TENOR
2018

Montréal, Canada – May 24-26, 2018

Fourth International Conference on Technologies for Music Notation and Representation

Edited by:
Sandeep Bhagwati
Jean Bresson

ISBN 978-1-5251-0551-7

International Conference on Technologies for Music Notation and Representation – TENOR'18
Montreal, Canada
May 24-26, 2018

Editors: Sandeep Bhagwati, Jean Bresson

Proceedings published by: Concordia University
ISBN: 978-1-5251-0551-7

Editorial coordination: Jean Bresson
Logo design: Nicolas Taffin

PREAMBLE/POSTLUDE

TENOR 2018 was the 4th international TENOR conference, and the first TENOR conference on North American soil. The conference took place at each of the three co-hosting universities: day 1 at CIRMMT-McGill University, day 2 at Hexagram UQAM and day 3 at Concordia University.

CALL FOR PAPERS

The call for complete papers yielded 33 submissions, out of which 26 were accepted for publication and presentation. These were grouped into 6 thematic sessions: Listening, Comprovisation, Music Representation & Analysis, Notation Tools, Shapes, and Scoring.

New Collaborations. The conference, as it happened, was experienced as an organizational and as a research success – it broadened the scope of TENOR, and it brought the research community closer together in terms of inter-operability, shared terminology and understanding around common issues and questions. Being able to pursue the work of several research groups over time also has generated new collaborations, one of which is the newly funded TENOR Network hosted at Concordia University and funded by the Social Sciences and Humanities Research Council of Canada, which is expected to unite and coordinate the research efforts of 16 participating institutions worldwide, as well as expand the scope of TENOR activities both locally and globally.

New Conference Elements. TENOR 2018 also introduced two new elements into the conference, both of which were deemed an unmitigated success and are expected to become a standard feature of future TENOR conferences: a separate track with a call for scores, and a Performer Lab.

Geographical Clustering. The provenience of speakers and research groups was international but showed a very unequal geographical distribution: France (8), UK (4), Australia (4), Canada (4) were well represented, while the USA, Sweden, Belgium, Germany, Italy, Portugal, Hong Kong, and Lebanon were represented by one speaker each. This reflects the distribution of papers submitted. The origins of the TENOR conference lie in France – the strategic hope that holding the conference in North America would attract more North American researchers (particularly the US American scene) was unfortunately not fulfilled, despite extensive canvassing. The same applies to the explicit call for notations and representations of non-eurological music traditions – only one of the accepted papers addressed Arabian music. It is hoped that future conferences will show less geographical clustering and a more geographically and culturally balanced distribution of research teams.

Gender Imbalance. Another concern was gender balance. While the conference organizers had announced two female keynote speakers in the hope of being more inviting to women researchers and artists, the response to the call for papers still was heavily unbalanced, and this is reflected in the papers accepted (2 women-led papers out of 26). Moreover, out of 7 accepted women authors and co-authors, only 2 could come to Montréal to present or attend the conference. One hope of the organizers was that including a keynote speaker from dance and an explicit call for scores and representations from dance, theater and other related arts would also boost participation by women – but that did not materialize either, again despite extensive canvassing.

TENOR SCORE CALL

In addition to the call for papers, we also issued a call for TENOR-related scores which attracted 45 submissions. The jury for the score call was composed of the artistic directors of each of the 4 concerts: Eldad Tsabary (CLOrk), Sandeep Bhagwati (matralab), Terri Hron (Performer Lab), Cléo Palacio Quintin (Supermusique), plus representatives of the co-organizing universities: Robert Hasegawa (McGill) and Jean Décarie (UQAM). The jury chose a short-list of 15 scores. Each ensemble chose a set of scores that they would explore with a view to performance. A period of practical negotiation followed, centering on questions of technology, available performers etc.

From this call, the following scores were chosen for performance during TENOR 2018:

CLOrk Concordia Laptop Orchestra (Concert on May 24, 2018)
Karlheinz Essl – *more or less*
Amy Brandon – *Hidden Motive*
Justin Yang – *Musicmobile I*
Kristina Warren – *Listening Not Guaranteed*

matralab and Ensemble ILEA (Concert on May 25, 2018)
Ryan Ross Smith – *Study No. 38*
Kevin Gironnay – *Shaping Freedom within Experimental Improvisation*
Elisabeth Schimana – *Virus for Percussion*
Georg Hajdu – *Swan Song for Violoncello, Percussion and Live-Audiovisual Environment*

Performer Lab (Concert on May 26, 2018)
Jef Chippewa – *something like this but not this and not that either*
Sebastian Adams/ Carl Ludwig Hübsch – *deciphering*
Jonathan Bell – *And the sea*

Ensemble Supermusique (Concert on May 26, 2018)
Ciaran Frame – *Thallus*
Marta Tiesenga – *Vik*

Scores selected by the jury, but not performed:
Kalun Leung – *Twisted Twister*
Jacob Sello – *Catch_Up 4.5*

Each ensemble could also choose to add other works to complete their programming. Works by Cat Hope, Sandeep Bhagwati, Nicholas Ryan, Joane Hétu and Danielle Palardy-Roger were thus performed as part of the concert series during TENOR 2018.

PERFORMER LAB

TENOR 2018 also introduced the performer lab - an offer to the community of music performers. The lab was offered as a course with a participation fee of $150. In addition, potential participants were assessed on their level of instrument competence. Finally, 7 participants were chosen. In this lab, the participants were introduced to the performance of situative, technological scores by composer and Terri Hron, who in turn invited several guest lecturers. Each of the composers whose score had been chosen from the call also introduced their own score. Over 5 days, each comprising 6 hours of workshop, the group prepared 4 technological and situative scores for performance.

WORKSHOPS

Following-up with the tradition of previous TENOR editions, two workshops of 3 hours each, were proposed to the participants: *Working with MaxScore*, by Georg Hajdu, and *An Introduction to the bach family* by Daniele Ghisi.

