

Cognitive styles reflects how music is created.

2.2.1 Composition Styles

Barry Eaglestone and Carter [2007] discovered composition approaches can be generalized into two categories: Composition Styles Refinement-based approach The composer creates the overall structure of the composition first, and then refines it.

Composition Styles Synthesize-based approach The composition emerges gradually as the composer explores ideas with the audio material.

The majority of the composers are refinement-based and unsatisfied with their software tools.

Interestingly, there are some clear tendencies shown within these two groups. Refinement-based composers tend to be the majority, and are mostly unsatisfied with the music software they use for composition. Besides, refinement-based composers are mostly imagers and the synthesize-based verbalizers. It suggests that the music software designers should have more concerns of the refinement-based approach.

2.3 Creative Workflow: How do Musicians Compose Music?

The composer's focus shifts constantly while composing music.

Creating music is an extremely intricate process. The composer's focus is shifting frequently between different levels: pondering the song structure on the macro level, or the note arrangement in the micro level. A new idea might pop up while they are working on another idea. Abrams et al. [2002]

2.3.1 Iterative cycle: Ideation, representation, and evaluation

Creative workflow is an iterative process.

Composing music is a constant iteration of idea catching, developing and evaluating. Normally it starts with a new idea; the composer develops it and judges it Coughlan and Johnson [2006]. Then another idea emerges and the cycle iterates, as shown in 2.1.

2.4 Music Software Tools

There are various music software tools supporting users in different aspects. They could be classified as follows:

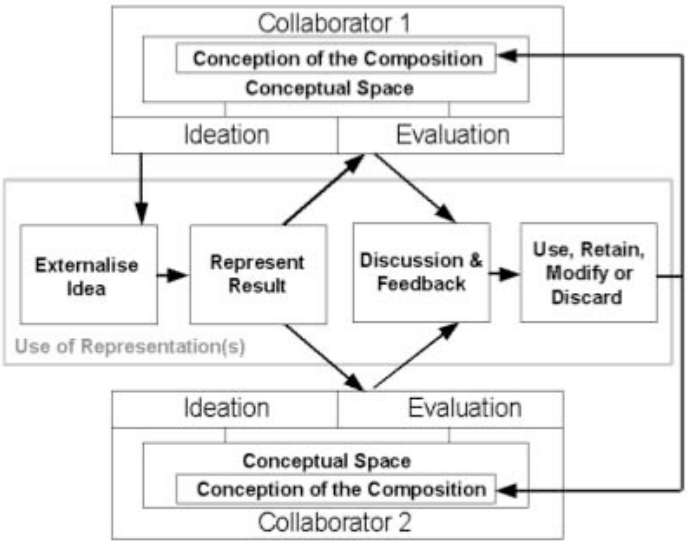

Figure 2.1: The creative workflow of solo or collaborated setting, taken from Coughlan and Johnson [2006].

2.4.1 Tools as a Virtual Music Instrument

The virtual music instruments demonstrate distinct idiosyncrasy

Software instruments are tools for composers to create the musical content. Their interface design is tailored to the different requirement of the sound features they intend to model. Some software instruments borrow the metaphors from the typical physical interfaces, such as Moog Modular[1] 2.2, while the others employ an abstraction layer of graphical controls for the composers to modulate sonic features, such as Operator 2.3. Software composition environments are usually able to integrate external virtual instruments.

2.4.2 Tools as a Composition Environment

Musical elements could be organized by sequencing software tools.

To allow composers to create songs on the computer, typically the environment is geared with a time-based linear sequencer to offer an interface for the users to record, edit, and arrange their musical materials. Currently there are

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

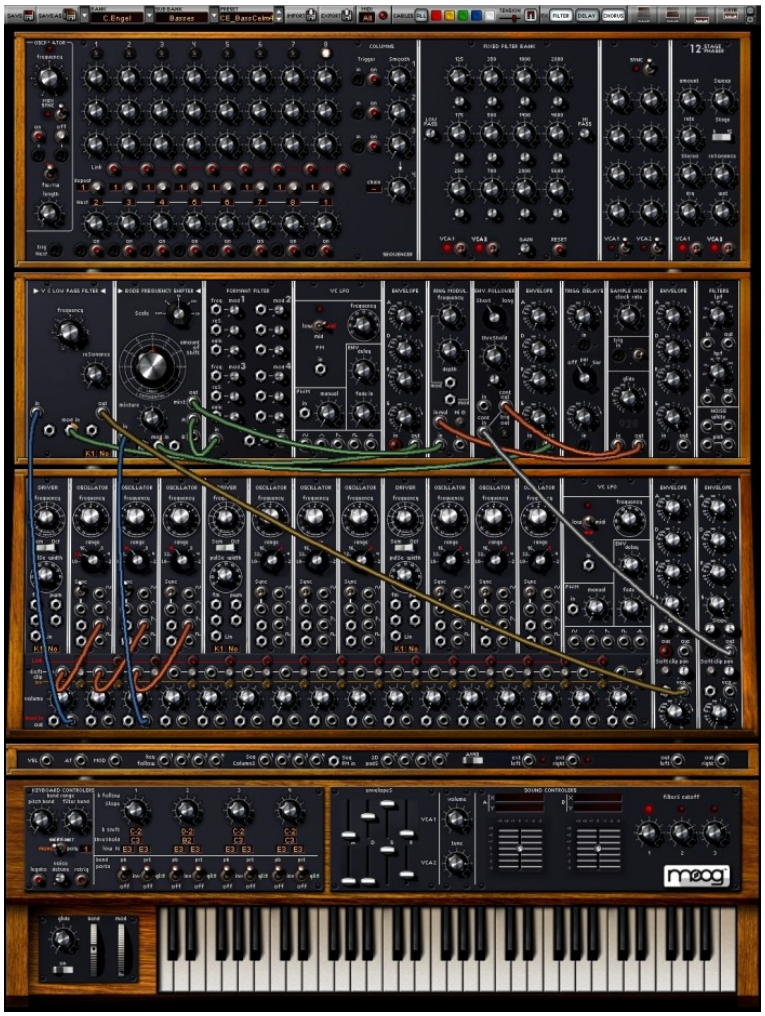

Figure 2.2: A software simulation of Moog Modular.

two kinds of music software sequencers that are commonly used:

Track-Based (Music production, mixing tools)

Track-based environment is geared for manipulating preconceived musical material

Music software in this category is fundamentally similar to a mixer and a multitrack tape recorder. They are equipped with extra features that are exclusive to the digital medium. Preconceived materials are recorded into parallel tracks, which make it easy to have an overview of each single

source medium. For example, the composer can work solely on the track of bass, listen to its continuity, and edit its flow of velocity through the whole song. Professional studios use such software widely for film or music production. Pro Tools[2] 2.4 falls into this category.

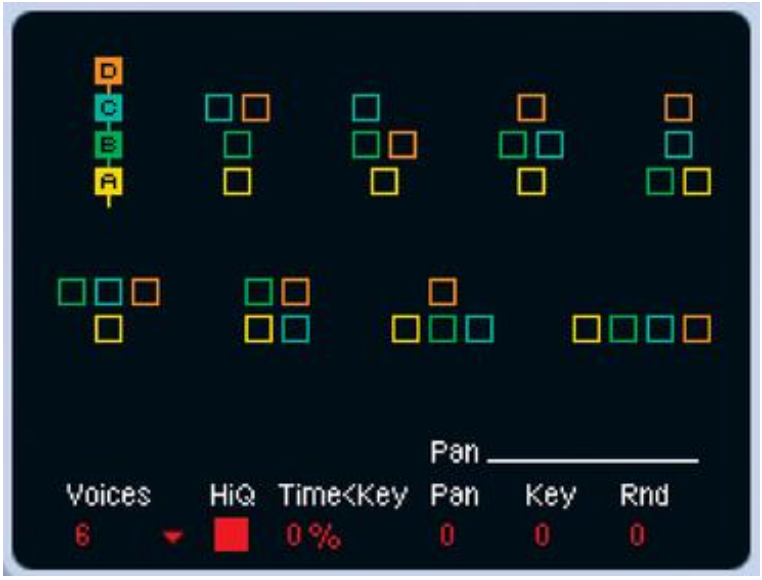

Figure 2.3: Operator, a virtual instrument from Ableton Live.

Pattern-Based

Pattern-based environment is novices friendly.

Pattern-based environments allow musicians to create music in pieces of pattern. Normally they are equipped with a step sequencer and a piano-roll to allow users create rhythms and melody. Fruity Loops[3] 2.5 is a representative system of this category.. To assemble the musical pieces to a song, the users arrange the pieces into a play list. They are beloved by novice users as the concept is easy to understand.

[2]http://www.digidesign.com/
[3]http://flstudio.com/

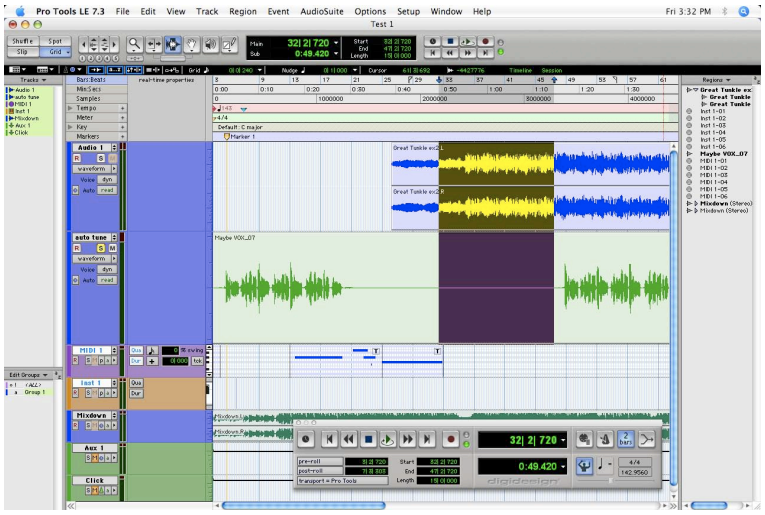

Figure 2.4: Pro Tools, a track-based composition environment.

Figure 2.5: Fruity Loops, a pattern-based composition environment.

Mixture

The system blends the typical approaches.

Take the music software Ableton Live [4] for example. It is a mixture of the above kinds of environment, intending to blur the separation between recording, composition and performance M. Duignan and Biddle [2004]. It offers the typical track-based sequencer as well as the ability to create music in pattern-based manner.

[4] http://www.ableton.com/

Chapter 3

Related work

3.1 Desktop Composition Environments

There are numerous software composition tools commercially available. Here we demonstrate only the representative systems and the alternatives from the previous research, which are aiming at assisting creative composition.

3.1.1 Hyperscore

Hyperscore allows the user to create music without any musical background. Musical features are represented by color, shape, and line texture.

HyperScore Farbood et al. [2004] is music software that allows the user to compose a piece of music without any musical training.

It represents a level of abstraction of the musical features by graphical elements such as color, shape, and line texture. The 2-D sketch window 3.1 represents time and pitch, which conveys what is happening structurally in the song. The user draws different ”motifs” on separate motive windows to produce short phrases, which could be further blended in the sketch window. Furthermore, in order to support the user with no musical background, HyperScore takes functional harmony [1] into consideration. By taking

[1] Functional harmony is a term used to describe music where changes in the predominate scale or additional notes to chords are explainable by

the "Harmony line" the user draws in the sketch window, the system analyzes the ultimate harmony combination, and avoids the occurrence of discord. Therefore, as the user blends different phrases, he focuses on the high-level representation of the song without the distraction of the theoretical concern.

Abstracted control makes it easy to change musical features, but precision is forfeited.

HyperScore encourages the users to build music in short patterns. In the composition phase, it hides the detail from the users by replacing the patterns with color stripes. Harmony lines offer users an abstracted control to change the musical features. Nevertheless, because of its abstracted nature, it is hard for the users to access the low-level information to perform a specific sound modification. Our system adopts a different approach: while we also allow the users to edit their musical concepts in the high level view, they are able to "zoom-in" to the low-level view to make precise modification.

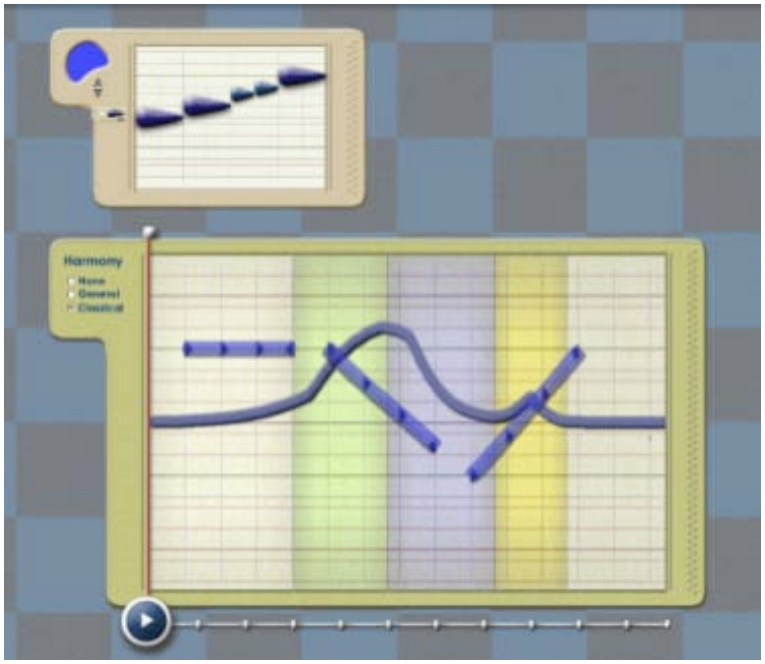

Figure 3.1: The blue lines drawn in the sketch window (down) are associated with the motive window on the left side.

their place in stabilizing or destabilizing a tonality.

3.1.2 QSketcher

QSketcher is an software environment specialized for making film music.

QSketcher Abrams et al. [2002]is an environment for composing music for film. Based on their observation that current tools are merely aiming at the support of realizing preconceived ideas, they developed a tool that could assist composers at the early stages of creative process. As shown in figure 3.2, they propose the creative workflow: capture, organize, manipulate.

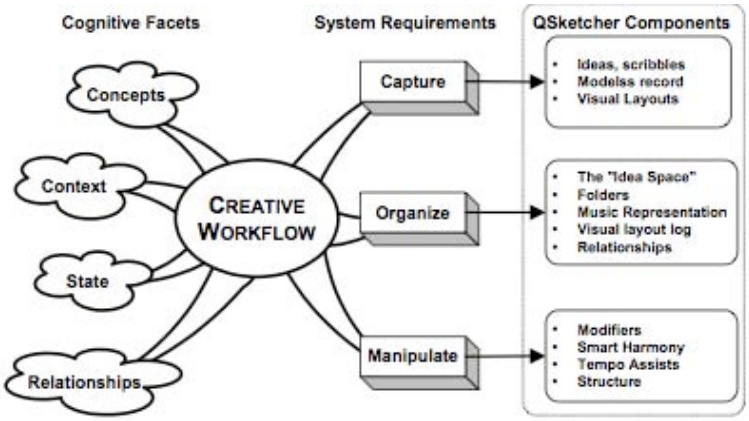

Figure 3.2: The system architecture of QSketcher against the creative workflow.

Idea space intends to organize the composers' ideas semantically.

As the creativities may emerge in various forms, such as hand drawing, recording, or sound clips, etc., they provide composers various methods to capture them. Only few models are introduced in order to avoid distraction. The "idea space" 3.3 is the work surface where all kinds of ideas are organized. In order to visually present how composers mentally relate one idea to another, "Ancestry links" connect related motifs, such as recurring thematic elements. It makes a solid binding between musical elements, which the typical tools are failed to achieve.

Ancestry links are not integrated into the song structure and only show the interrelationships of the material verbally.

Ancestry links 3.4 are beneficial for organizing the musical material, but they are isolated from the related musical concept; their semantics are missing and therefore cannot be accessed effectively. Furthermore, the black strips do not convey any information about their individual musical features, which makes them even harder to use. Our system

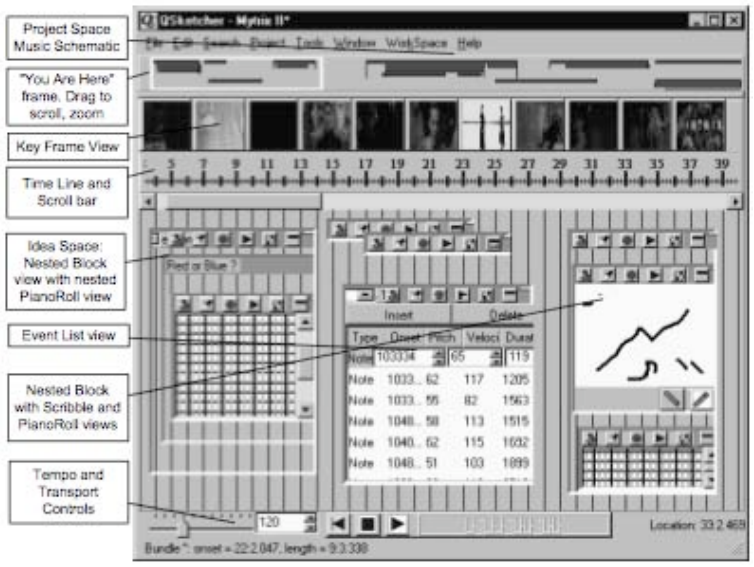

Figure 3.3: The interface where all ideas are organized.

Figure 3.4: Ancestry Links

intends to address this issue by integrating this concept into the high-level representation of music, and visually propagating the influence of the ancestor to the offspring as a visual cue.

Though they endeavor to make the system reflect composers' mental model, the supports are mainly verbal. For example, "Database palette" offers structural view to access the musical elements. All materials are stored here and could be organized in folders of arbitrary labels. As imagers are dominant in composers' cognitive style (see 2.2.1—"Composition Styles"), the verbal palette cannot satisfy the majority.

3.1.3 CyberBand

CyberBand demonstrates a scaleable interface by using Modifers; Intricate controls are masked from novices.

Wright et al. [1997] CyberBand 3.5 is a pattern-based composition environment. It demonstrates a multi-level approach that can suit to composers of all level. Novices can compose music in the shallowest manner – arranging and combining the existing patterns; however, as they gain acquaintances of the system, they unfold the finer controls to achieve more precise composition. This multiple levels concept is supported by CyberBand's central idea: "Modifers". Instead of changing the original content directly, the users lay modifiers over the original content. Modifiers range from sound modulation to changes of the rhythm. There are predefined modifiers while free for users' further change. Modifiers could be stacked or overlapped, which enable complex and precise controls over the original material. Hence, competent users are not limited by CyberBand's low ceiling design. The modifier metaphor is now pervasively used by commercial software.

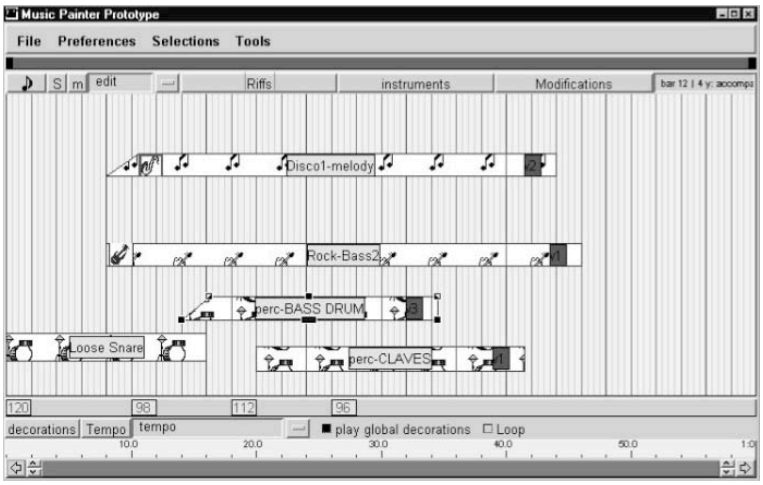

Figure 3.5: A score sheet of CyberBand

Multi-tracks metaphor has little importance to an interface for composition Abrams et al. [1999]. Take CyberBand for example, there is no tracks metaphor in the environment and therefore each block is a unit; no horizontal relationships are imposed between blocks.

3.1.4 Propellerhead Reason ReDrum

The user data is passively organized, which increases the cost of idea input.