KEYNOTES

Finally, three keynotes were presented at the conference: by composer and "veteran" TENOR researcher Cat Hope (Monash University, Melbourne), by ground-breaking and internationally renowned musician Elliot Sharp (New York) who has experimented with scoring techniques since the 1980s, and by dance notator and historian Valarie Williams (University of Ohio, Columbus). The abstracts of their talks have been added to the proceedings.

AUDIOVISUAL DOCUMENTATION

All talks and concerts of this conference have been audiovisually documented. Links to view a selection of these materials (authors have the right to veto this form of dissemination) will be made available to the general public on the TENOR 2018 website http://matralab.hexagram.ca/tenor2018/ from September 2018 onward.

Sandeep Bhagwati
TENOR'18 Conference Chair
Berlin, June 3, 2018

ORGANIZING COMMITTEE

Sandeep Bhagwati (*General Chair*, Concordia University / matralab)
Eldad Tsabary (Concordia University)
Robert Hasegawa (McGill University / CIRMMT)
Ichiro Fujinaga (McGill University)
Jean Décarie (UQAM / Hexagram Network)
André Eric Letourneau (UQAM / Hexagram Network)
Christopher Salter (Concordia University / Hexagram Network)

SCIENTIFIC COMMITTEE

Hasnizam Abdul Wahid (Universiti Malaysia Sarawak, Malaysia)
Andrea Agostini (composer, France)
Trevor Bača (Harvard University, USA)
Sandeep Bhagwati (Concordia University, Canada)
Alain Bonardi (Université Paris 8, France)
Jean Bresson (*Program Chair*, IRCAM / UMR STMS, France)
Pierre Couprie (Université Paris-Sorbonne / IReMus, France)
Ricardo Dal Farra (Concordia University, Canada)
Elsa De Luca (CESEM, Universidade Nova de Lisboa, Portugal)
Dominique Fober (Grame, France)
Jason Freeman (Georgia Institute of Technology, USA)
Ichiro Fujinaga (McGill University, Canada)
Yann Geslin (INA-GRM, France)
Daniele Ghisi (IRCAM / UMR STMS, France)
Jean-Louis Giavitto (IRCAM / UMR STMS, France)
Rama Gottfried (CNMAT, USA / HfMT Hamburg, Germany)
Gérald Guillot (University of Teacher Education Lausanne, Switzerland)
Karim Haddad (IRCAM / UMR STMS, France)
Georg Hajdu (HfMT Hamburg, Germany)
Keith Hamel (University of British Columbia, Canada)
Richard Hoadley (Anglia Ruskin Universit, UK)
Robert Hoeldrich (University of Music and Performing Arts Graz, Austria)
Cat Hope (Monash University, Australia)
David Kim-Boyle (University of Sydney, Australia)
John MacCallum (LRI, Inria/Université Paris-Sud/CNRS, France)
Thor Magnusson (University of Sussex, UK)
Chris Nash (University of the West of England, UK)
Stefan Östersjö (ORCiM-Orpheus Institute, Belgium)
Helena Palma (University of A Coruña, Spain)
Eleanor Selfridge-Field (Stanford University, USA)
Ryan Ross Smith (State University of New York at Oneonta, USA)
Alice Tacaille (Université Paris-Sorbonne / IReMus, France)
Lindsay Vickery (Edith Cowan University, Australia)
Anders Vinjar (composer, Norway)

ARTISTIC COMMITTEE

Sandeep Bhagwati (*Chair*, Concordia University / matralab)
Robert Hasegawa (McGill University / CIRMMT)
Jean Décarie (UQAM / Hexgram)
Cléo Palacio-Quintin (Ensemble Supermusique)
Danielle Roger-Palardy (Ensemble Supermusique)
Joanne Hetu (Ensemble Supermusique)
Eldad Tsabary (CLOrk)
Terri Hron (Performer Lab)

CONFERENCE STAFF

Conference Coordinator: Jen Reimer (matralab)
Technical Coordinator: Joseph A Browne (matralab)
Web: Theo Stojanov, Max Stein (matralab)

Stephanie Merulla (catering)
Öyku Önder (light design)
Kaja Levy, Mehdi Rahimdokt, Dan Harvey (matralab)
Jason Pomrenski (UQAM)
Sylvain Pohu (CIRMMT)
Kevin Moon (Concordia)

TENOR STEERING COMMITTEE

Dominique Fober
Sandeep Bhagwati
Jean Bresson
Pierre Couprie
Yann Geslin
Richard Hoadley

http://tenor-conference.org

KEYNOTE (I)

Cat Hope (Monash University, Melbourne)

Why Music Needs Graphic and Animated Notations Now

Graphic and animated music notations enable contemporary musicians to make music in ways that are more in keeping with contemporary performance practices that are increasingly more collaborative and polystylistic. Whilst contemporary music practice continues to expand into a wide array of styles and techniques, it could be argued that music notation has not evolved at the same pace. While traditional, so called common practice notation has remained the most appropriate way to communicate information about tempered harmony and subdivision of meter, graphic and animated notation can provide an opportunity for the representation of actual sounds and their communication from composer to performer. This is particularly useful when engaging in micro-tonality, pulse-less music, non-linear structures, improvisation, aleatoricism, interactivity and electronic music. Importantly, these notational approaches can be designed to enable increased input from performers from any musical genre, reflecting the collaborative practices that are a signpost of current music practice. Due to its asemic potential, graphic notation can be used to direct the musicianship of performers in very different ways to traditional notation – add to this ability to the generative and interactive capacity of computer programs and online collaborations, and we have the opportunity to redefine the parameters of music notation.