Reason ReDrum[2] 3.6 is a step sequencer drum machine. Its interface provides a consistent focus of control for editing patterns; however, its limited visibility makes it hard to access the content. Furthermore, four banks provide a passive support of pattern categorization. As stated in Abrams et al. [2002], to perform the common activity of expanding motifs, users have to manually copy each pattern to the new location, which will impede the creative process due to its low effectiveness Coughlan and Johnson [2006]. We would like to improve these common issues found in most commercial software by introducing another workflow, which will enhance the cohesion among musical elements so as to reflect the composers' musical concepts.

Figure 3.6: Reason Redrum is a software drum machine.

3.1.5 Ableton Live

Ableton Live is a loop-based software music sequencer. It contains two views called "session view" for creating the content, and "arrangement view" for song construction at liner time lines.

[2] http://www.propellerheads.se/

A New Dimension enhances usability

Typical pattern-based sequencers convey only one-dimensional information (see chapter 2—"Background"), as the musical materials are only grouped according to their occurrence of a specific pattern. The interrelationships of the musical elements across different patterns are therefore missing. Ableton Live instead employs a view conveying two-dimensional information where patterns could be formulated horizontally, and the properties of the instruments are distinguished vertically. Looking into the musical elements within one vertical track, they tend to be cross-referenced, inter-changeable, and organized in a progressive order, even though there is no time information conveyed in the session view. This concept proves to be a success as it is favored pervasively by musicians of all level.

The interrelationships of the musical elements are demonstrated vertically.

Vague relationships between session view and arrangement view

As mentioned in Abrams et al. [2002], current tools seem to support composers visualizing the mental maps of the relationships of musical elements, while in reality the interrelationships are subject to be broken due to any modifications from the composers. Take Ableton for example, the behavior of the session view and the arrangement view are not in sync. When the user records a pattern from the session view to the arrangement view, the arrangement view makes a "screenshot" of the current pattern. As long as the user changes its original content in the session view, all counterparts in the arrangement are left as orphans. With the time, faulty and confusing information increases.

Vague connection leads to the unexpected behaviors of the system to the users.

In our system this issue will be resolved in that the high-level view is closely bound to the low-level, meaning that all changes on the both levels will reflect to the other. It assures the users predictable behaviors of the system.

We demonstrate the consistency of all levels.

Figure 3.7: Session View of Ableton Live

3.1.6 Conclusion

Our system centers on visualizing the composers' musical concepts.

While most commercial sequencing software are geared to support composers with preconceived ideas Abrams et al. [2002], CyberBand, HyperScore, and QSketcher are aiming at supporting composers' creative process. Nevertheless, the interrelationships of the musical elements are not well considered in these interfaces, in spite of the effort made from QSketcher in the verbal level.

Aiming at this issue, we would like to propose an interface that centers on representing the composers' musical concepts. Divided into two levels, the system presents a clear view that reflects the high-level definition of the song, and the low-level structure to enhance the cohesion of the musical elements.

3.2 Circular Interface for Musical Composition

3.2.1 Jam-O-Drum

Community drum circle metaphor for people to improvise music together

Jam-O-Drum Blaine and Perkis [2000] is a seven-foot diameter circular table with several drum pads embedded around the rim. By using the metaphor of the community drum circle Hull [1998], where people form a circle and play music, the project intended to explore new ways for people to improvise music together.

The drum pads comprise physical controls and visual cues to the users.

The drum pads are equipped with force sensor for velocity control, and piezoelectric sensors in order to trigger the back-end audio engine by hitting the pads. Its game-like visual interface design offered several different strategies to allow the participants create drum rhythms by taking turns or collaboratively. Basically it cues the user when to play, and provides immersive audiovisual feedback to enhance the joy of interacting with the table as well as help synchronize the tempo among all users.

Automation facilitates novice users while impedes expression.

To guide novice users, the back-end audio engine quantizes the rhythms being produced and automates the volume with a predefined envelope. While it gave the novice users the feel of satisfaction with little effort, the experienced users felt the system limited their expression.

This project appeared to be a success in the social aspect of music collaboration. People are less concerned about performing in public because of the engaging interface design.

3.2.2 Jam-O-World

Jam-O-World is a collaborative musical game over table top.

Jam-O-World Blaine and Forlines [2002] is a redesign of Jam-O-Drum as a musically enhanced game. A MIDI drum pad inside a turntable disk is integrated into the original design. Each player uses a turntable to control one of the concentric rings in the centre of the table 3.8. In order to get a virtual ball to fall into the centre from the circular

maze, the players have to devise a pathway collaboratively by turning the disks around.

Figure 3.8: CircleMaze on the Jam-O-Whirl.

The turntable disks enable the users to reassemble the musical materials.

The game creates another dimension of interaction through music, as the background music can be set in motion when the players rotate the disks. Also, the players can hit the disks to send a midi signal to the back-end audio engine, which will in turn trigger the related audio samples. Furthermore, in order to avoid the chaos of arbitrary combination of the musical elements, the background music tracks exist in two and four bar phrases designed to be 256 recombinant in interlocking configurations. When the players rotate the disks in 90 degrees, the related music loop would be offset to the next incident of downbeat. As a result, the players are less conscious of their influence of turning the turntable than hitting the pad, since their influence on the background tracks is highly restricted to ensure the harmony of the whole output.

Jam-O-World demonstrates the versatility of using circular control.

From Jam-O-World we see the potential of using circular interface to manipulate the content of music. Since music has the tendency of being symmetrical and reoccurring, the musical components still sound pleasing and complete even after being offset. As demonstrated in Jam-O-World, rotating the turntables can generate 256 different combina-

tions.

3.2.3 DaisyPhone

DaisyPhone is an instrument for remote real time collaboration.

DaisyPhone Bryan-Kinns and Healey [2004] is a circular step-sequencer used to support group improvisation when the participants are not present physically in the same location.

Circular interface as a shared instrument

Circular interface represents the looping nature of music.

To find the form for their shared instrument, they observe novices using typical music sequencer. It turns out that the loops they created are subject to be abrupt from the end to the start, which is mainly due to the inadequate representation of the looping concept by the linear sequencer. Aiming at demonstrating the cyclical nature of looping music, they opt for presenting music in a circle. The circular interface of Daisyphone consists of 45 notes, which could be set and unset and are played when the gray arm passes by the nodes.

DaisyPhone uses color and the radius of a circle to represent musical features.

4 spokes (see figure 3.9) could be chosen from the centre, which represent the four instruments: piano, bell, glockenspiel, and percussion. The volume of the instrument can be adjusted by changing the color of the spoke. The distance of a node to the center represents its pitch. Also, another smaller circle containing 20 nodes is the session selector, which enables the users to have multiple sets of creation in real time.

The GUI of DaisyPhone is beneficial for idea formulation.

Furthermore, in order to show that their design of circular interface fits to the criteria of supporting idea formulation of music, they reflect on Tabor's musings A space for half-formed thoughts[3] ,which depict an imagined space for manipulating and formulating ideas half-consciously. Firstly, DaisyPhone focuses on the spatial metaphors on the two dimensional representation of the pitch and sequence of notes. Secondly, it offers multimodality by representing

[3] http://flow.doorsofperception.com

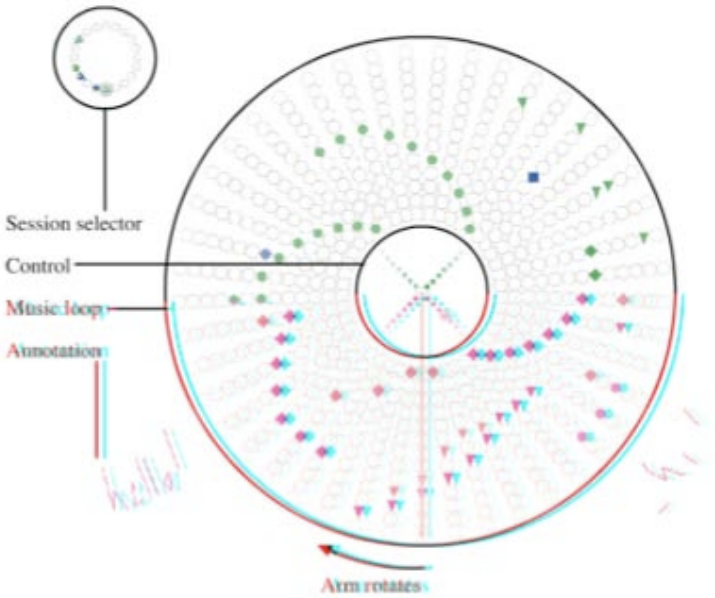

Figure 3.9: The GUI of DaisyPhone.

both sound and graphics of what the user is currently interacting with. Thirdly, the interface provides a way to grasp specific patterns even if the overview is messy. Finally, the messy nature of the interface provides variable focus for the user to experiment.

The users created more continuous loops with DaisyPhone.

According to their user studyBryan-Kinns [2004], the concept of a circle behind DaisyPhone can be quickly grasped; the created loops demonstrated great continuity, where the tendency of dividing the loop into four quarters, can often be found. After a short term of using DaisyPhone, the users were able to create tuneful compositions.

The intricate interface design can hardly lead to logical composition.

Due to the intricate interface design, people tend to draw patterns randomly without musical consideration. On the one hand, it is a test bed for users to formulate ideas. On the other hand, just after several minutes of the group usage, the interface tends to break down since too much information is present and unexpected modifications to the nodes arise from remote users.

We have a pretty similar approach of using a circular inter-

face to that of DaisyPhone; however, as their focus is collaboration, the circular interface acts as the ”canvas” of music, where musical elements are mixed together.

Our system offers a clearer overview for creative composition.

On the other hand, with supporting musical composition as our target, we present each instrument in a circle as a ”color block” in a ”pallet”. The users are assisted with a clear overview of the elements they apply as well as the ability of reconfiguring and reusing musical elements independently in the other set.

3.2.4 ReacTable

ReacTable is a modular instrument for group improvisation.

ReacTable Jordà et al. [2007] is a shared music instrument for group music improvisation. A camera is installed under a table, which will track the position of fiducial markers attached to the physical objects. The back-end engine processes the images captured by the video camera and then produces the accompanying visual information for the musical objects on the desk. Depending on the attached definition of the musical objects, they may transform the sound attributes, change the metronome, add audio samples, or emit recurring pulses. Musical objects can be added, removed, and assembled in arbitrary manners. Immersive visual cues between the musical objects intend to inform the users of the sonic information they convey as well as the interrelationships among the objects.

Our system simply the phase of conceptualization with a structural view.

We are similar in the way that we both employ circular step sequencers to trigger sound in a reoccurring manner; however, we have very different perspective in that their improvisation approach presents the output as a stream, meaning all musical objects are blended to generate one singe sound output. While we intend to simplify the formulation (conceptualization) of the intricate content as a structural overview, where the ”musical chunks” 2—“Background” are accessed. The disks do not only used to trigger sound clips but also employed as glue to enhance the cohesiveness of the musical material.

3.2.5 Some other commercial circular interfaces

The commercial circular interfaces are not integrated into the composition environments, and therefore increase the cost of creative input.

There are some commercially available circular interfaces for rhythmic composition and sound modulation, such as Atomic[4] 3.10, Replicant[5] , and GrooveMaker[6] . They are all designed for editing one single music loop, equipped with versatile functionalities to modify sound attributes. While they intend to support the users' creative activity in detail, they also hinder the user from reaching their creative goal, because they are merely applicable at a specific point of the composition process and cannot be integrated into the holistic song structure to convey the musical semantics. In order to continue composers' tentative process Coughlan and Johnson [2006], the user needs to apply another host to manually access these external components in a low-level manner. The creative activity is therefore distracted. Our system organizes the users' inputs to the disk interface semantically and further reflects their musical concepts on it.

3.2.6 Conclusion

From the aforementioned projects, such as DaisyPhone, Jam-O-World, we believe a circular interface outdoes the linear ones, as it represents the concept of reoccurring nature of music; it offers a low ceiling entrance for novices as well as versatility for the advanced users.

The way circular interface is employed in our project is unique in that we integrate the disks into the whole composition environment, where the composers' musical concepts are considered and will be visually presented on the disks.

[4] http://www.algomusic.net/
[5] http://www.audiodamage.com/effects/product.php?pid=AD013
[6] http://www.groovemaker.com/

Figure 3.10: Atomic, a circular step sequencer for making drum rhythms.

Figure 3.11: Replicant, a delay-based loop editor.

Chapter 4

Design

This chapter outlines the scope of the system. We will start by demonstrating how creative composition is currently performed in the track-based environment, and the alternative approach by using our system. In fact, track-based interface is originally not geared for creative workflow Abrams et al. [2002] while it is generally regarded as the standard tool for all aspects of music composition. Therefore, we shall briefly compare the track-based creative composition against our proposed approach, to show how the composers can benefit from using our system. Finally, we will elaborate the design decisions we made for having the focus on rhythmic composition with circular interface.

4.1 The scope of the System

• To allow the users to create, edit, and arrange the musical material, which will be visualized to reflect their musical concepts.

• To organize the musical material in the way that could be used effectively.

• To introduce an interface that reflects the concept of looping music.

4.2 Creative Workflow of Track-Based Environment

Before we elaborate the design of the proposed system, we shall give a brief view of how creative activity is performed in the typical track-based environment. This workflow is based on the result of the field interviews 5.3—"Result" we conducted, where we observed many composers creating music on the track-based interface. As shown in figure 4.1, the interface represents a flat data structure, as the data can only be distinguished by the name of the track or the elements. To interact with the interface, the composer always has to access the low-level data

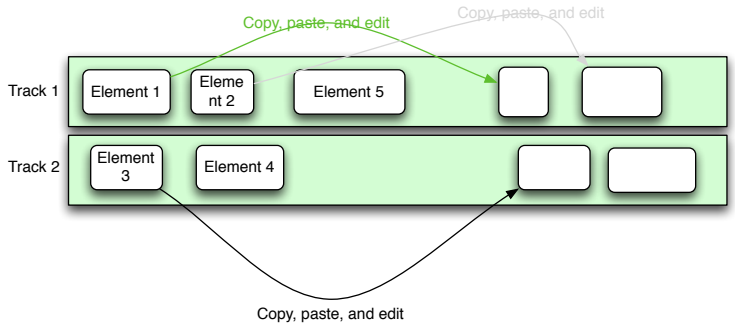

Figure 4.1: Using typical track based interface for composition

Figure 4.2 shows an example outcome of the composition. Low-level information makes it hard to spot the desired data.

4.3 Creative Workflow of the Proposed System

4.3.1 Propagation of Variation

The design intends to let users access musical material based on chunks.

The design is based on the finding of the field interviews 5.3—"Result"- creating music is an evolving process. As

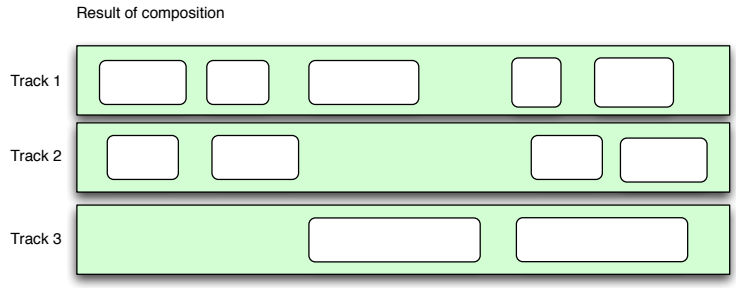

Figure 4.2: The overview of the composition

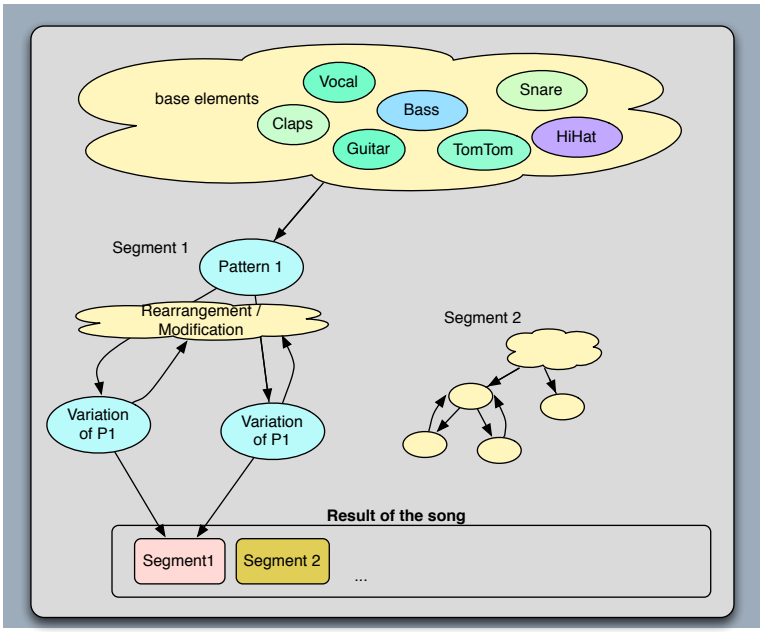

Figure 4.3: Musical composition is an evolving process.

shown in figure 4.3, a segment might be composed of several similar patterns, which are actually the outcome of rearranging and modifying the same source of musical elements. This tendency also reflects the idea of ”musical chunks” (see chapter 2—“Background”) creativity occurs mostly at the level of accessing the grouped musical elements; the chunks are blended to form the composer’s musical concepts.

The system is divided into two layers.

The system is divided into two layers: High-level and Low-level view. High-level view demonstrates the composer's abstract definition of the song structure, from which the composer can "zoom-in" to the low-level view to manipulate the detail of the song. The workflow is illustrated as follows:

The system enables the users to create content effectively and logically.

The composer starts with an initial idea (see chapter 4.4). He manipulates the contents to produce the initial motif. Then, when the composer captures an extended idea and would like to realize it (which is common with the refine-based composers, see chapter 2—"Background"), he can specify this intention on the high-level view. As he shifts the focus to the detail, the system recognizes this intention and lets the composer create this variation by modifying a duplicate of its original content, as shown in figure 4.5). In this way he can concentrate on working out the new concept over the original without the effort of technically transferring the data. In addition, since the detail is encapsulated in the high-level view, the composer can interact with the interface in terms of his own musical concepts, and "zoom-in" to a specific detail only when needed.

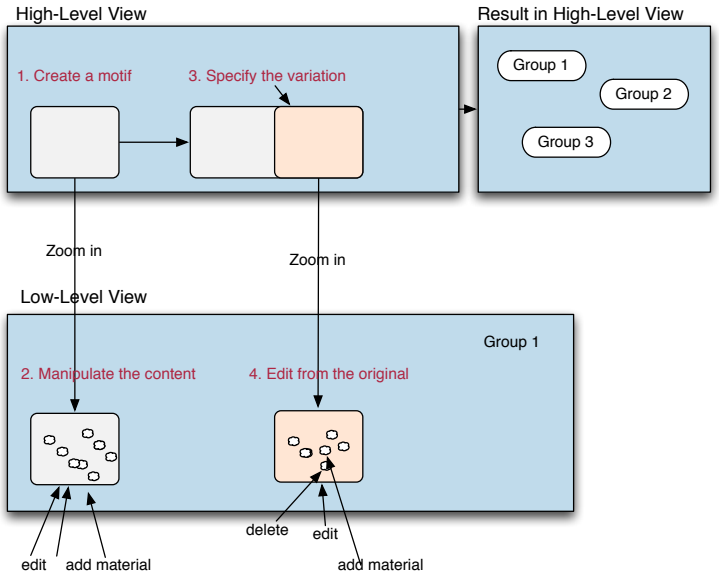

Figure 4.4: How musical elements are created with our system.

4.3.2 Grouping Functionality

Musical elements are grouped and accessed based on their interrelationships.

Furthermore, the system groups the chunks of musical elements based on the composer's musical concepts. The composer can retrieve a group of patterns by accessing his own definition of the structure, instead of the raw data. With these groups of chunks, the composer can compare or search the elements in context of their interrelationships, as shown in figure **??**.

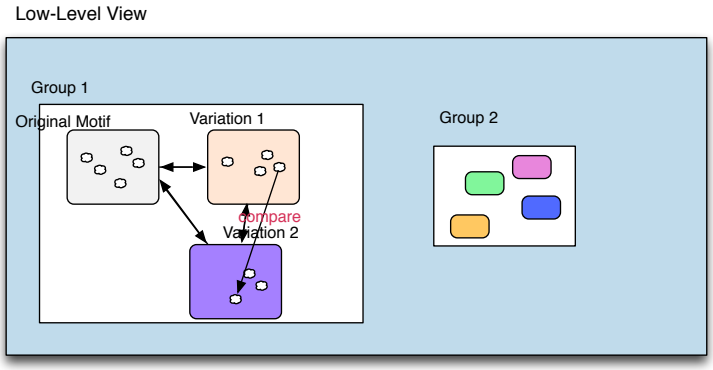

Figure 4.5: Musical elements are grouped logically in the low level.

The system provides a playground that assists the users to organize their tentative ideas, and consistent behaviors in these two layers.

On the other hand, when the user wants to experiment with the existing material (which is common with the synthesize-based composers, the low-level view provides a playground for ideation (see figure 4.6) as all elements are organized based on their interrelationships. Moreover, all changes will be reflected to the high-level representation without the common issue of the blurred representation of the composer's mental link (see 3.1.2—"QSketcher" and Ableton Live 3.1.5—"Ableton Live").

4.3.3 Vertical Prototype

The proposed workflow could be applied to support creative composition with other instruments.

The proposed workflow is independent of which type of instrument is being used. The composers can freely manipulate their musical concepts over the high-level interface. The input to the system could be both external audio and

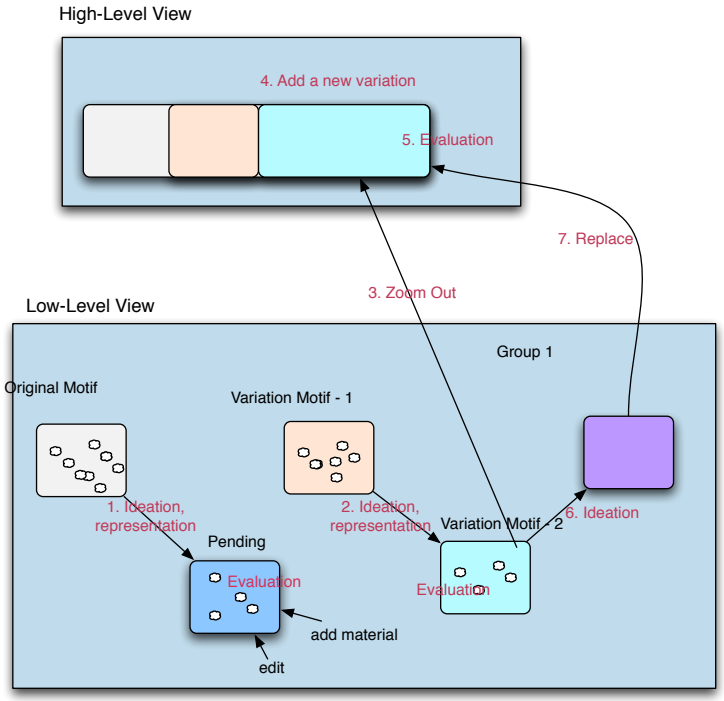

Figure 4.6: The system keeps all tentative ideas logically so that the composers can access them easily in the later stages.

MIDI. Nevertheless, we believe the system can further reduce the cost of creative input as long as the creative input interface is integrated, meaning that the input interface also takes the musical semantics into consideration. Therefore we would like to develop a vertical prototype by devising an input interface tailored to the proposed workflow, as mentioned in chapter 1.3—"Rhythmic Composition as a Vertical Prototype". To set up a focus, we consider the result from our chapter 5.3.1—"Similarities in Composition Process". There is a higher tendency for the composers to use software tools to program rhythms than melodies. On the one hand, some of the composers are very often annoyed by the limitations of the tools for making rhythms. On the other hand, rhythms are cyclic and shall be easily automated. Nevertheless, they often complain about the clumsiness of these software tools for making rhythms. Therefore, we set our focus of the input interface on rhyth-

mic composition, in that we believe it will alleviate a great amount of composers' workload.

We employ a circular step sequencer for rhythm manipulation for several reasons. Firstly, a step sequencer provides a simple concept of using normally 16 buttons to represent a pattern, with each button being 1/ 16th of a measure[1]. This idea is easy to be grasped by the novices but also competent for making rhythmic music. Secondly, from the field interviews and the previous research DaisyPhone we notice the clumsiness of manipulating looping music in the linear time line. According to the survey conducted by DaisyPhone (see chapter 3.2.3—"DaisyPhone") such an issue could be improved when the cycle is presented as a circle. Also, Jam-O-World 3.2.2—"Jam-O-World" demonstrates how easy it is to reassemble the musical material by turning the disks around. Thirdly, when a step sequencer is wired as a circle, a measure could be easily divided into equal portions, as we learnt by using a clock. Lastly, compared with the liner sequencer, circular form adds one more dimension for the musicians to explore the 'geometry' of patterns, such as a pattern of triangle, for example. Circular form introduces a new approach of making music visually.

The solutions to the minor issues such as the mechanical rhythms generated from the step sequencer will be discussed in chapter 10.2.5—"Expressiveness of Rhythms". With this prototype, we focus on investigating how to make the input interface follow the composer's creative workflow so as to make the iteration of ideation, representation and evaluation easier to perform on the desktop.

4.4 Conclusion

By focusing on this very specific aspect of rhythmic composition, we try to find out how to support the composer's creative workflow. In the future we can employ this model to support the creative workflow for other instruments (see 10.2.1—"Pluggable").

[1] A measure is a segment of time defined as a given number of beats of a given duration.

Chapter 5

Field Interviews

5.1 Method

In order to create an interface that follows the target users' workflow, we need to understand how they actually work in the field. We design the structure of our field interviews based on the method "Contextual Inquiry Beyer and Holtzblatt [1998]". This is an interview technique that comprises preparation, evaluation, analysis, and design phases. The following is the structure of our interview based on the contextual inquiry principles:

The field interviews are designed based on the method "Contextual inquiry".

Context
Every composer has a distinct manner of disposing their instruments and interfaces, which reflects how they prefer to accomplish the creative activity. We try to understand the context by following the participants performing their task and acquire insights of their workflow.

To understand the context of their behaviors.

Parternership
The interviewer and interviewee play the model of an apprentice/ master during the interview. The interviewee leads the situation and elaborates to the interviewer on how and why he performs a specific action.

We play the role of an apprentice.

Interpretation
During the composition process, we observe their work-

Immediate interpretation of the findings to the users.

flow and methodology of composition. We interpret our findings immediately to the participants to avoid any misunderstanding. After the composition is completed, we analyze the participants' workflow heuristically.

Direct the users to the target of our interests.

Focus
We would like to discover issues on structuring musical material with software tools; however, it is very likely that the participants spend too much time refining a detail sound parameter or building independent patterns. In such situation, we should encourage the participants to create a complete piece of song.

The participants were asked to compose a piece of music.

Task
The interviews were conducted in the participants' familiar environment (mostly in their studio or home). We asked the participants to compose a piece of music with their familiar software tools and no time limit. At the end, we reported our interpretation to the participants to make sure they were accurate.

5.2 Participants

The participants were composed of composers of different levels.

Participants composed of 2 professional, 3 experienced amateurs and 2 novice composers. Their experience of music composition ranged from 1 to 15 years, and they all had frequent access to music software tools. We inquired the professionals in that they may have in-depth viewpoints to the tools they use for years. On the other hand, as professionals tend to be attuned to the interfaces they work with, we also interviewed the novices to discover fundamental issues.

The following is the summarization of the user profile.

5.3 Result

The novice composers tended to be reluctant to use the track-based sequencer.

Each interview took around three to four hours. Some participants presented a half completed composition or some

	Instruments	Sequencer	Music Style	Composition Approach
Participant 1	Hardware, modular, software instruments	Cubase	Techno	Refinement-based
Participant 2	Guitar, software instruments, piezoelectric	Ableton Live	Experimental	Synthesize-based
Participant 3	Software instruments, sampler	Ableton Live	Hiphop	Synthesize-based
Participant 4	MPC drum computer	Ableton Live	Hiphop	Synthesize-based
Participant 5	Keyboard, bass, software instruments	Logic	Electronic	Refinement-based
Participant 6	Software instruments	Fruity Loops	Electronic	Synthesize-based
Participant 7	Guitar, percussions	Garage Band	Acoustic	Refinement-based

Table 5.1: The musical profiles of the participants

musical fragments that they intended to work on, and proceeded to refine them in the interview. The others started from scratch, improvising with their instruments during the interview. Novices spent most of the time building patterns. They were able to demonstrate their musical concept by triggering the patterns manually, but were reluctant to arrange them in the time-based sequences in order to structure the song "technically". Interestingly, none of them really knew the reason why they did not like to arrange their work on the sequencer. "Maybe it will be alright if I did spend time getting used to it", one participant said. On the other hand, the advanced composers appeared to have no complaint about structuring ideas on the track-based sequencer; instead they were more interested in the versatility of the tools.

Experienced composers tended to have a formulated workflow.

Besides, professionals appeared to have a "formulated" workflow; they have a specific order of idea construction that will quickly lead to the completion of a song: "Normally I would start with my drum computer to lay out the backbone of the song, and then…". On the contrary,

the participants with less experiences have more tendency to abandon their initial ideas and create a new song from scratch, or create random ideas which may lead to musical material for different songs.

The virtual instruments analogous to physical objects were less favored.

Some music software lends objects from their physical counterparts as the metaphor for controlling its interface, such as knobs, sliders, while others construct an layer of abstraction for manipulation M. Duignan and Biddle [2004]. Most participants felt easily frustrated with the interfaces that are analogous to physical objects. Take a mixer for example, one participant pointed out that there are so many identical controls on the interface. He always needs to rebuild the mappings between tracks on the mixer and the related instruments whenever he switch from the other interface to the mixer. It costs time and distracts him from constructing ideas.

Some participants also pointed out many features they have never used, and hope they can configure the interface to reduce the complexity of the appearance.

5.3.1 Similarities in Composition Process

As though the diverse tools and music styles the participants accessed, they have some similarities in their composition process.

Interface for creative input

The participants were unsatisfied with the support from their tools for making drum rhythms.

The participants had a certain tendency of choosing physical or software tools for creative input. The composers usually used physical interface for composing the expressive melodies, such as the guitar and the violin, while they used software tools for cyclic music such as drum rhythms or bass lines. They could automate the cyclic rhythms with the software tools and at times add subtle changes to make the flow of the rhythm more dynamical, while it is currently not trivial to perform with the expressive melody line. Nevertheless, many participants complained about the software tools for creating rhythms; it took lots of efforts to organize the composers' musical material with the interface, for example. Moreover, they also had difficulties to retrieve a

specific musical element they created.

Tentative workflow

Idea formulation is an evolving process; they constantly manipulate the same material to create several similar versions. They may change the dynamic of the duplicates, add or reduce the elements used in them. Then they may just apply one of the similar motifs into the song, or chain them in a row. Sometimes the participants simply left one pattern looping in the background and improvised with it, which would result in many unstructured phrases scattered over the interface. The participants spent a long time linking these fragments and refining details.

The composers constantly refined the same element, resulting in many similar versions of one idea.

Iteration

The participants improvise with their instruments to gather ideas, record the ideas with their software tools, and modify the musical elements. The whole composition process could be summarized by the repetition of these three procedures.

Similar procedures were repeated.

Contingency

The graphical abstraction of controlling the sound feature enables the user to roam playfully around the interface. We found that many participants spent a long time toying around with their instruments, without a concrete idea in mind. Sometimes they did not even know the purpose of the control they were manipulating, just intending to be "inspired" by the instrument.

The composers intended to be inspired by their instruments.

Colors as visual cue

Some participants relied heavily on colors to orient themselves at the sequencer view. They may intentionally specify the color of the material to reflect their song structure.

Colors were essential to many participants.

5.3.2 Issues

Music composition is an extremely complicated activity. Hence we noticed many problems related to diverse areas during the interviews. The significant findings to our research are summarized as follows:

There was no effective way to search a specific musical element.

Material Categorization in Literal Level
The user was looking for a specific data in the "library" folder, where musical elements are all stored in a data pool. The user can search through the file name, size, and times of reference (which the user did not realize the definition before I explained). Unfortunately, similar files may come from different segments of the song, while they are labeled in the same manner- bass001, bass002, etc.. Without the context of the material in the song structure as a reference, the composer can only listen to all the files trying to find out the target.

Tracks represent a flat data structure
Tracks are defined by the input format of the instrument, such as audio or MIDI. The participants, especially novices, arranged the material of the same input format in the same track. With the time, the material scatter over the interface without any effective organization.

Musical concepts cannot simply organized based on time.

Defining musical concept on the track-based interface
Some systems tried to help the users categorize their material by dividing the tracks into groups of time blocks. A participant pointed that the context of the material cannot be merely defined by time and therefore the feature is not practical. Besides, when the participant started playing the time blocks in a specific order, the time cursor shifted between the time blocks. The left-to-right metaphor of the linear time line was then broken, which made the participant unable to follow the composition.

Technical details impeded the composers' creativities.

Effectiveness
Coughlan and Johnson [2006] pointed, to support creative activity, the interface should reduce the cost of idea input. The more time they spend on the technical procedure, the less likely they are able to be creative. We noticed that, in order to expand existing patterns, most participants spent far more time duplicating material than actually manipulating the content. Very often they just get disoriented when shifting between different tracks and location.

Looping concept is absent
Many composers work with a pattern-based approach. They often have the need to offset a loop; however, the discontinuity of the liner interface makes it clumsy to perform

such a task.

5.4 Conclusion

The composers need a clearer and logical interface to assist their creative process.

The limitations of the current software tools are clearly shown when observing the composers in the field. In order to reduce the complexity of accessing musical data, musical elements should be organized into levels; the information outside the current level is masked from the user. The top level, especially, should reflect the user's musical semantics. In the low level, the elements should be organized based on the semantics formulated in the high level, instead of being based on terms borrowed from computer science, such as reference times. In this way, musical elements could be effectively reused in the low level, because they are linked to each other in accordance with their interrelationships in the composer's musical concepts.

Chapter 6

Storyboards

Our storyboards demonstrated two scenarios.

We create two storyboards to demonstrate two of the typical situations of using music software tools. The scenarios are intentionally designed to show the specific aim of our system. We would like to see if the proposed workflow does support the users more effectively for their creative activity, and the feasibility of the circular interface for rhythmic composition. Moreover, we also want to discover any overlooked aspects in our system by going through the design with the potential users. Many features of the first prototype appeared to be clumsy and were therefore disposed; however, the prototype had helped us make our concepts more concrete and shaped our design criteria a great deal. Design

6.1 Design

Pattern prototyping

A pattern library is used for storing template patterns. Arrangement view shows the song structure to the user.

Based on the design scope (see chapter 4—"Design"), we intended to make the process of making patterns by two steps: pattern prototyping and template refinement. While the user applies several musical elements intending to create a new pattern, the elements are stored additionally in the associated "pattern library" (see 6.1 view 3). Afterwards, the user can use the musical elements from the li-

brary as templates to create variation patterns. In this way, variation patterns could be created effectively and saved to create a logically related group, which reflects the user's musical concepts. Furthermore, based on the musical concepts created manually by the user, the interface visualizes the song structure in the arrangement view (see 6.1 view 4).

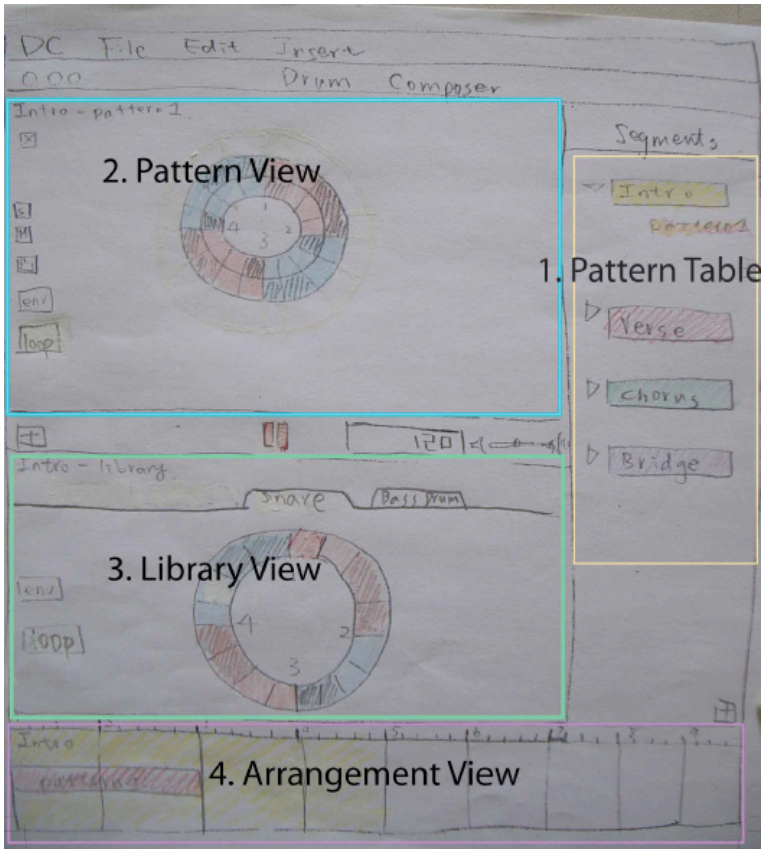

Figure 6.1: One example screen shot of the storyboards.

Concentric circles

Each ring is a subpattern, which could be edited and offset independently.

As mentioned in 4.3.3—"Vertical Prototype", we introduce the circular step sequencer into the system as a pattern editor. To allow the users to have an overview of the musical elements of a pattern, we visualize each musical element as a ring of the concentric circles (see 6.1 view 2). A ring could be independently edited, and offset.

6.1.1 Features

Sound clips take the roles of the musical instruments. Variation patterns are independent from their original.

Patterns are the elemental musical material of the prototype. A pattern may contain individual subpatterns for a number of different instruments. The user adds a set of sound clips (a set of drum kit samples for example) to generate the subpatterns, which are represented by the concentric rings. Furthermore, a pattern may have variation patterns. Variations can have different settings and layout from their original. The following section illustrates the main features of this prototype

1. Pattern Table

Pattern Table lists the template patterns in the first level and the actual patterns in the second level.

This view demonstrates a two-level tree structure of the song. The first level is the abstraction level (see 6.1 view 1). By accessing this layer the user can prototype subpatterns in the Library View [(see 6.1 view 3)before creating any actual pattern as well as specifying the name of the pattern. The second level is the representation level: the user can drop desired element from the library into this level, and further change its musical feature. In addition, when the user adds an external resource into an actual pattern, the resource will be automatically added into the associated library. We intend to encourage the users to create templates for each subpattern, so that they can create the extended patterns effectively by modifying the templates.

2. Pattern View

Pattern view demonstrates all corresponding material used in the currently selected pattern in concentric circles. The user can edit, and rearrange patterns here.

3. Pattern Library View

The library creates a template pattern for each sound clip.

A library is attached to each set of associated patterns to store its musical templates. Whenever an external sound

clip is added to a pattern, a blank template for this clip is added to the library. With a pattern in the pattern view (see 6.1 view 3) selected, the corresponding template patterns are shown in the library view. The user can edit the templates here, or drag a template to a pattern to reuse it with or without further modification (see Appendix A—"Storyboards").

4. Arrangement View

The users can structure their musical material in Arrangement View.

Arrangement View allows the user to arrange the patterns in a linear format. The background color and the label reflect the abstract definition by the user in the pattern view. The time progresses from left to right, which is the standard convention of the typical music sequencing software. Whenever the user creates a new pattern, it will be automatically added into the arrangement view. The user can manually specify its length and the position.

6.2 Analysis

6.2.1 Process

Animation and flying dialogs were demonstrated by "post-it" notes.

Audio and animation are highly involved in the interaction with the system. In order to simulate the real situation, we need to manually demonstrate the required dialogs and mimic the audio and animation feedback from the system. Therefore, we draw the screenshots on pieces of paper and use "post-it" notes for pop up dialogs and moving objects. As we intended to use color as an aid to divide the circle and group the patterns, color is applied occasionally to the drawing to gain feedback about the feasibility of color-coding. In addition, colors are used to highlight the user's current selection.

6.2.2 Participants

The storyboards were evaluated with 9 users. Because we would like to see if the proposed system reflects the practical process, the participants with musical composition experience were chosen: 5 with composition experience over 8 years and 4 with around 5 years experience. They all play traditional music instruments and 6 users have experience with music software. We encouraged them to ask questions and criticize during the session.

The participants with musical experience were deliberately chosen.

6.2.3 Scenarios

The demonstrated scenarios are summarized as follows, for detailed screenshots please view Appendix A—"Storyboards".