This presentation examines the potential for digitally facilitated graphic and animated notation to expedite contemporary music making and, more broadly, to foster collaboration between musicians and composers of different musical genres and cultures worldwide. Performers and composers are seeking out collaborations across different musical styles, requiring a form of music notation that conveys the fundamental principles of music understanding such as pitch, texture, dynamic, time and their nuance. The idea of common practice notation as a kind of universal musical truth is well overdue to be challenged, as it serves a decreasing slice of the music that most people relate to. Whist it may suit the silo-ed music of the past, it does not have capacity to bring musicians together in the future, or record the practices of today. This paper proposes alternatives, highlighting the benefits and potential of what could be a new era for music notation.

KEYNOTE (II)

Elliott Sharp (New York)
New Strategies for Music Notation

In this address, composer Elliott Sharp describes his long-term resonance with graphic approaches and how they have evolved to form his current strategies. Sharp's latest works in this realm are Foliage and Sylva Sylvarum, both graphic scores in the form of animated movies.

A score may be a detailed roadmap for performers but it may also operate in a less literal fashion. Music strictly determined in pitch and time is well represented by traditional Western notation while other musics require a visual manifestation that reflects inner workings governed by other parameters than harmony and melody: density, texture, flow. This translation of creativity from Inner Ear to sonic output may be affected by many factors and introducing a visual element exploits the porosity between these modes of expression. Sharp speaks of synesthesia as inextricably linked to his use of graphic notation, both as part of the generative process and in his own experience of realizing such scores.

Sharp first used graphic scoring in 1972 to address problems encountered in free group improvisation. He soon expanded this approach in 1974's Hudson River Compositions to incorporate algorithmic concepts emulating natural forms and processes. In subsequent works, Sharp utilized a variety of alternative notations to catalyse sonic activities in diverse ensemble situations. A significant breakthrough in strategy occurred in 2007 during the creation of the string quartet Seize Seeth Seas Seen when Sharp first composed fragments in musical notation, exported them as TIF graphic files, and then subjected them to processing in the graphic editing software Photoshop. The images were inverted, stretched, filtered, modulated with various waveforms, subjected to feedback, and otherwise distorted in a manner similar to methods used by Sharp in live mixing of instrumentalists performing his compositions. For a generation of musicians raised on sonics, texture, densities, sound editing on computer, and graphic notation of all types, interpreting Seize Seeth Seas Seen and subsequent scores was a clear and natural process: the music sounded the way it looked. The score clearly revealed its derivation from musical notation while manifesting its own visual identity both as instruction set and retinal art.

KEYNOTE (III)

Valarie Williams (University of Ohio, Columbus)

What Goes into the Score?
An integrated collaborative approach to document and preserve movement

Throughout time the documentation of movement is elusive. The idea of preserving dance and its contexts through various means gives way to numerous methods to record movement including multiple notation systems that are representative of movement and style. The notator, and subsequent director of those notated scores, tells a story of the dance utilizing aspects of representation and preservation. Over time, the notator changes and grows as he or she records and analyzes movement. To highlight the learning of the notator and director within a particular environment, a dynamical systems approach will serve as a framework for looking at several case studies of directing and notating movement based on the following five principles: a) notators are self-learning; b) their learning changes the entire system, not just the particular behavior being perturbed (Kelso, 1999); c) notators have an innately adaptive capacity which means they can change the ways in which they know, perceive, and act; d) individuality and agency derive from the fact that notators and directors are operationally autonomous, self-organized systems coupled with the environment (the studio and the score); and e) the process of implicit self-learning accounts for both self-individuation and self-change.

TABLE OF CONTENTS

AUTHOR INDEX

COMBINING SOUND- AND PITCH-BASED NOTATION FOR TEACHING AND COMPOSITION

Mattias Sköld
KMH Royal College of Music, Stockholm
KTH Royal Institute of Technology, Stockholm
mattias.skold@kmh.se

ABSTRACT

My research is concerned with finding a common notation for pitch-based, sound-based and spatialized music in an attempt to bridge the gap between acoustic and electronic music, also working towards the possibility of a holistic system for algorithmic composition based on music representation. This paper describes the first step towards this goal, focusing on the combination of pitch-based and sound-based musical structures, introducing a graphical notation system that combines traditional music notation with electroacoustic music analysis notation. I present how this was tested in practice in a case study within the framework of composition education at the Royal College of Music in Stockholm, where composition students were working with, and reacting to, the system.

1. INTRODUCTION

As a teacher of composition I have noticed that students often feel they need to take sides with regard to electronic and acoustic music despite encouragement to work in both fields—you're either a studio composer or a score composer. Being a composer of both acoustic and electronic music I believe this to be an unnecessary side effect of the difference in craft and music theory surrounding the two sound worlds—it has little to do with the creative talents of the composers or the possibilities of expression in the media themselves. More serious than the problem of students' aesthetic identity is how this difference in music theory makes combining acoustic and electronic sound sources difficult on a compositional level, part of the reason being that their music theories use different systems to express the same thing, e.g. frequency values and note names. However, translating individual frequencies into note names is easily done, as long as microtonality is considered. The main problem lies in the representation of non-pitched sounds—an important part of electroacoustic music expression. Granted, there is a rich tradition of non-pitch-based music in classical music too, starting with Varèse's *Ionisation* [1] which ranks among the first pieces for percussion ensemble alone in the Western art music tradition, reaching a milestone in Lachenmann's *musique concrète instrumentale* as expressed in *Pression for One Cellist* [2] where the idea of extended instrument techniques is taken to the extreme. However, the notation for this music, particularly in the case of Lachenmann, has little to do with the representation of sound but rather deals with the representation of actions resulting in sound. One strength of traditional staff-based music notation, beside its widespread use, is its double nature both as a means for describing pitch-based musical structures and for prescribing musical performance. Without this feature, traditional ear training would not be possible. Also, the most fundamental aspects of traditional pitch-based notation can be converted into MIDI data, making possible algorithmic composition with pitch-based instruments in mind. For non-pitched sounds, the notation of instrumental works tends to rely on tailor-made solutions such as written instructions or drawings of hands and objects over instrument bodies. Electroacoustic Music representation on the other hand tends to focus on timbre and sound classification, often loosing detail with regard to individual pitches despite Denis Smalley's remarks on the importance of intervals when pitches are present [3]. By combining electroacoustic music analysis notation with traditional notation of pitch, also introducing space as a parameter—another important aspect of electroacoustic music—I aim to bridge the gap between these different sound worlds, also making a new kind of algorithmic composition possible, where pitch-based, sound-based and spatialized music can be visualized, simulated and/or generated using sound synthesis and/or sample banks of concrete sounds.