1. Create the accompanying drum patterns for a song

In this scenario, we demonstrate how the system supports the user to effectively structure their song by expanding the materials, re-use musical elements, and perform the creative activity as an evolving process.

2. Review the materials after a period of time

In this scenario, we would like to demonstrate how the system logically categorizes the musical materials, so that it keeps the user working on the high-level view of the whole song. When a specific detail is needed, the user can quickly "zoom-in" to examine only the desired information, without being distracted by unrelated data.

6.2.4 Findings

All participants agreed that it would be beneficial if the system could support them expanding their patterns, and classifying the musical material as how it relates them mentally.

The system did not follow their their composition process.

Nevertheless, the workflow of the proposed system does not fit to their composition process.

Through this prototype we had many in-depth discussions with the composers. They pointed out the irrational aspects of the design by comparing with their own process. The feedback is summarized as follows:

Discontinuous workflow

Prototying patterns were unintuitive.

The idea of prototyping in one view, and realization in another view is not intuitive. Some participants said they would never employ the prototyping concept if no one instructs them. Normally they just start with an initial idea and let it evolve. They suggested to get rid of the prototyping step before creating any actual pattern. Instead, let the user create the first pattern and use its content as the template.

Misuse

Loose relationship between the library and the resulting patterns may cause misuse.

The library is intended to encourage the users to reuse the material from the library to create variations; however, the user can create a dramatically unrelated pattern by not reusing the material in the library. The facilities of the library and categorization will then have no impact.

Terminology

Some terminologies were misleading.

Initially we define the name of the Pattern View as "Segment", because we think a pattern could be a segment in the simplest case. Nevertheless, a segment generally contains a couple of distinct patterns. The term resulted in confusion to the participants.

Concentric circles break down as it grows

It was difficult to view multiple subpatterns in concentric circles.

Many users appreciate the concept of presenting loops in circles for many reasons. For example: the content could be easily accessed in one single ring, and the arrangement appears more visual than in the linear step sequencer; however, concentric circles do not convey as much information as the single one. Most participants find it hard to access the combination as a whole, or split it into individual rings. They prefer the layout of parallel linear tracks to concentric circles for a clearer overview.

Automation of the sequencer doesn't work

The users did not expect the Arrangement and Pattern Views to be in sync.

We wanted the arrangement view to reflect the user's current state by automatically adding the newly created pattern in the pattern view to the arrangement view. However, many participants suggested they might just want to try out an idea in a pattern, which will not necessary be applied to the song. They regarded the automation functionality as an unexpected behavior, which might lead to confusion.

Two dimensions required in Arrangement View

Patterns shall be able to be laid in parallel.

A wrong assumption we made in this prototype is that we think the composer's musical concepts could be segmented into a chain of time blocks. Therefore, the arrangement view can only place patterns one after the other. Many participants pointed out that the patterns might also intersect or lay in parallel to each other. The flexibility of arranging elements needs to be improved.

Challenge - Keep it simple Norman [1988]

In this prototype we have tried to integrate the information of all levels in one screen. The Arrangement View demonstrates the high-level definition, and both the Editor View and the Library View show the detail. The similar functionally of Editor and Library might bring about many potential slips, such as /description error Raskin [2000]. It is not necessary for these four views to be displayed simultaneously all the time. We need to come up with a better strategy to display only the necessary information to users, while it is still easy to shift focus to the other level.

Redesign of the system

From the storyboards we realized that we have made several incorrect generalizations and assumptions of the composers' musical concepts. This results in unexpected behavior for the users and makes the system hard to grasp. We will improve these strategies in the next iteration.

Chapter 7

Paper Prototype

Paper prototype shows the fundamental issues of the design.

Paper prototype is a quick and low cost evaluation technique to discover fundamental issues in the interface design at the very early stages Snyder [2003]. Because Paper prototype only shows the high-level definition of the interface, the users can focus on giving fundamental feedback without worrying about the implementation details of the interface.

The system is split into two layers to reduce the complexity of the interface. One shows the structural view of the song and the other shows the detail.

Based on the findings from the storyboarding 6.2.4—"Findings", we want to divide the information into two layers to reduce the complexity of the interface. We split the system into two windows: the pattern editor window and the arrangement window. The arrangement window shows the structural view of the composition and the pattern editor assists the user in creating the content. Because the intricate interaction between user and system occurs mostly in the idea realization, which will take place in the pattern editor window, we want to first examine the design detail of the editor with this prototype. Furthermore, as we have made several changes to the process of making patterns, we need to ensure the new approach does fit to the composers' workflow.

7.1 Design

7.1.1 Changes

Return to the Basic

Each view with one focus.

As mentioned before, we will divide the system into two windows with each window having only one focus, in order to simplify the information load of each view. In the next software prototype we will demonstrate how this division allows users to easily shift between the two views.

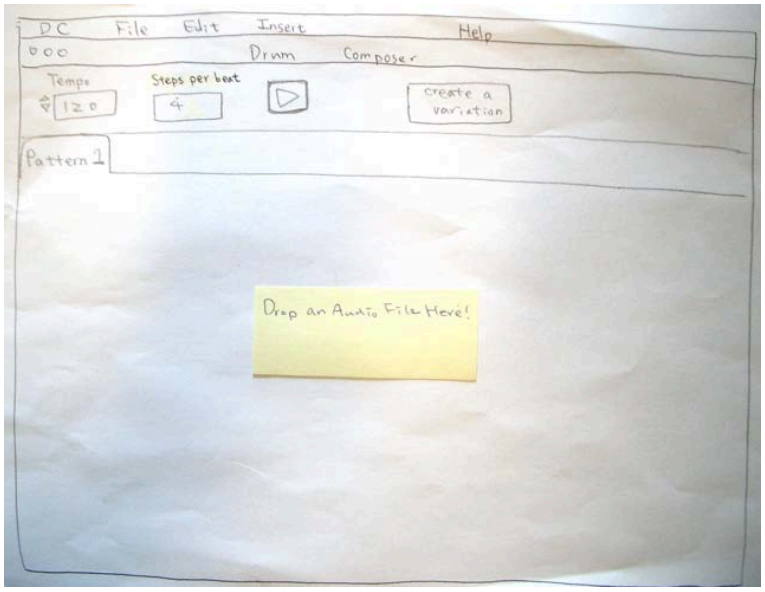

Figure 7.1: A screenshot of the pattern editor

One Single Disk for a Subpattern

A set of disks replaces concentric circles.

Concentric circles appeared to be distractive when growing bigger, and therefore failed to give the users an overview of all subpatterns. Therefore, in this prototype we apply one single disk for each subppatern. Nevertheless, inability to compare two distanced subpatterns would be a potential issue of the new setting. Due to the static nature of paper

prototyping, we will discuss its impact on comparing patterns in the software prototype (see chapter 8—"Software Prototype".

Leave out the Library View

As the concept of template patterns appeared to be unintuitive to the participants in the last prototype, we intend to simplify the process of creating a variation pattern. When the users enter the editor, they encounter a blank tab indicating the data required to proceed to compose music (see 7.1). Once they drop an audio clip into the tab, a disk appears as the editor of a subpattern for the corresponding audio clip (see 7.2). The users can start editing the subpattern and further add some other audio clips to take the role of the other instruments in the composition. Whenever the users subsequently intend to expand the current pattern, they click on "Create a Variation" to make a duplicate of it.

The users work directly on the resulting patterns, instead of template patterns.

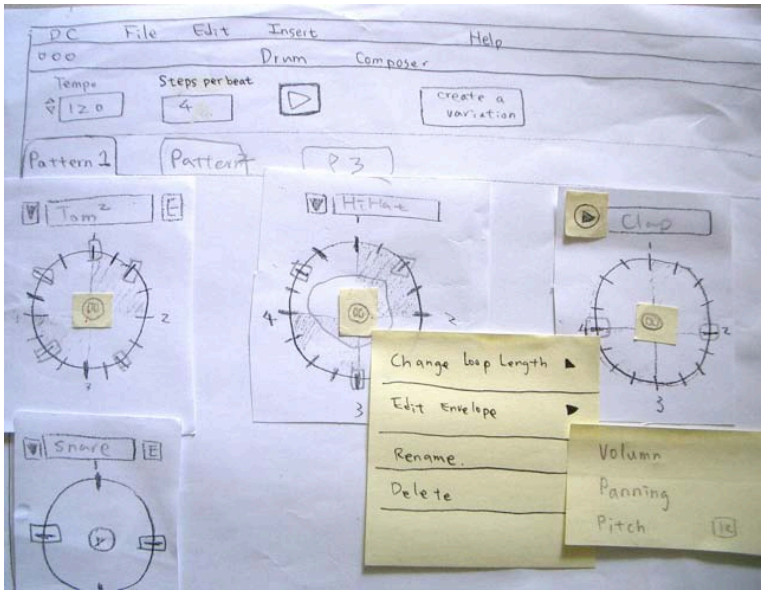

Figure 7.2: Stickers and pieces of paper are used for flying dialogs and status of the buttons

Furthermore, all patterns stemming from the same pattern share the same set of musical material; if one audio clip is added into a pattern, the clip will be attached to all logically related patterns, and vise versa. However, all patterns

All conceptually related patterns share the same set of material.

are independent as the users can manipulate or deactivate any of their own subpatterns without influencing the other variations. As a set of variations are supposed to be a group of similar patterns, they should share many common features. We use such a logical constraint Norman [1988] to avoid the misuse of the variation concept. For example, the user can create an unrelated variation pattern by deleting all its original content, and then add the new material into it. In this way the concept of variation propagation would have no impact to the data organization. If the user wants to create a new pattern but not an extension, they are supposed to create another document. We believe this concept can assist the composers in organizing patterns logically. In addition, in order to give the users more control over specifying their musical concepts in each variation, the layouts of the variations are independent.

7.2 Features

7.2.1 Toolbar

Tempo
The user can set the number of beats per minute.

Steps per beat
The concept is referenced from the conventional drum machine, where the user can set the time span of each step.

Play/ Stop button
The user can play or stop all activated subpatterns.

Create a variation
As long as the current pattern is not empty, the button is activated and allows the user to create a variation pattern based on the current pattern.

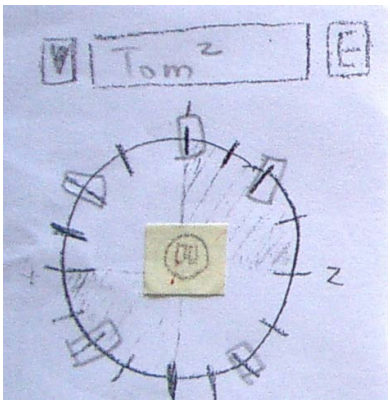

Figure 7.3: The paper prototype of the disk interface

7.2.2 Disk Interface

Steps
The number of steps in a loop determines the loop's time span, meaning that disks can have different repetition periods.

Play Toggle
The toggle allows the user to play a single subpattern. The user can toggle a couple of subpatterns to try out different combinations of the current set of instruments.

Activation / Deactivation Toggle
It allows the user to activate or deactivate the playback of a related instrument. With this functionality, even though all variations share the same set of musical elements, they are able to have different subsets of elements being displayed.

Envelope Menu
An envelope menu is attached to each subpattern, where the user can adjust the basic music features such as panning and volume. We put an envelope control on the disk layout because they tend to be accessed frequently.

Envelope Editing Pane
With an envelope selected, the user can draw the value of the musical feature. This feature will be further elaborated in the software prototype.

The Name of the Instrument
The name is by default the file name of the audio clip. It is free from the user's changes.

7.2.3 Main Menu

To simplify the interface, we leave out all disk-specific functions to the context menu of the disk, such as change the length of a loop, and change envelope. The functions shown in this prototype are as follows:

File
Import, export: The user can import, or export a single pattern in order to use it in another project. They will be discussed further in chaper 10.2.8—"Standard Features".

Create a new pattern
It creates a document for a new pattern.

Insert
New audio file: To add an audio clip into the current pattern

7.2.4 Context Menu

When the user right clicks on a disk, the system displays its associated context menu.

Change loop length
To specify the number of steps used in one ring.

Delete
To delete the selected disk.

7.3 Analysis

7.3.1 Participants

The participants consist of 2 computer scientists without experience in music software but with knowledge of interface design, 3 composers and 2 participants with experience only in traditional instruments. We include computer scientists this time especially to discover technical issues of this prototype. Also, we enquire the novice users' opinion to gather the missing information for the inexperienced users.

The participants comprise the composers as well as the computer scientists.

7.3.2 Task

Wizard of Oz

Before the session began, we informed the participants we were going to play the role of a computer. We informed them that they would be given a cursor to indicate their instructionsand we would display the tool tips and moving dialogs according to their instructions. We asked the participants to explore the interface and try to create some patterns. If the participants have a musical composition background, we asked them to reproduce two of their own patterns on the paper prototype.

We would react to the participants' instructions as a computer during the task.

7.3.3 Results

Each user test took around 40 minutes. With the aid of the simulated cursor the participants were able to quickly immerse themselves in the paper prototype, although they tended to forget the role of the interviewer as a computer.

The composers all appreciated the concept of variation propagation. They thought it follows their workflow and would alleviate the required technical effort to compose rhythms.

The novice participants could already produce regular rhythms without audio feedback.

Because the audio and animation feedbacks are missing on the paper, the participants who have no experience with music software found it difficult to imagine the resulting pattern. They tended to assign some arbitrary beats and put more focus on interacting with the other parts of the interface, such as how to create, duplicate patterns, but less interaction with the disks. Nevertheless, the beats they arranged had the tendency of the multiples of 45 degree, which indicated they were already able to create regular rhythms on the interface without any audio feedback. On the other hand, the participants with knowledge of musical software were able to easily grasp the concept and quickly produced their desired rhythm. Furthermore, they were keen to explore how to manipulate the envelopes in the circular form.

In addition, since they were just holding a cursor made of a piece of paper, they did not know they were able to interact with the paper prototype with the inputs from a mouse.

7.3.4 Findings

Monotonous entrance hard to find

Many participants did not access the context menu associated with a disk.

We intended to keep the access of functionalities in our interface monotonous Raskin [2000] by offering only one single entrance to access the functionalities of a single disk. In other words, the disk-related functionalities cannot be found in the menu bar. However, many participants tended to browse the menu to see the available functionalities the first time they used the system. As they did not find many controls related to the disk either from the menu or the disk layout, they could not proceed. A participant suggested the user should be able to perform all functionalities by using menu bar. This issue will be examined again in the software prototype to see if it would be better off when a mouse is present.

Multiple entrances to playback audio

The users can play audio with different methods, which often resulted in confusion.

On the other hand, the local play button conflicts with the system play button as they both have the same goal of tog-

gling at least one instrument. Therefore, the local play button will be removed to simplify the interface.

Modes introduces conflicts

Two control areas overlaps.

The control area of the envelopes conflicts with the location of the Play button. Although it is easier to trigger when it is in the center, the interaction within a disk would be much more complicated in order to avoid the mode conflicts. We will move this button outside the disk in the next prototype to avoid introducing modes into the system. Similarly, the steps conflicts with the control area of the envelopes because they have an intersection zone, as shown in 7.3. The issue will also be resolved in the software prototype.

7.4 User Suggestions

Consistent arrangement of the variations

The related patterns share the same set of material as well as the same layout of the arrangement of the disks.

In his prototype the user can configure the layout of variation pattern. Some participants expected the synchronous layout among all variations, so that when they switch between different variations they can easily spot the conceptually linked instruments.

Terminology

Several terminologies need to be corrected.

Several terminologies used in the prototype are not precise. For example, as we expect the user to use audio clips as the resulting output, "Drop an audio file here" conveys broader meaning. We will instead use "drop an audio clip here" in the next prototype. Also, only the experienced participants understood the term: "Envelopes".

The term should be more straightforward.

"Change the loop length": Many users pointed out that the labeling is confusing. As the disks appear to be of the same size, they do not see it as will be changed to set number of steps. We will change it to *"change the number of steps"*.

The functionality was too advanced to most participants.

"Steps per beat" is the resolution of a step. The higher the number the shorter time span a step takes. However, it

tends to be confusing when considered altogether with the idea of the number of steps per loop. Many participants needed further explanation to grasp its meaning. We believe by using a more straightforward terminology such as *"resolution of a step"*, accompanied with visualizing their resolution by the size of a step, the problem could be resolved. It will be discussed in the future work.

Folding the deactivated subpatterns

A participant suggested to fold the unused disks to simplify the interface.

Activated and deactivated subptterns do not have apparent visual differences from the disk layout. A participant suggested that the system folds the deactivated disks to hide the unused data. However, folding will further introduce a mode that reduces the visibility of the subpatterns and would not save any space in our design. Therefore, we will instead gray out the deactivated disks to give a stronger visual cue of the status of the disks while still keeping the consistent overview of all disks.

Play/stop/pause buttons

Pausing the audio engine is the standard functionality of the conventional software tools. We will include it in the next prototype.

Fundamental envelops

A fundamental envelope for editing sound features.

Many composers needed the standard effects ADSR[1], which can facilitate them editing the audio source on the fly, instead of using external audio editing tool. It will be considered in chapter 10.2.8—"Standard Features".

7.5 Conclusion

We gained divers opinions from two different perspectives.

It is a correct decision to include participants from different areas. We have gained very diverse opinions to the prototype, while many of the opinions have an influence to our system design. For example, the computer scientists spot

[1] An ADSR envelope is a function for modulating some aspect of the instrument's sound, such as loudness, over time.

the issue of modes we introduced while the composers suggested the essential features for rhythmic composition.

The evaluation shows our design is beneficial to the composers.

From the composers' feedbacks we are confident that the concept of variation propagation will be useful for their creative activity. Nevertheless, in terms of the interaction among disks, since the audio feedback and animation are inherently missing on the paper prototype, the applicability of the layout could not be examined. It will be further explored in the software prototype.

Chapter 8

Software Prototype

After examining the detailed interaction in the Pattern Editor, many ambiguous design issues have been adjusted. Now, we develop the software prototype to concretely demonstrate the proposed concept.

8.1 Design

DISCO is comprised of two views. The Semantic composer represents the overview of the musical structure to the users, and the Pattern Editor allows the users to access the low-level musical elements in a logically organized order. Figure 8.1 shows the conceptual architecture of DISCO. Both windows are divided into three parts: a customizable tool bar, a side bar, and a main editor. Their design and changes are elaborated separately as follows:

8.1.1 Semantic Composer

Tracks are neutral for arrangement.

In this prototype we introduce a multiple tracks interface to the semantic composer, so that the users can arrange their musical material in parallel. The tracks have no imposed definition; they are neutral to be linked to all musical elements. However, through the later user study we

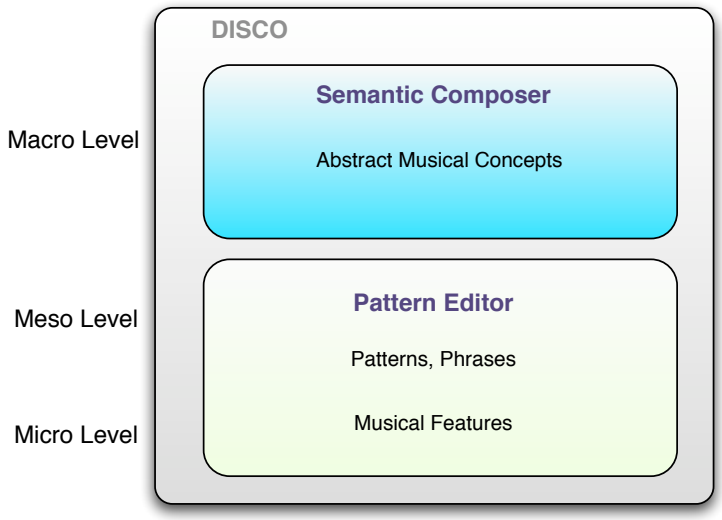

Figure 8.1: The conceptual architecture of DISCO

found this design decision conveyed a false constraint to the users, which will be discussed in the results section 8.4.2—"Findings". Furthermore, the system will not automatically append the newly created material to the arrangement view to avoid unexpected behaviors to the user.

Musical elements are grouped in chunks. Chunks are interchangeable.

Apart from the changes, we also add several grouping functionalities to the arrangement view. The users were able not only to arrange elements one after the other as in the conventional track-based sequencer, but also to group the musical elements according to their interrelationships. Variations are interchangeable, which will facilitate the phase of evaluation of end products a great deal. For example, in the typical setting, users need to directly access the low-level musical elements. The pending ideas will be lost upon any further user changes. In our system users can easily try out different combinations of ideas since the system supports them in keeping the tentative ideas in a logical order.

8.1.2 Pattern Editor

Several modifications will be made in the Pattern Editor to reflect the analysis of the last prototype. However, its layout will mostly stay the same. Figure 8.2 shows a screen shot of the main screen.

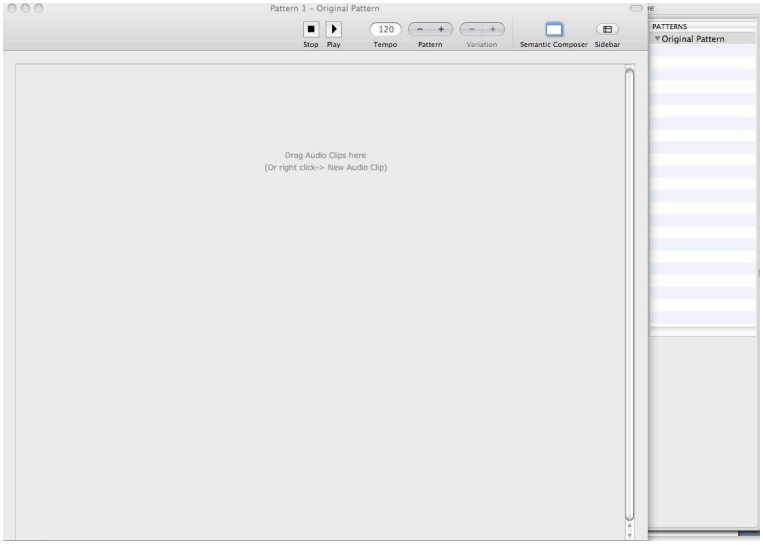

Figure 8.2: The Pattern Editor

8.2 Features

8.2.1 Semantic Composer

Create Items

Arbitrary number of tracks could be employed to arrange the material 8.3. As our prototype is aiming at pattern-based composition, the tracks are calibrated based on measures. The numbering of measures is displyed under the tool bar.

Arbitrary number of tracks for arrangement

The attached side bar shows the current list of patterns, which is synchronized with the content of Pattern Editor.

Advanced drag and drop helps users group their musical concepts.

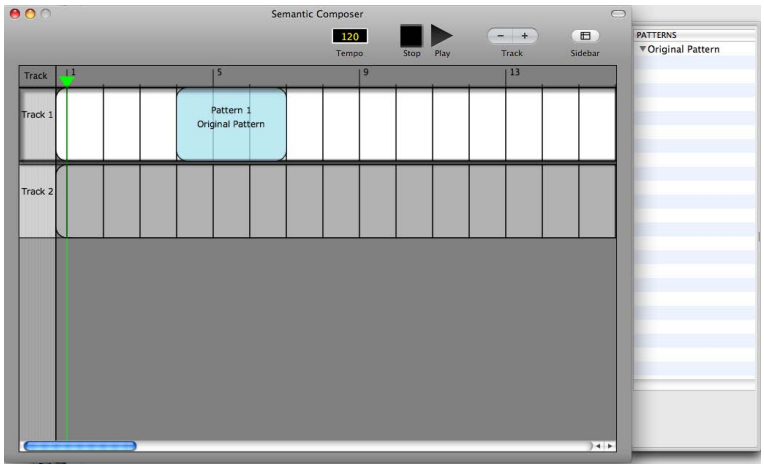

Figure 8.3: The Semantic Composer: arbitrary number of tracks could be employed.

To add a new pattern, the user can drag a pattern from the side bar to the desired destination, or double click on an item on the side bar. While dragging, a transparent pattern image indicates the pending destination of the dragged item, as shown in figure 8.4. On the other hand, deleting a pattern could be performed by accessing its context menu as well as by using the delete key from the keyboard.

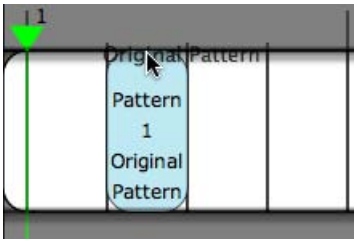

Figure 8.4: The image shows the destination the drag.

Furthermore, a pattern could be dropped onto its variation pattern, to partially replace the base pattern without messing up its original content as well as further group up the user's musical concepts 8.5.

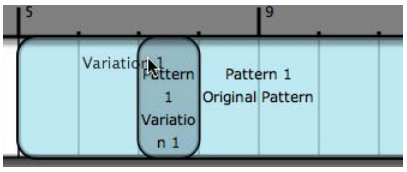

Figure 8.5: Variation 1 is dropped onto the Original Pattern.

Edit Items

The system highlights the selected item to increase the visibility of the current operation. By double clicking on a specific item, the user can directly zoom-in to view the item's details.

The focus could be easily shifted to different levels.

In the evaluation phase, if the user comes up with an extended idea of a pattern, he could make the request by accessing the pattern's context menu (see figure 8.8). His focus will then shift to the Pattern Editor to realize the extended idea by modifying a duplicate of the original. As a result, the user could easily reuse the musical elements based on their correlations. On the other contrary, in the typical setting the user has to interpret each low-level element and retrieve their interrelationships manually.

Evaluate the Content

As we discovered through the field interviews that many composers tended to rely on color to differentiate the musical elements, we assign similar colors to related musical elements (see figure 8.6).

While the measure ruler informs the user the measure of the related musical element, the time indicator represents the actual time of the currently processed elements. It shifts back to the start position as long as the audio engine is terminated or the stop button is pressed. The tool tip is shown when the cursor hovers over the time indicator or when the time indicator is dragged (see figure 8.7) Also, the user can drag the time indicator to a specific position to start playing

Time indicator displays the actual time information.

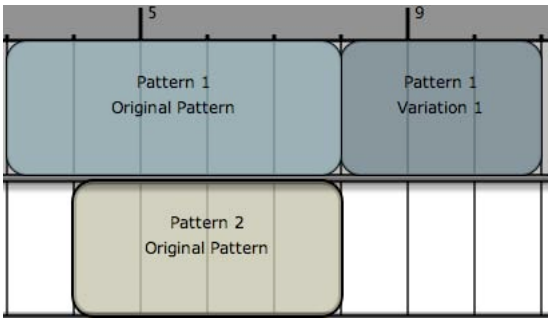

Figure 8.6: The colors indicate their interrelationships.

the musical material from an arbitrary measure.

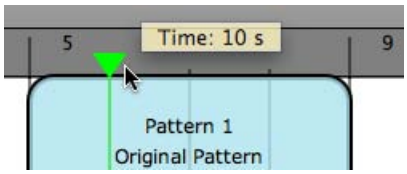

Figure 8.7: The time indicator shows the actual time.

Interchangeability of the variations enhance the process of evaluation

In our field interviews, we noticed that the composers tended to replace a musical element with similar versions to find out the best fitting version for the whole work. Therefore, to make the process of evaluation even more effective, the system allows the user to switch the content of a pattern between its variations by accessing the pattern's context menu (see figure 8.8).

Figure 8.8: The context menu of a specific time event; the content is interchangeable.

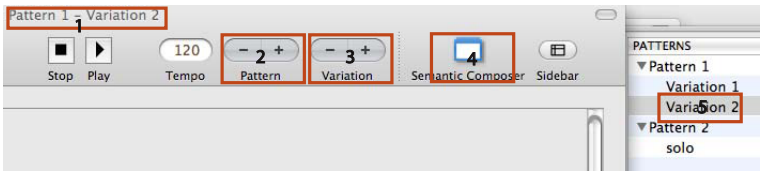

Figure 8.9: The window name indicates the selected pattern (part 1).

8.2.2 Pattern Editor

Create Patterns

Create initial or extended patterns
The user can create a blank pattern by using the "Pattern" button (see figure 8.9 part 2). A pattern in the top level will be created accordingly. To create an extended pattern, the user first selects the pattern he wants to extend, and clicks on the "Variation" button (see figure 8.9 part 3); a duplicate pattern will be generated under the pattern it extends. The system automatically switches the content view to the newly generated pattern with the name specified on the window (see figure 8.9 part 5).

In addition, to add an audio clip into the pattern, the user can access the context menu (see 8.2.2—"Edit Patterns"), access the menu bar or directly drop audio files into the edit window.

The first time the user creates a pattern, the side bar automatically appears to inform the user where to access the stored patterns.

The table inside the sidebar represents a shallow tree structure of the stored pattern (see figure 8.10). The first level is the initial pattern to which all the descendents refer.

As we noticed in the field interviews, while the composers are editing variations, they switch back to the initial pattern frequently to compare the differences between the variations. In order to improve the effectiveness of the composer's creative process, we use color as a visual cue to al-

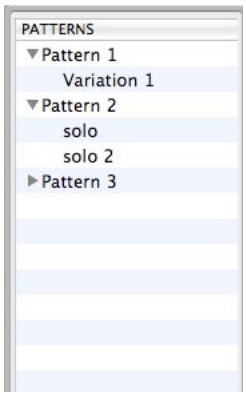

Figure 8.10: The sidebar represents a shallow hierarchy of the patterns.

low users to compare different versions without switching between them.

For example, figure 8.11 is an initial pattern and figure 8.13, figure 8.12is its descendent. The red racket indicates the "missing" step from the original, while the yellow step indicates the add-on to the original pattern.

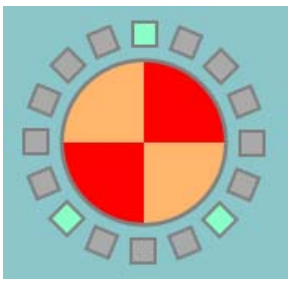

Figure 8.11: An initial pattern.

Edit Patterns

Pan uses spatial mapping while Volume use circular metaphor.

Envelopes When the user clicks on a disk panel, it is highlighted. To manipulate the basic sound feature of an instrument, the user can access the popup menu on the disk panel 8.14. As shown in figure 8.12, the control of Pan employs the spatial analogy; when the user drags the control to the

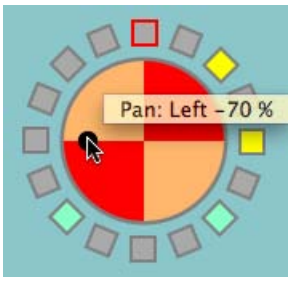

Figure 8.12: A variation pattern with the pan envelope.

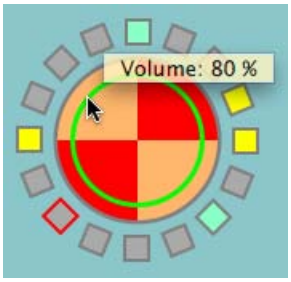

Figure 8.13: A variation pattern with the volume envelope.

left, the sound of the related instrument will come from the left side of the speaker, and vice versa. In addition, figure 8.13 shows a glowing ring for controlling the volume. The radius of the ring is used as a metaphor for the value of the volume. The ring grows when the user drags it outward from the center.

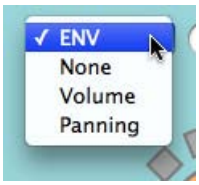

Figure 8.14: A popup menu for editing sound envelopes.

Context Menu Depending on the location where the user clicks, a context menu pops with distinct content 8.15. The selections are lists as follows:

1. New Audio Clip

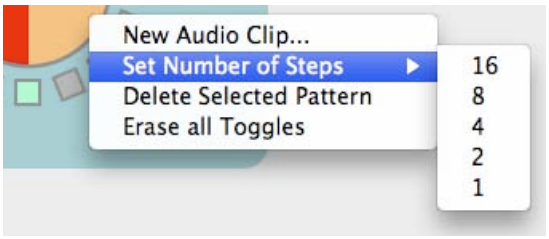

Figure 8.15: The context menu of a subpattern.

The user can add a new clip as the sound output of an instrument by accessing the context menu in the main editor of the Pattern Editor.

2. Set Number of Steps

This feature allows the user to change the time span of a loop. The size of the disk also reflects its length; the steps are limited to multiples of two, to keep all loops in sync.

3. Delete Selected Pattern

The selected subpattern can be deleted. As explained in the paper prototype, all associated subpatterns in the other variations will also be deleted.

4. Erase all Toggles

Clicking on the item erases the current creation when the user wants to start from scratch with the same instrument.

Switch View

When the user is done with creating patterns, he may switch to the Semantic Composer window to arrange patterns. He can click on the "Semantic Composer" button on the toolbar to bring the window to focus (see figure 8.9 part 4).

8.3 Implementation

This software prototype is implemented as a Cocoa application. Cocoa is an object-oriented application environment designed for developing Mac OS X native applications. Cocoa is both in Java and Objective-C (++); however, because the Java interface is no longer maintained in parallel with the Objective-C interface, we develop the prototype in Objective-C to obtain more up-to-date support. To gain the most out of the operating system, we employ the audio processing services from Core Audio. It is tightly bound to the Mac OS and offers solutions to all audio needs. As Core Audio is written in C language, our classes are a mixture of Objective-C and C language.

8.3.1 Model-View-Controller Paradigm

We adopt the MVC paradigm in our system design 8.16 It is a recommended design pattern for Cocoa applications. The model offers a data source that is isolated from the visual and logic layers. The controller acts as the intermediary between the view and the model. It could offer control logic for the views as well as perform coordinating tasks for an application. The view is a visualization of the model to the user based on the attached control logic. In the following section we will illustrate how we designed the system based on this paradigm.

Document-Based Architecture

MyDocument 8.17 contains the data shared in the system. As the application is instantiated, it creates two extra window controllers. The EditorWindowController creates the PatternEditor window to mediate between the source data and the user's input into the view. The BrowserWindowController creates the SemanticComposer window and a ViewController to update the view against the source data manipulated in the PatternEditor window.

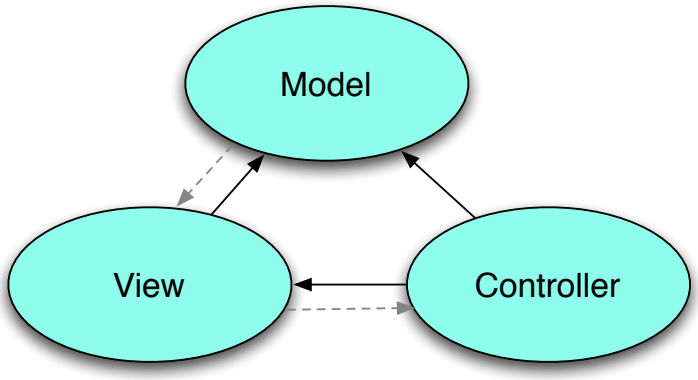

Figure 8.16: The Model-View-Controller paradigm

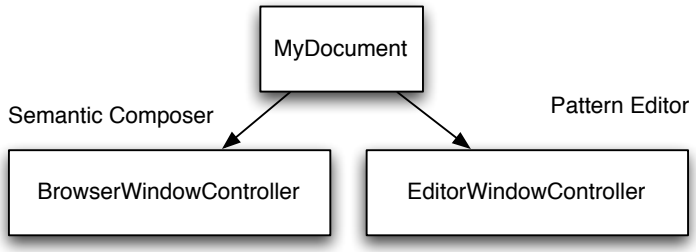

Figure 8.17: Document-Based Architecture

8.3.2 Pattern Editor Classes Structure

Model Classes

Ring

This class represents all information related to a single sub-pattern. It contains instance variables for audio file information, musical features, and visualization information for the circular interface.

RingCollection

A RingCollection represents a set of conceptually related

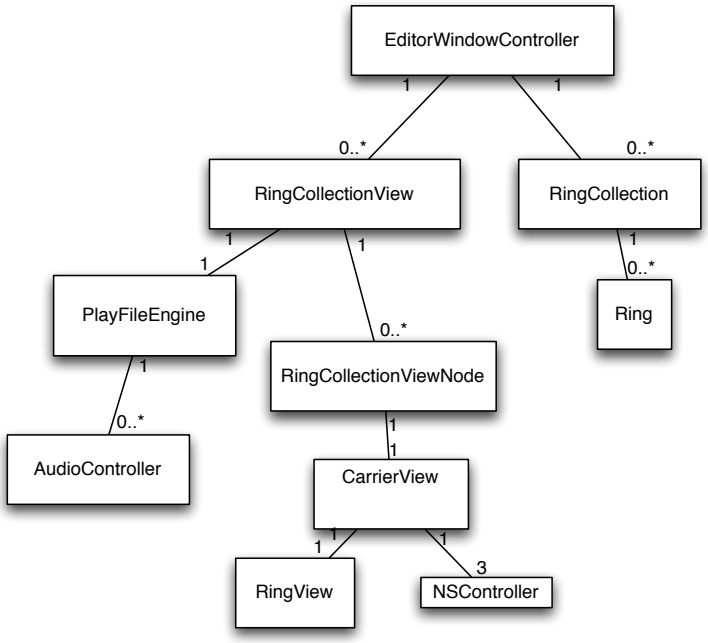

Figure 8.18: Pattern Editor Classes Structure

patterns. It contains an array storing all variation patterns. An instance variable rings specifies the currently selected set of subpatterns.

View Classes

RingView

All interactions with the circular step sequencer occur in this view. As shown in figure 8.18, it observes the Ring model to keep track of all changes from the EditoWindow-Controller to the Ring model, and visualizes it as a circular step sequencer.

CarrierView

It archives the controllers and views for presenting an audio file as an instrument, which comprises a RingView and

3 controllers for specifying envelopes, toggling the state of a subpattern and displaying the name of the attached audio file (see figure8.19).

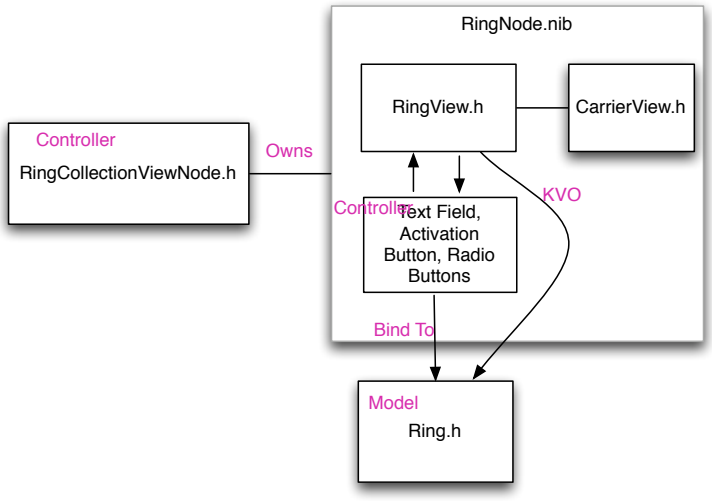

Figure 8.19: A ring node archives several controllers and two views

RingCollectionView

This view keeps the presented information in sync with the model's content. As shown in figure8.20 the RingCollectionView observes the instance variable rings in RingCollection. It generates a RingCollectionViewNode for each object in rings. When the user adds or deletes a subpattern, the RingCollectionView reflects this change by reloading its array of RingCollectionViewNode.

Controller Class

RingCollectionViewNode

It is a coordinating controller. It owns the views and controls for the GUI of one subpattern. RingCollectionView can easily update its subviews by accessing RingCollectionViewNode.

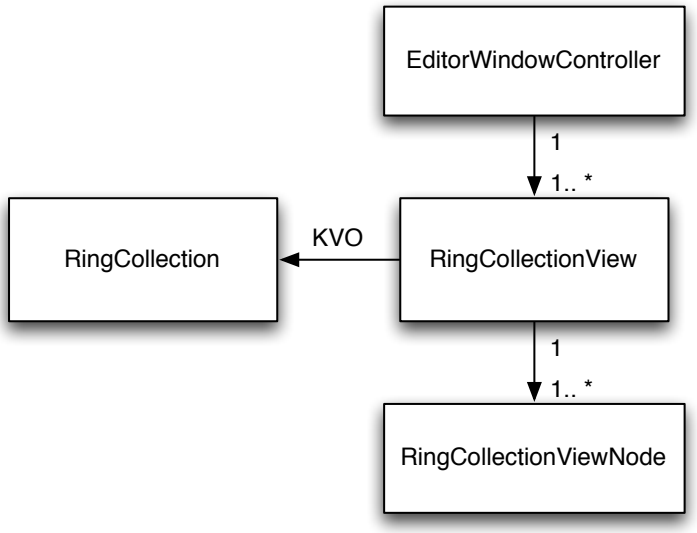

Figure 8.20: The view observes any changes to the model

Audio Engine

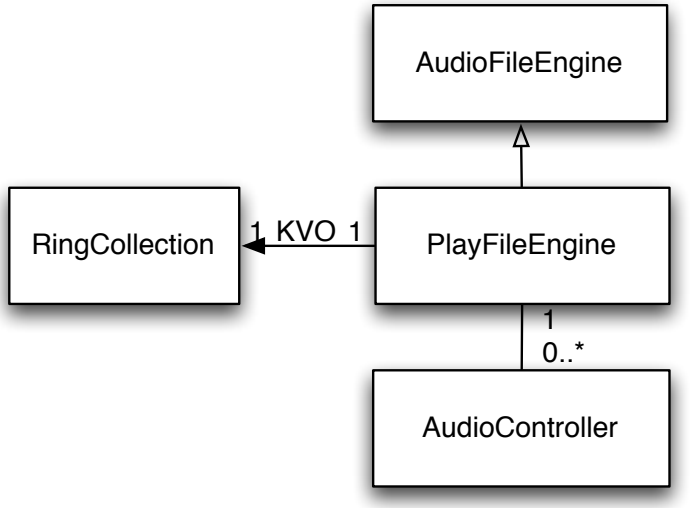

Figure 8.21: Play File Engine

AudioFileEngine 8.21 can be used to display audio stream.