2. BACKGROUND

Despite the genre's relative youth, electroacoustic music representation has already a long history beginning with Pierre Schaeffer's initial research into the description and classification of sound objects [4], followed by Denis Smalley's theories of Spectromorphology [3], introducing a framework for describing sound in transformation as well as spatialized sound. Lasse Thoresen, assisted by Andreas Hedman, builds on these ideas in *Spectromorphological Analysis of Sound Objects* [5] where they provide a well-developed notation system for the analysis of music as heard. In Pierre Couprie's overview of algorithms and digital technologies in music notation [6] we get a sense of the multitude of notation systems now available, all with different purposes, such as algorithmic notation, interactive notation etc. Some systems expand traditional notation

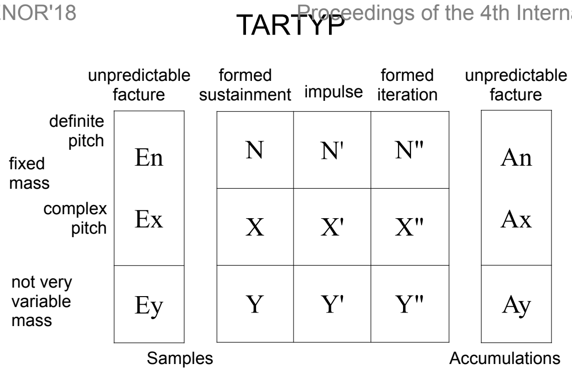


Figure 1. From a larger set of categories in Schaeffer's TARTYP [4], Thoresen and Hedman keep these categories and adapt them for graphic notation analysis [5].

while others look for completely new ways of communicating musical ideas. At the heart of most new systems of representation are their relations to artistic problems, such as the need to communicate with music-reading musicians over a network, or the need to communicate the structure of the spatialization of a piece. The artistic problem addressed in this paper is the problem of having two completely different sets of music theory for working towards the same kinds of concepts depending on whether there are electronic sounds present or not. Beside causing artistic problems, the music theory discrepancy described here also has an effect on how the same subject is taught to composers of different genres. The best example is ear training where traditional teaching relates sound to traditional notation while sound-based ear-training borrows from the audio engineering field, focusing on frequencies and measured amplitudes. That being said, there is a fundamental difference between a tone and noise so the challenge is to find a way for both to co-exist within one and the same system where the important aspects of both types of sound are taken into consideration.

3. THE NOTATION SYSTEM

Thoresen, assisted by Hedman, builds his analysis system on sound classification categories listed in Schaeffer's TARTYP table [4], with the starting point in the balanced micro object categories N, X, and Y, as well as the extreme macro-object categories, E and A (see Figure 1), providing notation symbols as well as several additional notation features to describe sounds in great detail. Thoresen with Hedman have renamed and adapted Schaeffer's categories for use with graphic notation and have created new intermediary categories to complete the system. One such category is the *dystonic* sound, a category between pitched and complex sounds (Schaeffer's N and X [4]) to denote clusters and inharmonic spectra. See Thoresen's and Hedman's article [5] for a concise description of their notation system.

For my first prototype of the notation system for pitch, sound and space I place Thoresen's and Hedman's symbols, slightly modified, over a fixed frequency grid, in this case a traditional staff-system, adding indicators for spatialization, notated with circles above the system. As with pitch, durations were also notated with traditional symbols. (see Figure 3). The basic changes to their system at this point had to do with taking advantage of the possibilities of indicating spectral information with fixed values, e.g. instead of showing spectral width with symbol indicators, I introduce the possibility to indicate the frequency range of a spectrum using a dashed vertical line with an arrow that clearly indicates the spectral space occupied by the sound, the arrows pointing towards tendencies of change in spectral width (see Figure 3). Beside minor changes to the notation itself, it was necessary to re-think the notation as symbols of actual measurable sound rather than phenomenological descriptions of a listening experience. One initial problem related to this was deciding exactly where to place the sound object symbols on the staves. For pitched sounds there's no reason to deviate from common practice of placing the symbol at its root frequency, but how about inharmonic sounds and noise?

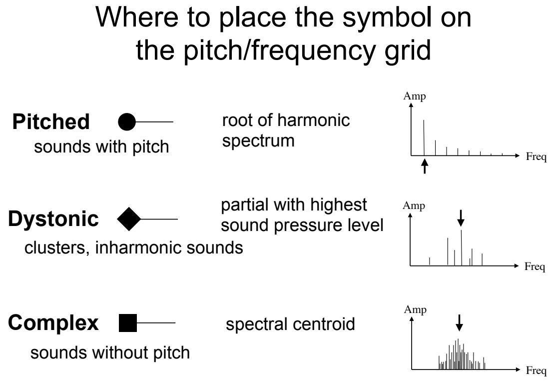


Figure 2. An image presented to the students to show the strategies for deciding at what pitch/frequency to place the symbol for the different kinds of sounds.

Figure 2 shows my solution to this problem with regard to the three basic sound spectrum categories. As in Thoresen's and Hedman's notation system for analysis [5], I encourage indicating additional partials or other sound components that are important for the identity of a sound. For the first prototype of the notation system I suggested a very simple indicator of spatialization where a sound's duration is indicated in the fashion of ambisonics 3D-panning software with two circles—one top view and one front view as can be seen in Figure 3. This was because we would not work specifically with spatialized sound for the course where I first tried the system, as explained below.

4. CASE STUDY

4.1 Introduction

For several years composition students at the Royal College of Music in Stockholm have been exposed to Lasse Thoresen's and Hedman's spectromorphological analysis notation [5] with the aim of bringing awareness to timbral structures in other works as well as their own. The course module, simply called *Sonology* and part of the

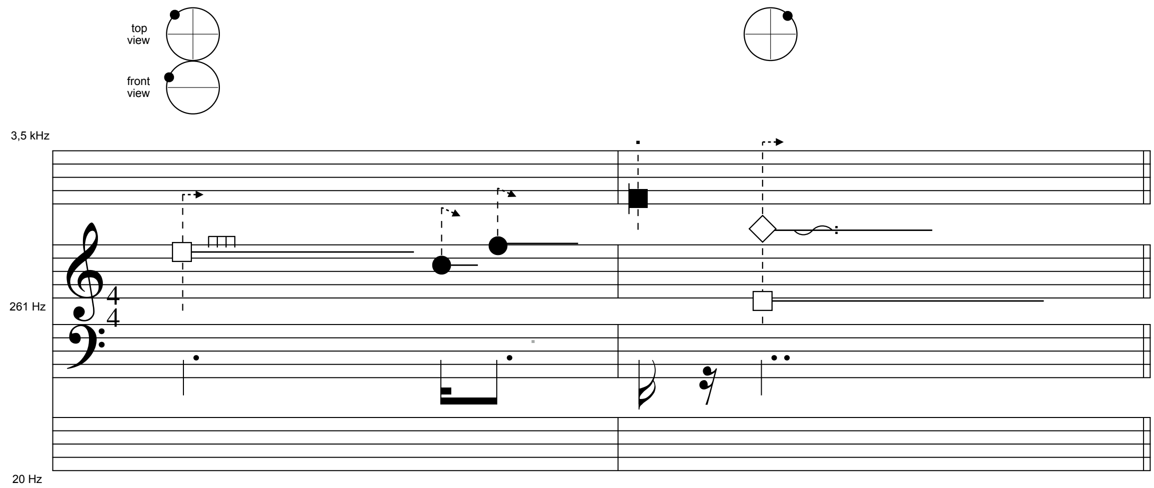


Figure 3. A notation example to demonstrate the combination of sound-based notation, traditional staff-systems to indicate pitch, and ambisonics-style indicators of spatialization above.

course *Sonology and Studio Technology* [7], introduces basic ideas from Schaeffer's typo-morphology [4] before introducing Thoresen's and Hedman's development of Schaeffer's ideas and how this is expressed as symbolic notation. The students work with both analysis and composition. This year, I introduced the ideas described here, providing the students with a new system where spectromorphological analysis notation symbols are placed over a fixed frequency grid. With the students' written consent, I let the course module form a case study where I could try out the functionality of the notation. Due to the course's overall focus on sonology and basic studio technology I decided to limit the study to electroacoustic music work, leaving the inclusion of acoustic instrument performance for the next stage of my research. What I hoped to learn from the study was:

- If there could be agreement in interpretation of the symbols
- A sense of the notation system's intelligibility—if there were aspects of the notation system that were particularly hard to grasp
- Whether problems occur when analysis notation is placed over a fixed frequency grid
- Whether problems occur when new symbols are combined with traditional notation
- A sense of the artistic relevance of working with this system of representation for composition

4.2 Participants

Seven students (4F, 3M; average age 27.7, SD = 6.9) attending the course agreed to participate in the study. They were all Swedish citizens. All participants were composition students at the bachelor level, familiar with traditional music notation, while none of them had worked with aural sonology notation before.

4.3 Method

The process for the case study was as follows:

1. Construct a notation system prototype that would qualify to meet the demands of the course module in aural sonology [7] while introducing the concepts mentioned above, with the exception of spatialization
2. Have composition students realize a given score using this notation also reflecting over their experience
3. Make initial improvements to the notation system based on initial feedback as well as my own teaching experience
4. Have composition students create and realize a short score of their own, using this notation, also providing a written reflection of their experience
5. Evaluate the study

The students' reflections from the first notation assignment were given verbally from the time of the assignment to its presentation and any new input with regard to the functionality of the notation was noted and eventually documented. The final assignment had a required written reflection. This division was practical since the students' initial questions with regard to the notation emerged gradually as they grew more familiar with the system.

There were no particular restrictions regarding the tools and/or technology used for the course module assignments. We listened to, and discussed the assignments together in a studio for electroacoustic music. For their own notation, I provided a pdf with empty staff-systems with clefs and a frequency scale on the left side. I instructed the students that they could print the pdf and write by hand, use it as background in graphics software, or construct an equivalent staff-systems in a notation software of their choice. Regardless of these choices they were required to hand in the finished scores in digital form.