PlayFileEngine is a specialized audio file player for the step sequencer. PlayFileEngine generates an AudioController for each audio file, which will be stored in an audio queue. When the user presses the play button, the central clock of PlayFileEngine ticks the audioControllers in the audio queue to display the audio.

PlayFileEngine contains an output unit for audio playback, which is bound to a mixer unit for multiple audio inputs. AudioController routs the audio input to its effect units to modify the volume and pan, and then sends the result to the mixer unit in PlayFileEngine to display the audio. Figure myImgRefaudioengine illustrates the message flow.

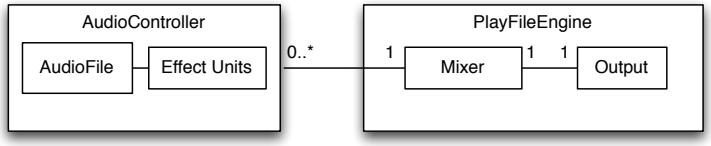

Figure 8.22: Audio controller

8.3.3 Semantic Composer Classes Structure

The architecture of the Semantic Composer is modified from Jonas' Tactile Editor. The original design and classes description can be found Jonas [2008]. Many modifications are made in the Semantic Composer to synchronize with the Pattern Editor as well as render audio events. Figure 8.23 illustrates the class architecture of the Semantic Composer.

Most interactions in the Semantic Composer occur in CenterstageView. It accepts users' mouse or keyboard events, which are then processed by ViewController. The class Clip represents the time event to trigger a pattern specified in the Semantic Composer window. It contains the instance variables for the start time and the end time of an audio event. An array of tracks is used to store the time events based on their location of the vertical axis of the screen. For each time event an AudioController is generated in

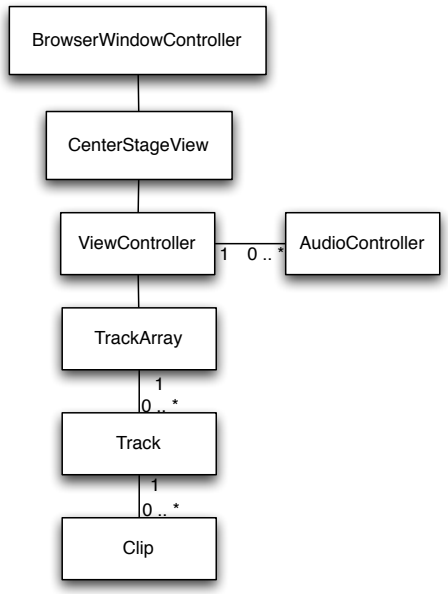

Figure 8.23: Semantic Composer Classes

the ViewController to render the target audio stream to the PlayFileEngine.

8.4 Analysis

The prototype was evaluated with 4 composers and 3 computer scientists. Similarly we wanted to acquire feedback from the different perspectives of these two groups. They were encouraged to produce a piece of rhythmic music containing several different patterns.

8.4.1 Results

All participants understood the concept of variation propagation and regarded it useful for musical composition. The consistent layout among variations reduced the required effort for the composers to orient themselves with the con-

All participants regarded the concept beneficial.

tent.

Nevertheless, many deficiencies were found when observing them interacting with the system. These findings and their impacts to the system design are summarized in the next section.

8.4.2 Findings

Gestalt: Tracks limit the potential behaviors

The interface suggested the wrong behavior.

As most of the participants are already very familiar with the conventional tracks metaphor, they still relied on the tracks to group their material. Many of them still positioned the material physically. To make the visualization of material centered on the musical semantics, we will leave out the tracks to avoid distraction.

Preconceived behavior of the system.

An alternative metaphor to introduce the new concepts.

Similar to the last finding, most participants were already used to the typical composition software, where the base element will be erased when another item is dragged on top of it. The participants instinctively arranged the materials one after the other. To solve this problem, we consider the example from 3.1.3—"CyberBand", Abrams et al. [1999], where they offer various kinds of "mask" to allow the users to modify the original content. Similarly, we can introduce a "mask" metaphor to make the users feel as if they are actually applying an external effect over the base element, which will replace the content of a pattern with another variation.

Nevertheless, after explanation, many participants appreciated such a concept, because they can evaluate tentative ideas without messing up the initial content and group them according to their interrelationship. They agreed this feature would leave out many chores they used to perform.

Reaction of dragging an item

Many participants expected the items to react upon their

dragging, while they actually need to drag the item over half of the neighboring block before the dragged image appears. We will reduce the threshold of motion to resole this issue.

Adjustable hierarchy

The user can freely configure the structure.

Many participants employed the variation functionality to perform ideation, meaning that they begun with a raw idea, experimenting with the idea in different variations. Finally they might feel satisfied with one of the extended ideas, intending to replace the original with it. Therefore, we will enable the users to freely configure the structure of the pattern table in the side bar in order to make the data organization closer to their musical concepts, as shown in figure 8.24.

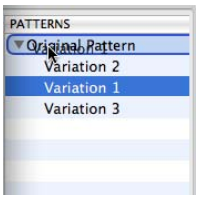

Figure 8.24: The user could drag items to change the hierarchy of the patterns.

Confusion of the steps

The size of the the disk did not have an impact.

Changing the number of steps of a disk could change its time span. To avoid confusion, we intended to use the size of the disk as a metaphor to show the differences of the time span of the disks. Unfortunately, this did not have a strong visual impact as the disks are surrounded by different number of steps. Many participants did not consider this concept intuitive but it can be easily grasped once they gain some experience with the system. However, most of them suggested keeping this feature, as the flexibility it could bring about is far more valuable than the confusion it reveals at first sight. We shall consider adding more visual cues such as changing the saturation of the overall color of the disk in the future.

The activation button conveys false message

The deactivated disk will be darkened.

Many participants considered the button as a play toggle, as there is a speaker icon attached to the button. To make the functionality easier to understand we will grey out the disk interface when the user switches it to the deactivated mode (see figure 8.25.)

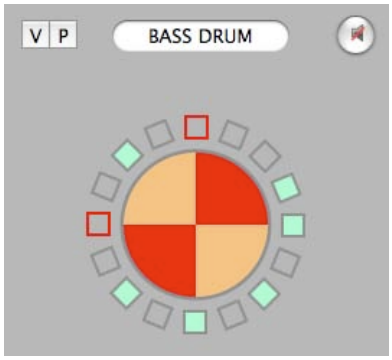

Figure 8.25: The panel is darkened when deactivated.

Distraction

It is problematic to identify one instrument when many others are present

As the set of disks grows, it gets harder to find a specific subpattern, because the users do not really access the disks with the file names. The standard functionalities such as mute, solo, and preview an audio file could assist the users orient themselves 10.2.8—"Standard Features".

Multiple modals alleviate the potential issue

Instrumentness and affordances

As we considered in 7.3.3—"Results", the users might need a way to visually compare the disks. In fact, they usually compare the content of the subpatterns by listening to the audio output instead of comparing them visually. While listing to the audio output, the user's visual focus is on the disk he intends to edit. In other words, the user regards the audio feedback as the static reference, while the visual representation of a disk is the variable he is currently manipulating.

However, it could be that the users just adapt to what the interfaces afford themBertelsen et al. [2007]. For example, if the system allows them to merge two disks, they could then compose patterns using a visual approach, such as making

a counterpoint or a symmetric pattern against another pattern.

8.4.3 User Suggestions

use DISCO in live sessions

We received numerous suggestions about the usage of variation propagation. Firstly, since extended ideas could be quickly constructed, many composers would like to use it in a live session. Secondly, the clear structural view could be used as a shared musical material for live collaboration.

Experimental features are often requested.

In terms of the circular interface, some participants also suggested to allow more freedom of changing the time span of a loop. For example, apart from the regular multiple of 2, the system can allow irregular time span of a loop to generate contingent ideas. Furthermore, two pointers of start and end points of a loop could be added into the disk interface, so that the composer can generate new musical material simply with one single loop. However, while these ideas demonstrate the benefits of using circular interface, they are beyond the scope of our system. The adopted suggestions to our system are summarized as follows:

Creating the skeleton on the High-Level View

To offer a new way to construct the song

While structuring the song on Semantic Composer, the user might also capture a new idea, intending to realize it in a specific location. It would be convenient if the system allows him to specify the location first and then realize the content later, rather than realize the idea first and then switch back to the overview to position the idea. The latter takes higher cost of idea input. Therefore we shall include this idea into the next version, as shown in figure8.26.

Radio menu for simplicity

simplify the interaction

In this software prototype we employ a pop up menu for accessing envelopes. However, it will simplify the process of reaching an item when the items are presented in a radio menu (see figure 8.27). All selections are visible and easy to reach if there are only a few options.

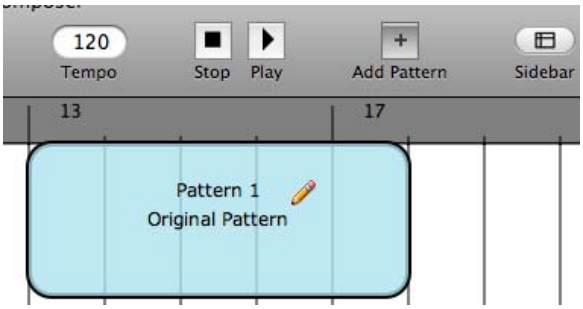

Figure 8.26: New patterns of variations could be created from Semantic Composer.

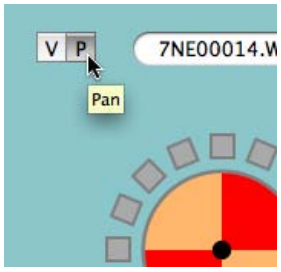

Figure 8.27: Items are visible and easy to select with radio menu

Simplifying the options

leave out redundant options

We noticed that the participants never use the setting of 1 and 2 steps while they prefer to stay with the setting of 16 steps gives the user most flexibility. Therefore, we will leave out the redundant settings, as shown in figure 8.28.

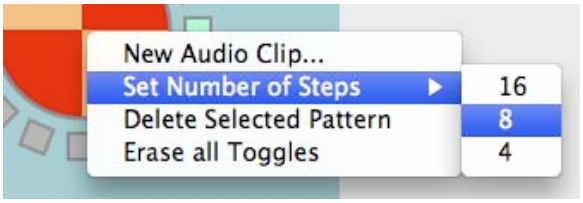

Figure 8.28: Unused options are left out.

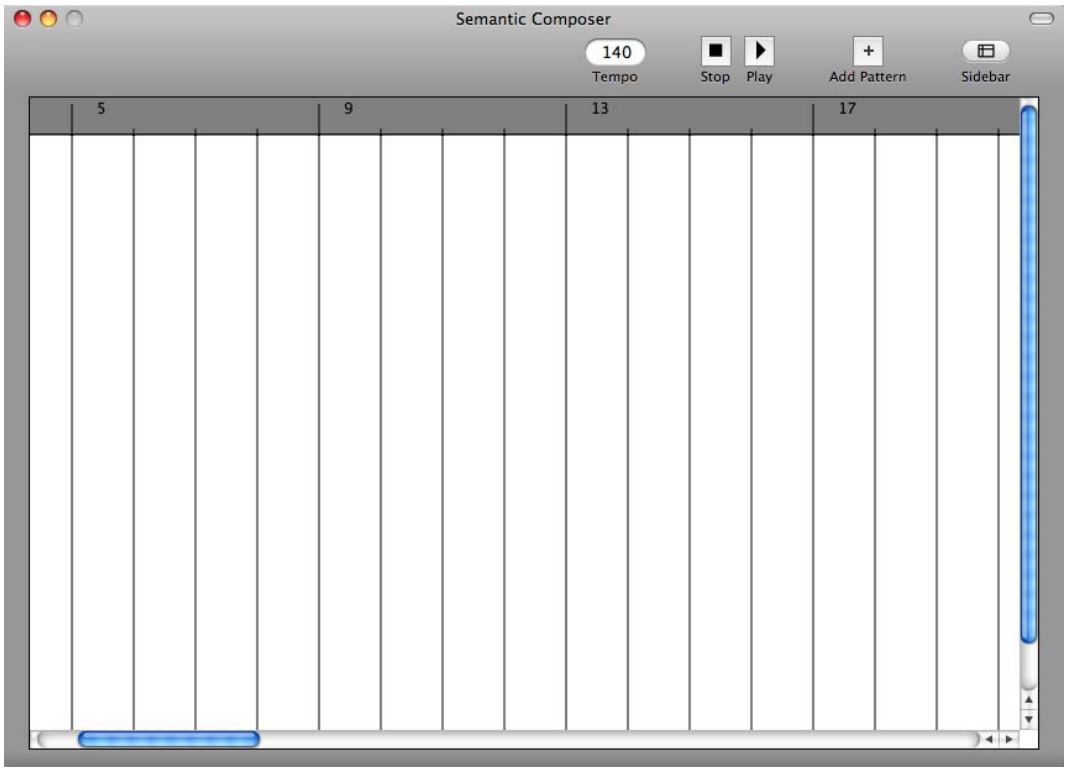

Figure 8.29: The layout of the Semantic Composer after the evaluation phase.

8.5 Conclusion

Many changes have been made since the evaluation. Figure 8.29 shows the appearance of Semantic Composer after removing the horizontal tracks. The redundant procedure of configuring track numbers for increasing horizontal space is also reduced.

In addition, each participant tended to process audio and visual feedback simultaneously but differently. We shall always take the differences in how the brain processes the information from its sensory organs into consideration when adding a new feature, to make the interaction between the users and the system effectual.

Chapter 9

Final Evaluation

Three objectives were set for this evaluation.

Through this evaluation, we would like to see if the proposed workflow is beneficial and favorable to composers for performing their creative activity. Several objectives are set for this evaluation. First, we observe if the participants can quickly grasp the concept of variation propagation and employ it to create the musical content. Second, we see how they employ the grouping feature of Semantic Composer. As a presumption they should be able to quickly shift focus between the overview and the detail to effectively evaluate the outcome as well as refine the musical material. Finally, we would like to find out if the interface also encourages them to perform creative activity.

9.1 Set-Up

Our system and Reason Redrum were used in the evaluation.

The evaluation was held either in a controlled lab environment or in the composer's workspace. A MacBook Pro running Mac OS X was used for this evaluation. Our software prototype and a typical software drum machine Reason Redrum (see chapter 3.1.4—"Propellerhead Reason ReDrum".) were employed during the evaluation session.

Before the session started we informed the participants about the functionality of the two composition tools and the task they were going to perform. Furthermore, we pointed

out that they were not evaluated in terms of the product they produce with the two systems.

9.2 Participants

The composers with various experience were chosen for the evaluation.

As the system is designed for people who are interested in creating music, the participants are composed of people with distinct musical background. All participants have musical composition experience from 2 to 20 years. Ages range from 23 to 40, and 3 out of the 13 participants are female. 6 participants are professional composers, while the others are of various kinds of occupation, mainly students.

9.3 Tasks

The participants were requested to compose several patterns and arrange them in Semantic Composer, and then repeat the same task with Reason Redrum.

We asked the participants to produce at least four patterns with Pattern Editor and structure the content in Semantic Composer, and the same with Reason Redrum. We had prepared four reference patterns; however, they were not obliged to reproduce them. No time limit was imposed to complete the tasks. The first task aimed at encouraging them to try the main features we proposed, rather than examining the composers' "performance" with the system. For example, without a specific task, the participants might just toy around the circular interface, rather than try to produce concrete musical content with Semantic Composer. The second task of working in another system was aiming at showing a common tool of composing rhythms to the participants. In a later stage we will enquire their preference out of these two different composition approaches.

We learn about creativity through the process of composition, rather than the result.

Moreover, as mentioned in Coughlan and Johnson [2006] Sawyer and K. [2003], quantitative measurement of end products for such a domain is not useful, because creative output in this domain is measured in terms of value to individuals. It is the process of composition where we learn about creativity. Therefore, to gain more understanding of the usefulness of our system, qualitative feedback from the

participants is more valuable. After testing these two systems, a qualitative enquiry through a questionnaire will be conducted.

9.4 Results

Each evaluation took around 70 minutes to 2 hours. The session held in the composer's workplace took in general longer and the composition was more fruitful. Without explanation the participants were able to immediately create rhythmic patterns. Their approach to composition could be roughly generalized into two types: some prefer to start with making a skeleton, meaning they produce the basic rhythms and arrange them on Semantic Composer, and then switch back to the Pattern Editor to make further refinements. On the other hand, some prefer to create many patterns at once and then switch to the Semantic Composer to reassemble these pieces.

Their creative approach could be generalized into two types.

All participants enjoyed making rhythms in a circular form. They found it very easy to divide beats in a circle. We also noticed that some participants had already built up some interesting visual rules to make subpatterns. For example, one participant used to make supplementary or symmetrical subpatterns against the others. Some also pointed out that they just learnt the geometrical representation of their favorite rhythms. Almost all participants requested for the feature to rotate the disk.

Visual approach to composition was found in the evaluation.

The most favorable part of the system is the concept of variation propagation. Many participants commented that it demonstrates exactly how they work out ideas and would like to organize musical material. Moreover, although it took a while for many participants to figure out the meanings of the different colors shown on the disk, once they grasped this concept, they appreciated the visual cues to the original version. Some considered the cues were inspiring for extended ideas while others liked it because they reduced the effort required to make comparison between different versions.

The propagation of variation was most favored by the participants. They also appreciated the visual cues on the disks.

9.5 Analysis

Right after the composition session, we gave each participant a questionnaire. The questionnaire sheet can be found in Appendix B—"Questionnaire". It is divided into four parts:

1. Background: In this part we enquire about the composers' composition style, and their experience with traditional and desktop-based musical instruments. We would like to know if the system has different impacts on users of different backgrounds.

2. Understanding of Pattern Editor: Questionnaires are used to further investigate how the participants have understood the special features the circular interface offers, and how they benefit from the concept of variation propagation and material organization when performing creative activities.

3. Understanding of Semantic Composer: To examine whether the interface indeed reflects their musical concepts, and whether the grouping feature assists them refining the content more effectively.

4. Encouraging Creativity: The participants are asked here do they feel encouraged or discouraged for creating musical material with our system. **Summary of the background information:**
9 participants have frequent access to composition software, such as Ableton Live, Cubase, Fruity Loops. They are mostly unsatisfied with the complexity of the interface or the bad organization of the data of their tools.

Moreover, many participants consider their composition process as initially collecting many idea pieces, which may come from existing sound samples, field recordings, or audio clips of their improvisation recording. Then, the song structure gradually evolves as they mix the musical material altogether.

Three participants who have long worked with track-based interface tended to be less in favor of the circular represen-

tation, and asserted they did not specifically find it troubling to work on details over tracks. Therefore, they did not feel the need to switch to the other system.

The statistical analysis of the questionnaire is summarized as follows:

The grouping functionality is favorable, but needs a fitting visual metaphor to convey its concept.

Figure 9.1 demonstrates if the participants consider the grouping feature of Semantic Composer helpful. They agreed that it demonstrated a better overview of their song structure, but also the ability to specify the song structure before realizing the content as well as the close connection to the low level data make it easy to refine the detail. However, many participants found it not intuitive to drop a pattern into another one. They suggested employing a different metaphor, such as folders, to make the concept more understandable.

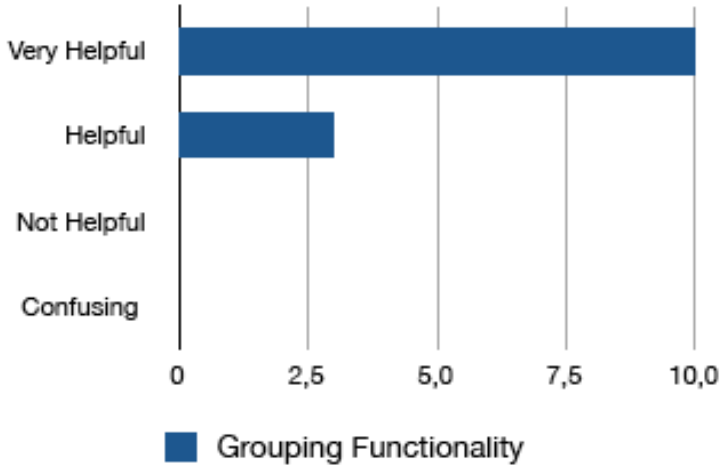

Figure 9.1: The figure demonstrates if the participants found the functionality useful.

The concept of variation propagation is in tune with the composers' creative workflow.

Figure 9.2 demonstrates their preference in creating musical material in terms of the proposed variation propagation and the typical approach presented in Reason Redrum. It demonstrates a clear inclination to the former as most participants considered it to be in tune with how they progress with the musical elements.

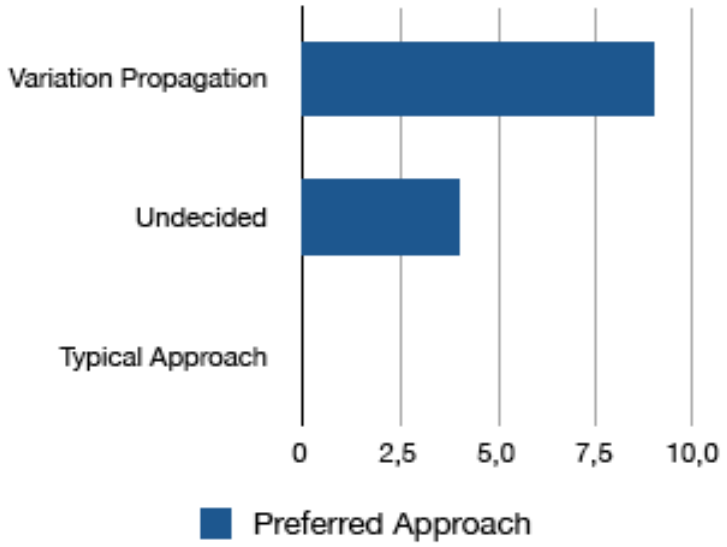

Figure 9.2: The preferred approach of composing rhythms.

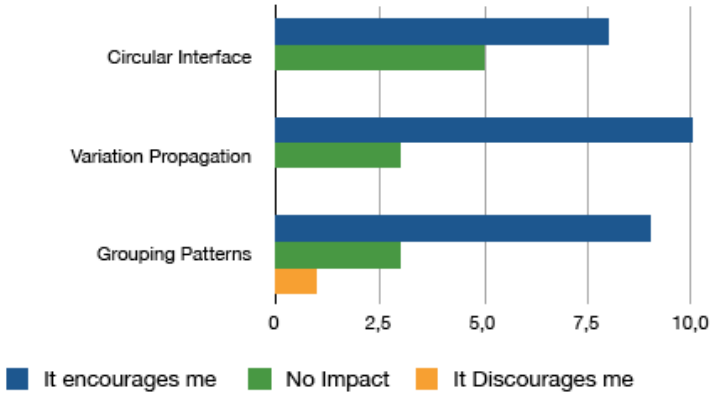

Figure 9.3: The graph shows if the participants found the system inspiring in different perspectives.

The logical organization of the material encourages the users to expand their ideas.

Figure 9.3 demonstrates if the participants consider the interface encourages them experimenting new ideas. The main reason of approval was that the system organized the material clearly and allowed them to expand ideas without losing the original work. In addition, the circular interface

offers them a visual approach to making music.

9.5.1 User comments

Unfamiliar with the new concept

Some participants were adapted to the typical tools.

As mentioned before, some participants pointed out that it might take a while for them to adapt to our system. For example, 5 participants commented they thought about patterns just like the typical tracks. They considered it would be more efficient with their developed habits when composing music.

Representation of the loop length

The concept of the loop length could be quickly grasped, though it is not intuitive.

As shown in previous evaluations, the feature of changing loop length is not easy to understand. 7 participants found it confusing the first time they changed the number of steps of a loop; however, 6 of them commented the concept could be grasped once being taught.

Limitation of the resolution of the beats

The users need finer resolution of the beats.

The current implementation of the maximum number of steps is 16, as it is easy to click on the screen. Some participants pointed 32 steps per loop would be the satisfactory resolution. Furthermore, the resulting rhythms sound very mechanical as the steps are statically divided into 16 parts. We will present several solutions to these two issues in the chapter 10.2.7—"Advanced Features for the Disk Interface".

Simplicity first

Simple interface design makes the users feel "in control".

Many participants were impressed by the simplicity of the interface. On the one hand this is due to the fact that the current implementation only demonstrates the essential features of a composition environment, while with only these features the composers can hardly proceed further. On the other hand, it also reflects a challenge of the current professional composition environments. To allow the users to be able to quickly orient themselves with the interface, the advanced features should be masked while being still easily reachable. Many users are often discouraged at the initial trail with those environments, as they do not know how to perform the most fundamental tasks.

Two-dimensional interface fuels contingency

The interface design is inspiring.

One participant pointed out that he and many composers in his acquaintance would assign random beats intending to be inspired by the instrument, when running out of ideas. He thought that the two dimensional circular interface offers an even better environment to satisfy such a need.

Compatibility with Other Tools

Many participants were passionate about the Pattern Editor and considered that it would benefit many computer musicians as a Plug-in.

Long-term observation of the users' behaviors would be valuable.

From the participants' feedback we are convinced that the system is beneficial to composers in many ways. However, due to the limited time of evaluation, there are some aspects, which could not be examined. Also, the developed habits of many participants impeded them from fully exploring the capability of this environment. We believe a long-term observation would be worthwhile. For example, we might find out the composers' tendency of grouping their musical ideas and further model such a behavior. Moreover, we could explore how the circular visualization of patterns influences the way people perceive a single pattern as well as the interrelationships among a set of subpatterns.

Chapter 10

Summary and Future Work

10.1 Summary and Contributions

Supporting creative workflow in music is an intricate task. As making music could be rather subjective, it is challenging to come up with an environment capable of assisting the majority. To achieve this task we started by researching how humans perceive audio as music, to support our viewpoints. Then we analyzed the current approaches to creative composition as well as the research addressing this area. In order to understand the real practice of the composition process, we conducted field interviews to observe the composers working with their tools. In the field we discovered many common issues of the software tools in terms of assisting creative composition. With this concrete motivation, we designed the first prototype and ran numerous iterations of the DIA cycle to refine it. We gained much understanding of the actual needs during each analysis phase, which influenced our system design significantly. Finally, we conducted a system evaluation to verify the applicability of the proposed concepts.

We have broken the convention of physical feature centered composition over desk-top interfaces by integrating the concept of musical semantics into the composition en-

vironment. The musical elements are grouped so that they can be easily and effectively accessed.

Our system is tailored to assist the tentative approach of creative composition. The composers can easily evolve their musical elements over the interface and shift between different levels of their musical concepts to perform ideation, representation and evaluation.

Moreover, to concretely demonstrate our concepts, we have demonstrated how composing rhythms could be easily performed using the circular interface, and how could the system actively assists the composers in organizing their musical material. Also, during the creative process, the system visualizes the related rhythmic features to the composers, which greatly enhances the effectiveness of the composition workflow.

Nevertheless, the system is by no means complete. It shall be able to integrate the other instruments into the composition environment. To reflect the musical semantics represented by other instruments to the system, the idiosyncrasies of those instruments must be considered and properly modeled.

10.2 Future Work

As DISCO is a vertical prototype of the composition environment we envision, in the future we shall focus on implementing the system horizontally.

10.2.1 Pluggable

A complete composition environment

To make this system as a complete environment assisting creative workflow, the capability to incorporate the other instruments is necessary. We could set up an ”instrument dock” for the users to specify the instruments they want to use, and similar to the current approach, the composers can group the material based on their musical concepts.

Furthermore, as long as the input could be converted into MIDI notes, the concept of variation propagation could also be employed for the other instruments. In this mode, the Pattern Editor would visualize the interface of the external instruments rather than the disk editor. As mentioned in 3.2.5—"Some other commercial circular interfaces", one of the flaws of the software instruments is that they are not integrated into the composition environment and could hardly be used effectively. Though we do not intend to modify the visual representation of the external instruments, with our approach all inputs could be organized semantically and reused effectively.

10.2.2 Post Production, Mixing

When the composer has completed his work, he needs to adjust the musical elements based on their sonic features, meaning he then needs the typical track-based environment for mixing the material. Based on the abovementioned environment, this could be easily achieved by organizing the musical elements vertically via their identification from the "instrument dock" (see 10.2.1—"Pluggable"), based on time horizontally.

10.2.3 Metaphors for the Grouping Functionality

The current implementation visualizes all time blocks on Semantic Composer identically. When a time block overlaps another to form a group, many users consider intuitively that the bottom block would be overwritten. We could visualize the bottom block as a container, or the upper block as a mask attached to the bottom to give the user a more concrete idea of the interrelationships between the items within a group. However, we need to trial the proposed metaphors with the users to see their applicability.

10.2.4 Plug-In

To benefit more composers, it is necessary to make the Pattern Editor into an Audio Unit to allow to be used in the other composition environments.

10.2.5 Expressiveness of Rhythms

There are already many software drum machines addressing on this issue. They set up several rhythmic functions to allow the user to configure the groove. These functions employ the algorithms that simulate the time latency of a rhythmic style, such as ”swing”. An example algorithm could be found Bilmes [1993].

Furthermore, we could allow the user to adjust their groove visually. Whenever the user wishes to adjust the expressiveness of the rhythm, we could show a scale on the rim of the disk for the user to slightly offset a step, or adjust the resolution of one step.

10.2.6 Collaboration and Live Setting

The concept of variation propagation to patterns composition could be employed as a shared resource among a group of improvisers. The collaborators could easily keep track of the other collaborators' musical material via the propagated visual cues. Therefore, even novice users could make musical rhythms as long as they follow the guidance from the other experienced collaborators. Furthermore, as demonstrated in 3.2.2—“Jam-O-World”, the disk rotation feature could enable the users to make variable rhythms without changing the beat arrangement of the rhythms.

10.2.7 Advanced Features for the Disk Interface

Loop Offsetting
Offsetting a loop is unnatural in the typical linear repre-

sentation as there is a discontinuity within the loop. On the other hand, in the circular representation, the users can easily perform the task by rotating the disk, or using a start and end pointer to specify the desired ambits of the loop. Furthermore, the time pointers could also solve the issue of setting the loop length by changing the number of steps. However, the circular concept is then somehow missing because of the jump between the start and end pointers.

10.2.8 Standard Features

Import and Export function for Patterns

We shall allow the users to export and import patterns so that the musical material is even more transportable.

Fundamental Facilities

We shall add the fundamental facilities such as attack, delay, sustain, and release for editing the sound features. The challenge is how to visualize them in the circular interface, which could be easily understood and manipulated. As demonstrated in 3.1.1—"Hyperscore", and 3.1.5—"Ableton Live", the controls could be masked with an abstract layer, while they are still easy to be operated.

Preview/ Mute/ Solo an Audio Clip

For both Semantic Composer and Pattern Editor, as the number of musical elements grows, it gets harder to distinguish between the sound sources. Preview allows the user to listen to a specific audio item by clicking on it. Moreover, it makes composition distractive when all outputs are present. Mute and solo functionality enables the user to disable a subset of the audio items, which are displayed simultaneously.

Preset Instruments

In order to allow novices to easily start with composition, the presets of popular drum kits shall be included into the system.

Appendix A

Storyboards

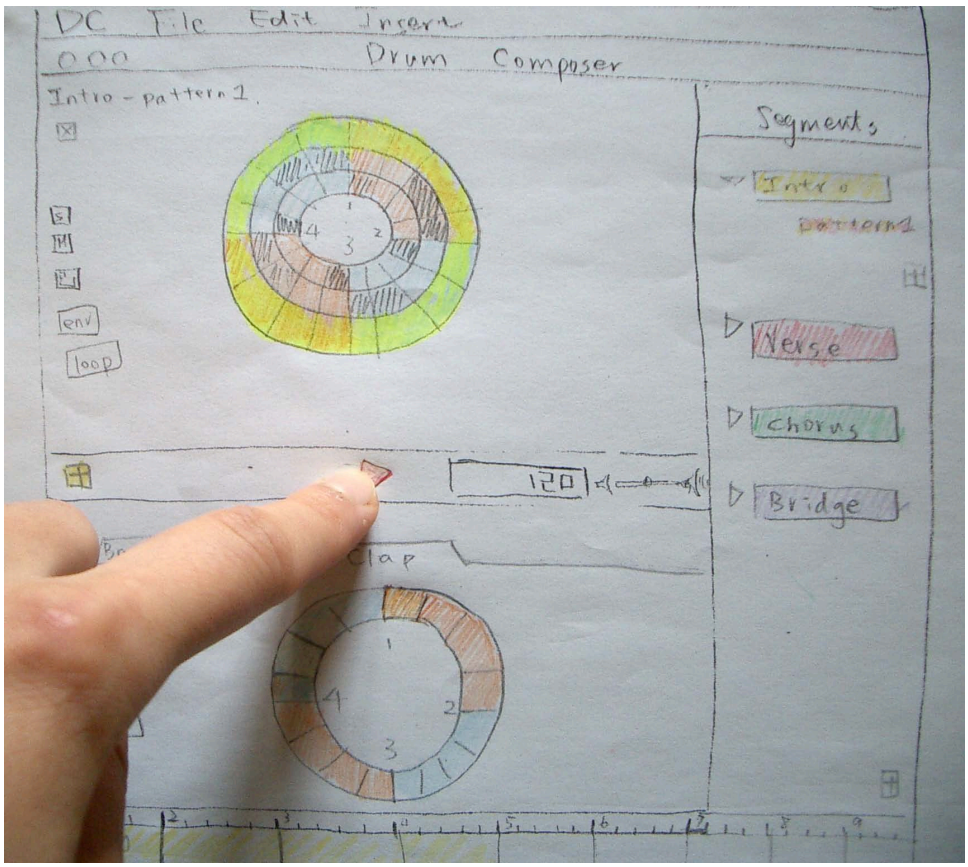

Figure A.1: Pressing the play button to listen to the resulting pattern

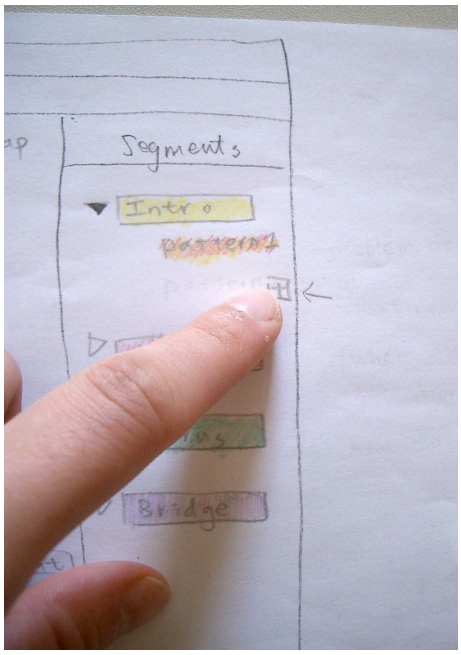

Figure A.2: Creating an extended pattern

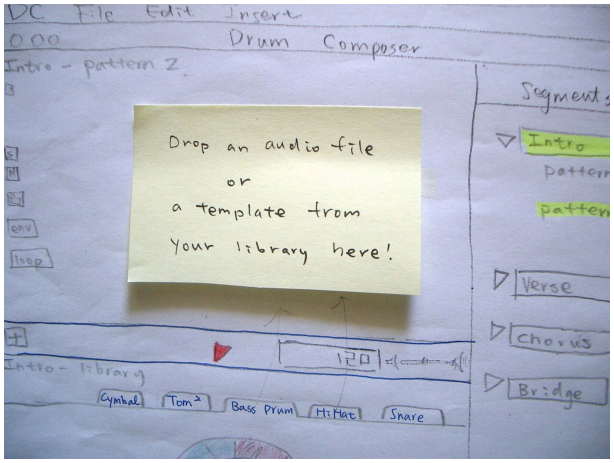

Figure A.3: The system informs the user to add new musical material.