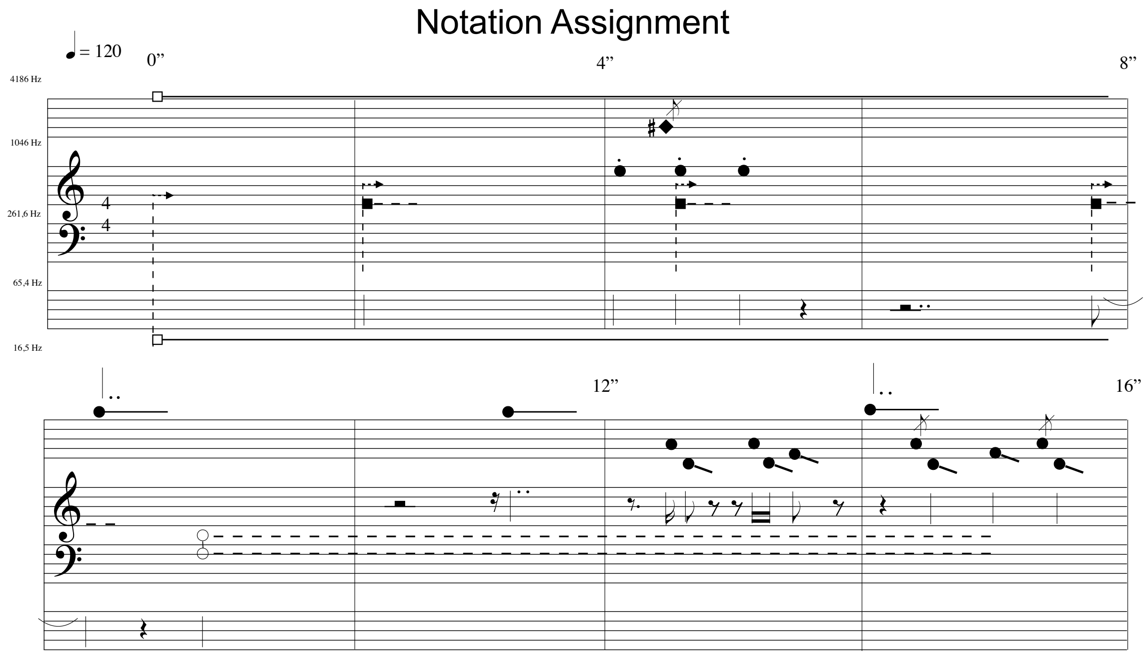


Figure 4. The score of the first notation assignment, which all students individually were asked to realize, using concrete sounds.

4.4 First Notation Assignment and Feedback

Because of the underlying idea of exploring artistic relevance, the first notation assignment was based on a short electroacoustic composition I made with a combination of electronic and concrete sound material, starting with buzzing electronic noises and ending with the characteristic sounds of seagulls (see Figure 4—the audio file can be accessed at *https://kmhsweden.box.com/v/tenor2018*). I created a score from this short composition and presented it to the students without playing the original composition. Incidentally, the original composition contained additional sounding details that I chose not to include in the score so as not to unnecessarily complicate the assignment. For the notation of time, I used a combination of traditional symbols in tempo 60, and time indicators in seconds, placed over the score. While note stems indicate durations, extension lines were necessary to show modes of energy articulation. Each student was given the task of realizing the score, using only concrete sounds—this was important in order to avoid the assignments becoming archetypal translations of the notation symbols i.e. using noise generators for complex sounds and pure oscillators for pitched sounds. I wanted artistic interpretations, not simulations of the score. Also, in not allowing synthesis, they had to listen to the sounds around them and reflect on their possible connections to the notation symbols at hand. However, I allowed transposition and filtering of the concrete sounds—it would otherwise have been difficult to meet the demands of the sound objects' positions in the frequency space. Upon hearing the sounding results of the students' assignments I got the sense of hearing different interpretation of the same piece. Because of the freedom in selecting sounds and the lack of indication of dynamics, the interpretations were quite different, but the shared structure with regard to sound objects and their spectral contents and placements in time, made for a relatively coherent collection of pieces. The questions and/or problems that arose as a result of the first notation assignment, can be divided into four categories:

1. The concept of classifying sounds and their energy articulation
2. New symbols combined with traditional notation
3. Conflicting information within the new notation system
4. Musical features missing in the notation system

The first category was expected and is something I face every year when introducing the concept of aural sonology to composers not yet familiar with this way of categorizing and describing sound. Particularly the concept of energy articulation and *facture*—the combined experience of energy over time—can be hard to grasp for someone used to traditional notation. But even basic understanding of what complex, non-pitch-based sounds are and how they behave can't be taken for granted. That being said, much of the confusion expressed with regard to understanding the score (Figure 4) could be related to the second category, that traditional music notation was combined with new notation. While the traditional notation of rhythm was helpful in describing the rhythmic sounds of seagulls, its placement inside the staff system made it at times confusing, particularly quarter notes whose stems could be mistaken for a new sonology notation feature. Another issue was how to make sense of non-pitched sounds placed over a traditional staff system. The most frequently misunderstood sound object was the interval of two pitched notes with iterated energy articulation starting in bar five and continuing to the end. This confusion emanated from the third category—conflicting information in the notation system. In the notation compendium that I distributed with the assignment there were summaries of Thoresen's and Hedman's sound categories which were not compatible with my instructions for how to combine symbols on the musical grid. The fourth and last category concerns elements missing in the score and these questions were raised particularly with regard to dynamics. Naturally the lack of dynamic indicators in the assignment's notation led to the greatest variations overall in the interpretations of the score. During the presentation of the assignments in class we discussed different solutions for this.

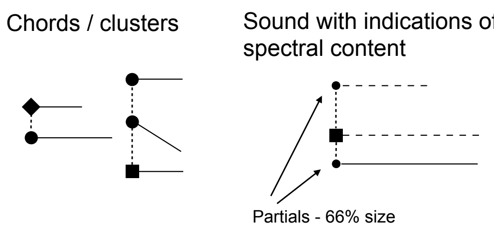


Figure 5. Example of one modification to the notation system following the discussion after the first assignment—using smaller note heads to indicate when a sound component functions as a partial in another sound's spectrum.

4.5 Modifications

Some of the problems mentioned needed to, and could be, addressed immediately. Therefore, I made some clarifications and modifications to the notation for the second assignment:

- Use small note heads (66% of original size) to specify when symbols are partials to a main sound rather than equal chord/cluster components (see Figure 5).
- Use the same dashed vertical line for indicating spectral width and for connecting partials/chord notes
- When combining indicators of spectral width and traditional notation of time, place rhythmic information on separate single staff lines to avoid confusion.
- A more detailed frequency scale is placed next to the clefs to help with placing non-pitched sounds and their spectral width on the staff-system—this was practical since most students relied on software spectral analyzers to discover the bandwidths of their non-pitched sounds.