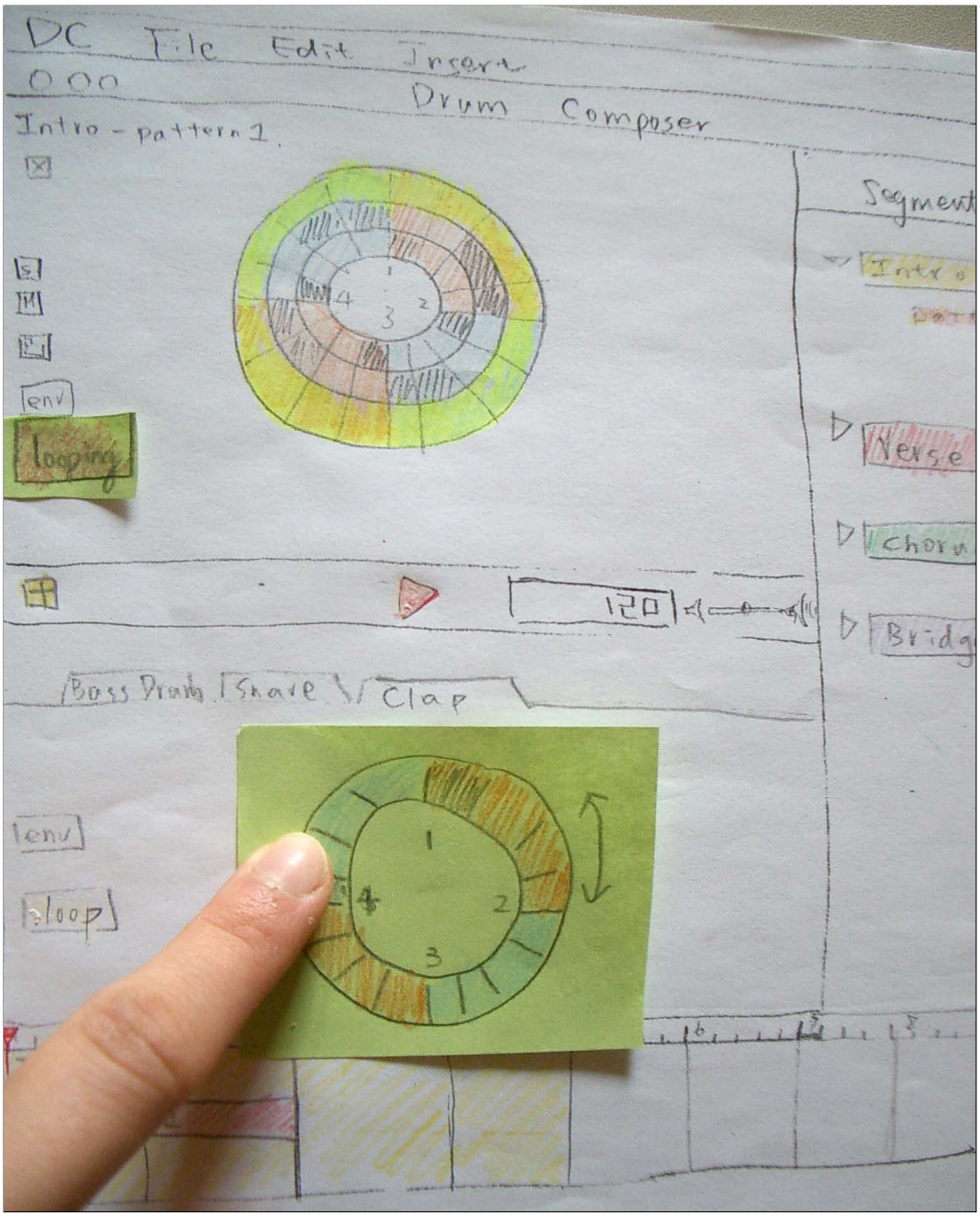

Figure A.4: Offsetting a disk in the library view

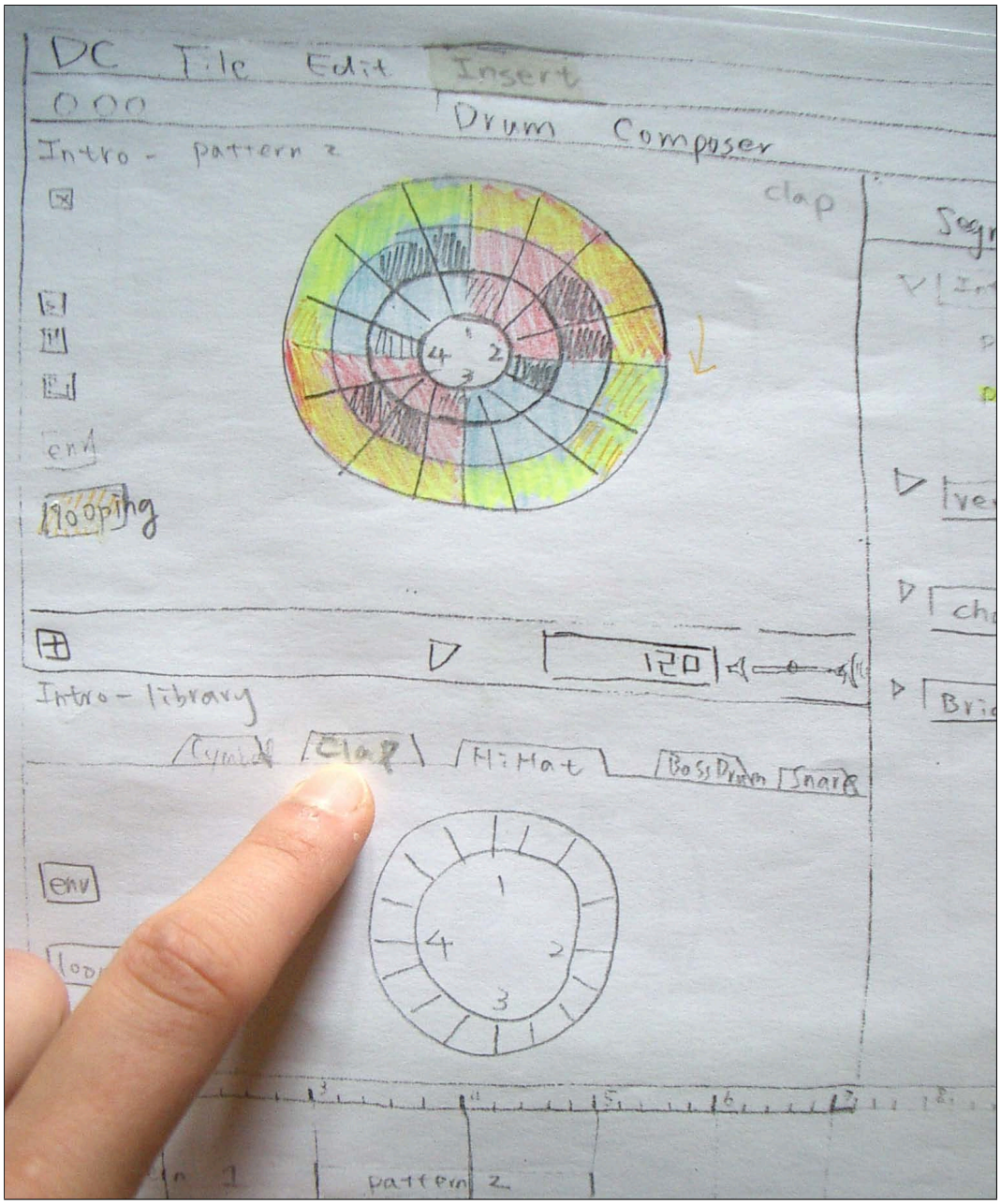

Figure A.5: Previewing the template patterns in the library

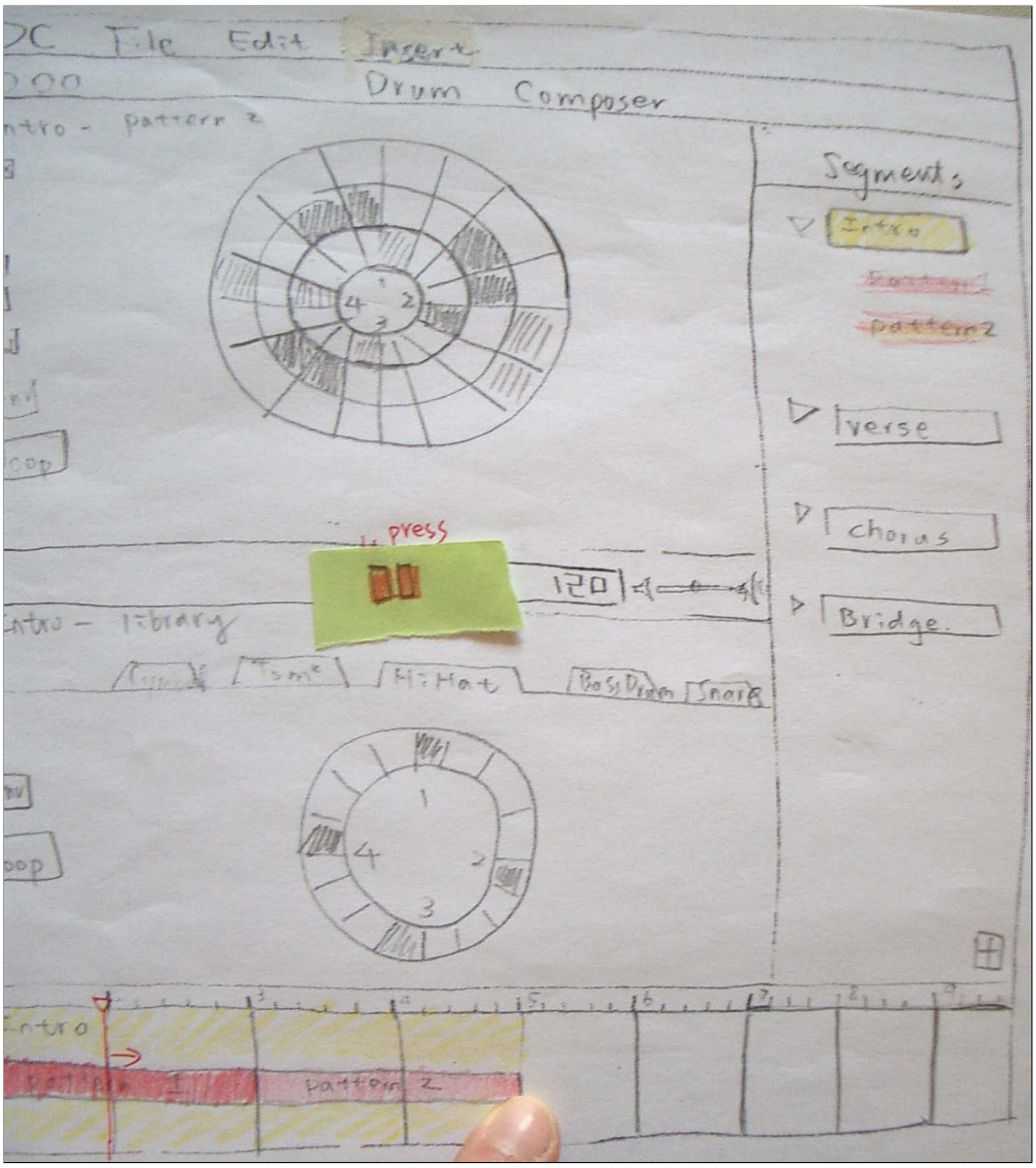

Figure A.6: Adjusting the time span of a pattern

Appendix B

Questionnaire

Questionnaire- USER TEST DISCO
Date/Time:

Personal Info

Age:

Gender:

Occupation:

Favored music styles:

1. Do you play any music instruments? If yes, please list the instruments you play, and for how long have you been playing it.

2. Do you use any musical software? Please list your favorite ones (or the ones that you are familiar with), and for how long have you been using them.

- What do you like about them for composing music?

- What do you dislike about them for composing music?

3. Do you compose music? Please describe your musical style and the tools or instruments you use.

- For how long have you been composing music?
4. If you compose music, please describe your composition process.

Circular interface – Rhythm composition
1. How do you rate the division of beats?
Easy to understand +2 ☐ +1 ☐ -1 ☐ -2 ☐ Hard to understand
Reasons?

Figure B.1: The questionnaire used in the final evaluation - Page 1

2. How do you perceive the relationship between the length of a loop and the number of steps? (i.e. the more the steps, the longer the loop)
Easy to understand +2 ☐ +1 ☐ -1 ☐ -2 ☐ Confusing

Reasons?

3. Does the circular layout reflect how you think about patterns?
Does reflect +2 ☐ +1 ☐ -1 ☐ -2 ☐ Does not reflect
Reasons?

4. How do you rate the variation concept?
Helpful +2 ☐ +1 ☐ -1 ☐ -2 ☐ Not helpful

Reasons?

5. Which concept do you prefer for rhythmic composition:
1. pattern-dependent interface (where modifications to the original are propagated to its variations; as demonstrated)
2. pattern-independent interface (where patterns are created independently; such as the Redrum drum machine from Propellerhead Reason)
3. undecided
Reasons?

Figure B.2: The questionnaire used in the final evaluation - Page 2

Composer – Song structure construction

1. How do you rate the way of *grouping* patterns in the semantic composer?
Helpful +2 ☐ +1 ☐ -1 ☐ -2 ☐ Not helpful
Reasons?

2. Does the arrangement of patterns inside the semantic composer reflect how you perceive your song structure?
Yes +2 ☐ +1 ☐ undecided ☐ -1 ☐ -2 ☐ No
Reasons?

3. Do the following features have an impact on your ability to ***experiment*** with the music?
- Circular Interface
It encourages me +2 ☐ +1 ☐ No Impact ☐ -1 ☐ -2 ☐ It discourages me
Reasons?

- Propagation of variations
It encourages me +2 ☐ +1 ☐ No Impact ☐ -1 ☐ -2 ☐ It discourages me
Reasons?

- Grouping patterns over the timeline
It encourages me +2 ☐ +1 ☐ No Impact ☐ -1 ☐ -2 ☐ It discourages me
Reasons?

4. Any other suggestions to the project?

Thank you for your participation!

Figure B.3: The questionnaire used in the final evaluation - Page 3

Appendix C

DISCO in Use

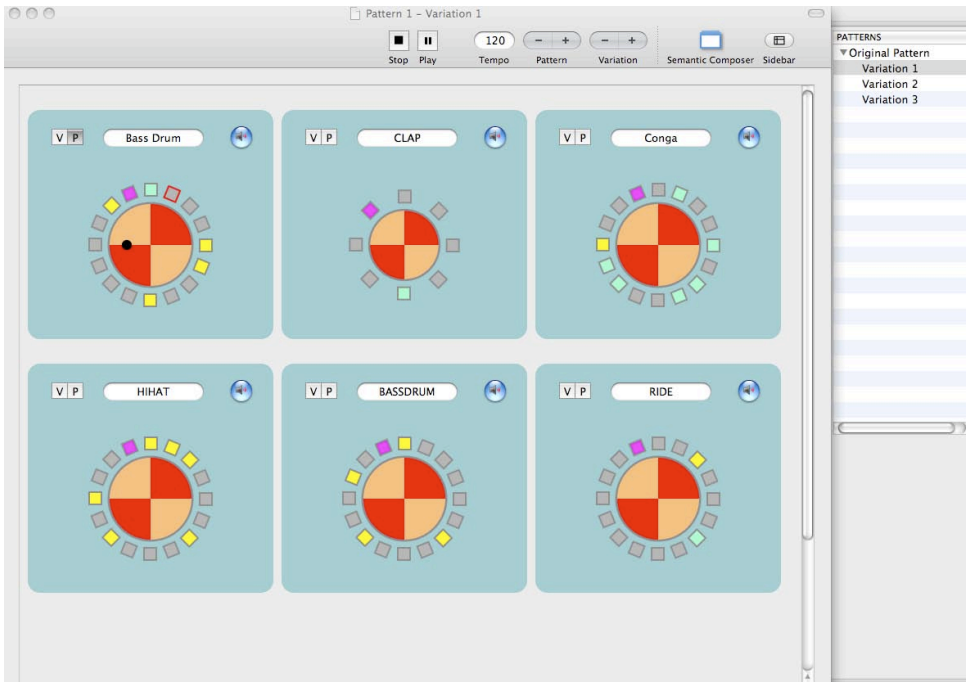

Figure C.1: The Pattern Editor

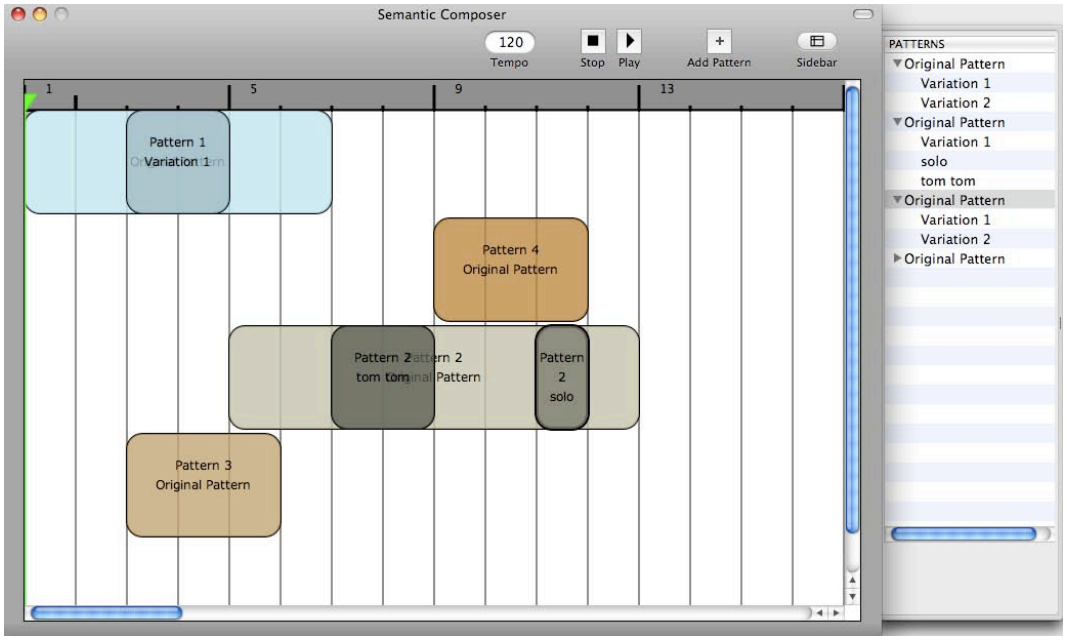

Figure C.2: The Semantic Composer

Bibliography

Steven Abrams, Daniel V., Oppenheim, Don Pazel, and James Wright. Higher-level composition control in music sketcher: Modifiers and smart harmony. In *ICMC '99: Proceedings of the 1999 International Computer Music Conference*, 1999.

Steven Abrams, Ralph Bellofatto, Robert Fuhrer, Daniel Oppenheim, James Wright, Richard Boulanger, Neil Leonard, David Mash, Michael Rendish, and Joe Smith. Qsketcher: an environment for composing music for film. In *C&C '02: Proceedings of the 4th conference on Creativity & cognition*, pages 157–164, New York, NY, USA, 2002. ACM. ISBN 1-58113-465-7. doi: http://doi.acm.org/10.1145/581710.581734.

Peter Holdridge Barry Eaglestone, Nigel Ford and Jenny Carter. Are cognitive styles an important factor in the design of electroacoustic music software? In *ICMC '07: Proceedings of the 2007 International Computer Music Conference*, volume 1, pages 466 – 473. ICMC, 2007.

Olav W. Bertelsen, Morten Breinbjerg, and Soren Poldoren Pold. Instrumentness for creativity mediation, materiality & metonymy. In *C&C '07: Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pages 233–242, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-712-4. doi: http://doi.acm.org/10.1145/1254960.1254992.

Hugh Beyer and Karen Holtzblatt. *Contextual design: defining customer-centered systems.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. ISBN 1-55680-411-1.

J. Bilmes. Techniques to foster drum machine expressivity. In *ICMC '93: Proceedings of the 1993 International Computer Music Conference*, 1993.

Tina Blaine and Clifton Forlines. Jam-o-world: evolution of the jam-o-drum multi-player musical controller into the jam-o-whirl gaming interface. In *NIME '02: Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore. ISBN 1-87465365-8.

Tina Blaine and Tim Perkis. The jam-o-drum interactive music system: a study in interaction design. In *DIS '00: Proceedings of the 3rd conference on Designing interactive systems*, pages 165–173, New York, NY, USA, 2000. ACM. ISBN 1-58113-219-0. doi: http://doi.acm.org/10.1145/347642.347705.

A. Bregman. *Auditory Scene analysis*. The MIT Press, Cambridge, Mass.,, 1991.

N. Bryan-Kinns. Daisyphone: the design and impact of a novel environment for remote group music improvisation. In *DIS '04: Proceedings of the 5th conference on Designing interactive systems*, pages 135–144, New York, NY, USA, 2004. ACM. ISBN 1-58113-787-7. doi: http://doi.acm.org/10.1145/1013115.1013135.

N. Bryan-Kinns and P. G. T. Healey. Daisyphone: support for remote music collaboration. In *NIME '04: Proceedings of the 2004 conference on New interfaces for musical expression*, pages 27–30, Singapore, Singapore, 2004. National University of Singapore.

Mike Collins. *Choosing and Using Audio and Music Software: A Guide to the Major Software Applications for Mac and PC*. Elsevier, 2004.

Tim Coughlan and Peter Johnson. Interaction in creative tasks. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 531–540, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: http://doi.acm.org/10.1145/1124772.1124854.

Morwaread M. Farbood, Egon Pasztor, and Kevin Jennings. Hyperscore: A graphical sketchpad for novice composers. *IEEE Comput. Graph. Appl.*, 24(1):50–54, 2004.

ISSN 0272-1716. doi: http://dx.doi.org/10.1109/MCG.2004.1255809.

Arthur Hull. *Drum Circle Spirit*, pages 17–42. White Cliffs Media Tempe, 1998.

Markus Jonas. Tactile editor - a prototyping tool to design and test vibrotactile patterns. Mastersthesis, RWTH Aachen University, Aachen, Germany, May 2008.

Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-619-6. doi: http://doi.acm.org/10.1145/1226969.1226998.

Ole Kühl. *Musical Semantics*. Peter Lang Publishing, 2007a. ISBN 3039112821.

Ole Kühl. An introduction to musical semntics: The gesture in modelization. In *ICMC '07: Proceedings of the 2007 International Computer Music ConferenceProceedings of the 2007 International Computer Music ConferenceInternational Computer Music Conference*, 2007b.

Ole Kühl. A semiotic approach to jazz improvisation. *The Journal of Music and Meaning*, 2007c.

P. Barr M. Duignan, J. Noble and R. Biddle. Metaphors for electronic music production in reason and live. In *APCHI '04: Proceedings of the 6th Asia Pacific Conference Human Interaction*, volume Volume 3101/2004. Springer Berlin / Heidelberg, 2004.

Donald A. Norman. *The Design of Everyday Things*. Basic Books, 1988.

Jeff Raskin. *The Humane Interface*. Addison-Wesley, 2000.

R.J. Riding and Cheema. Cognitive styles - an overview and integration. *Edcational Psychology*, 11:193 – 215, 1991.

Sawyer and R. K. Group creativity: Music, theatre, collaboration. *Lawrence Erlbaum*, 2003.

Ben Schneiderman. *Leonardo's Laptop: Human Needs and the New Computing Technologies*. The MIT Press, 2002. ISBN 0-26219-476-7.

Carolyn Snyder. *Paper Prototyping*. Morgan Kaufmann Publishers Inc., 2003. ISBN 1-55860-870-2.

E. Tulving. How many memory systems are there? *American Psychologist*, 40:385–398, 1985.

James Wright, Daniel V., Oppenheim, and David Jameson. Cyberband: A "hands-on" music composition program. In *ICMC '97: Proceedings of the 1997 International Computer Music Conference*, 1997.

Index

Typeset October 6, 2008