Figure 6 shows what the first four bars of the first notation assignment would look like with these changes in place.

4.6 Second Assignment and Feedback

For the second assignment the students were asked to create and realize a short score of their own with a total duration of 30 seconds, again using concrete sounds. The score had to include at least seven unique sound objects. The assignment also included providing the individual sound objects as separate sound files, and a written reflection detailing their process. I provided a pdf with empty staff-systems with frequency indications for the students to use with their computers or for writing by hand. There were less questions and problems addressed following the second assignment. Already as the assignment was given, students expressed how having control of the notation in this assignment rather than working with a predefined score, made their task easier. The problems that were addressed by the students following the presentations of the second notation assignment can be summarized as follows:

1. Introducing spectral information and non-pitch based sound to traditional staff-systems takes time to get used to
2. Few chose to write their notation before creating the music—it was easier to think of the system in terms of analysis
3. Notation of dynamics continued to be an important issue
4. It was hard to make room for all symbols on one four-staff system

The first category was expected. The second doesn't surprise me either—it was easier to make the music first and then notate the music. This way of composing, starting with the sounds themselves is common for works of musique concrète, being discussed by Schaeffer at an early stage of the genre's development [8]. Some provided new ideas for the notation of dynamics, varying the sizes of sound symbols or their extension lines. Many expressed difficulties in getting all the notation symbols into one single four-staff system, as if this had been a requirement for the assignment. This is understandable considering that the score I produced for their previous assignment had all symbols on one system. Indeed, for analysis and composition overview purposes having all sound objects sharing a system is convenient, in the same way as a piano reduction is practical to get an overview of an orchestral work. But I realized that in shaping the previous assignment in this way, I had myself used a descriptive approach to the notation symbols despite the assignment's explicit goal to explore the notation's prescriptive potential. It would have made sense to give them a score with the musical layers divided between different staff-groups as shown in the modified example of the first notation assignment (see Figure 6).

4.7 Case Study Conclusions

Comparing this study to previous runs of the sonology course module, the fixed frequency grid made the task more difficult for the students because they had to learn how to analyze sounds and extract their spectral information, though this is highly accessible these days thanks to analyzers and sonogram possibilities in open source software. On the other hand, by grounding their work in real measurable sound, their work was easier to assess—the

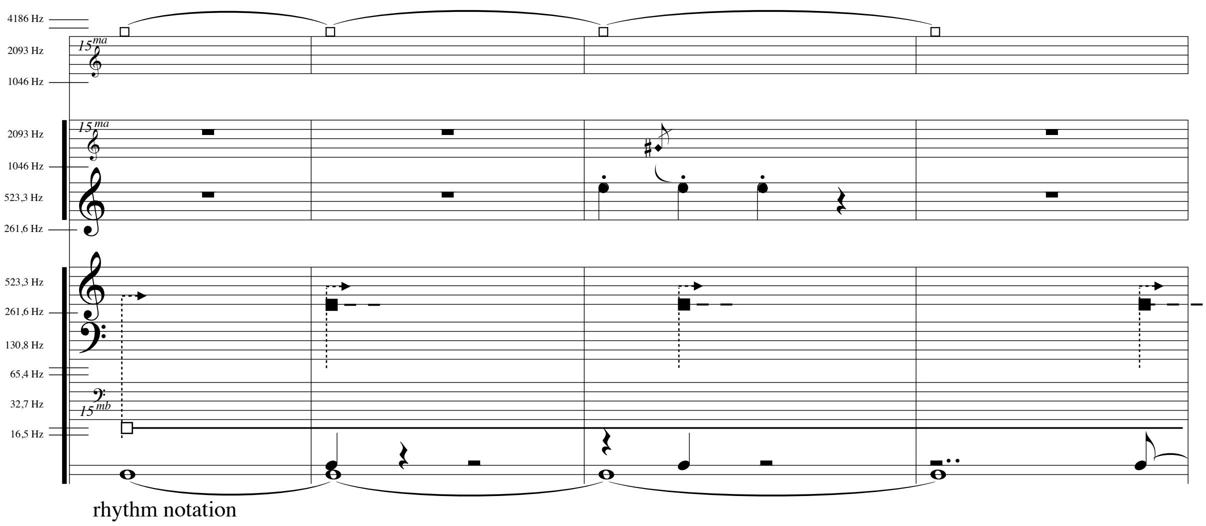



Figure 6. The first four bars of the first notation assignment with modifications following the feedback from the students.

symbols they put on their music staves had a real and measurable counterpart in sound. While proving a difficult task, working with combinations of complex, pitched and dystonic sounds as described here was not impossible for any of the students participating in the study. They all provided good work both in terms of artistic output and notational accuracy. I also found that by having them take part in the assessment of the system itself there was a sense of ownership of the notational language that made some students very active in discussing the functionality and possibilities of the system. Since this course module will run again, I will gradually gather more data with regard to the possibilities and challenges surrounding the usefulness of this system for composing and teaching. While my, perhaps utopian, research objectives raise fundamental questions regarding the nature of music notation, I hope that the pragmatic method described here will begin to show both the possibilities and the limitations of my approach to achieving these goals.

5. FUTURE WORK

When I will adapt the system for our electroacoustic music ear training course module, I will introduce a further developed notation for representing space. Here I will look with interest at the development of the Spatialization Symbolic Music Notation at ICST [9]. This notation system for spatialization already addresses another area I'm aiming towards in the near future, algorithmic composition, by introducing the possibility of moving between written symbols and data.

Also planned is a composition of my own for violin and electronics using the notation system for algorithmic composition, which will require transferring the graphical symbols into the digital realm. I will also work with a collaborative project where live-electronic instruments are explored and mapped using this notation in order to make them available for new ways of composing. Provided that these various tests prove to be fruitful, I imagine that a holistic system of representation as described here, that builds on acoustic music composers' and electroacoustic music composers' prior knowledge in their respective fields could be useful both for composition and teaching, doing its part to bridge the gap between the two sound worlds.

6. REFERENCES

[1] E. Varèse and N. Slonimsky, *Ionisation: for percussion ensemble of 13 players*. Milan: Ricordi, 2000.

[2] H. Lachenmann, *Pression for one cellist*. Wiesbaden: Breitkopf & Härtel, 1972.

[3] D. Smalley, "Spectromorphology: Explaining Sound-Shapes," *Organised Sound*, vol. 2, no. 2, pp. 107–126, 1997.

[4] M. Chion, *Guide To Sound Objects*. INA / Éditions Buchet/Chastel, 1983, trans. J. Dack and C. North.

[5] L. Thoresen and A. Hedman, "Spectromorphological analysis of sound objects: an adaptation of Pierre Schaeffer's typomorphology," *Organised Sound*, vol. 12, no. 2, pp. 129–141, 2007.

[6] P. Couprie, "Algorithmique et technologies numériques dans la notation musicale," in *Musiques Orales, leur Notation et Encodage Numérique (MEI)*. Les éditions de l'immatériel, 2015, pp. 99–115.

[7] "Sonology and Studio Technology Course Plan (in Swedish)," 2012. [Online]. Available: https://www.kmh.se/download/18.51f5bb6415ecf8601e7edbb8/1507887639426/DG1067.pdf

[8] P. Schaeffer, *In Search of a Concrete Music*. University of California Press, 2012, trans. C. North, J. Dack.

[9] E. Ellberger, G. Perez, J. Schuett, G. Zoia, and L. Cavaliero, "Spatialization Symbolic Music Notation at ICST," in *Proceedings of the Joint International Computer Music Conference / Sound and Music Computing conference – ICMC-SMC*, Athens, Greece, 2014.

METHODS AND TOOLS FOR TRANSCRIBING ELECTROACOUSTIC MUSIC

Pierre Couprie
IReMus, Sorbonne University
pierre.couprie@sorbonne-universite.fr

ABSTRACT

This article presents some tools and methods to carry out transcriptions of electroacoustic music. It introduces the relationship between sound analysis and image or drawing at the birth of electroacoustic music and explains the interest of creating transcriptions. The article contains a proposed framework, based on several years of practice, which links musical analysis to transcription, sound analysis and representation. The different parts of a transcription are then detailed and methods are proposed to create annotations with reference to various examples I have created since the late 1990s.

The last section presents the EAnalysis package, developed with Simon Emmerson and Leigh Landy at Leicester's De Montfort University, in order to create a tool for analyzing, transcribing and representing electroacoustic music. It introduces the interface, the architecture and the transcription features of this piece of software in relation to other technologies.

1. INTRODUCTION

1.1 Schaeffer and the transcription of acousmatic music

In his *Treatise on Musical Objects*, Pierre Schaeffer chose the term "acousmatic" in order to characterize listening which does not include the search for production and transmission practices. He placed listening at the heart of the studied phenomenon. This acoustic listening "symbolically forbids any relationship with the visible, touchable, measurable" [1]. On the other hand, the acousmatic listener can dissect the sound by isolating it, varying its playback speed, intensity, or repeating it. This is the first instrumented listening in electroacoustic music. Analog playback and editing technologies enabled Schaeffer and members of the Groupe de Recherches Musicales (GRM) to analyze sounds through the concept of sound objects.

Studying music created without images through transcription or visualization can seem paradoxical. Indeed, Vincent Tiffon has written that: "The sonogram allows the visual transfer of sounds and music that are precisely designed outside this visual context. At the heart of this paradox, the contradiction resides between acousmatic music and an anti-acousmatic analysis method. Listening to the spectrum changes the pure acousmatic character of this music" [2]. However, in 1952, Schaeffer imagined the possibility of working on the relationship between image and sound through abstract painting: "Some concrete music works immediately demand the graphic translation and it would not be impossible, for example, to compose a concrete music by expressing the equivalence of matter and form of an abstract painting. In any case, this painting would be a better score than notes on music staff paper. Thus, there is an indisputable link between these two new phenomena, which establish a solid bridge between painting and music" [3].

Gaël Tissot has explained the complex relationship between music and visual arts within the GRM [4]. He argues that there has often been a convergence between the morphological work of the composer and the notion of plasticity stemming from the visual field, while going beyond the scope of the Groupe Recherche Image (GRI). Jacques Vidal and François Delalande's animated transcription [5] of the fourth movement of Schaeffer's *Études aux objets* (1959) demonstrates that the idea of using visualization for an instrumented listening to concrete music dates back to at least the 1970s. As for the instrumented listening activity itself, it probably began with the history of the recording [6]. The shift between acousmatic listening and instrumented listening has been accentuated with the advent of digital technologies and their graphical interfaces, since the manipulation of sound can only be controlled through a representation of it.

Acousmatic music and sound visualization are complementary. They are listening practices in which a form of visualization improves the understanding of the phenomenon being listened to. With musical analysis, visualization is even an essential prerequisite, as acousmatic listening is used several times during the analytical process, benefiting from all the possibilities of instrumented listening.

1.2 Why would we trancribe?

We have previously mentioned the lack of visual support as a hindrance to the development of electroacoustic music analysis. The ethnomusicologist Simha Arom points out that the study of traditional oral music requires one "to have a global picture of the sound document in front of us at all times" [7]. The proximity between the analytical approach used in ethnomusicology and the one used in the electroacoustic works has allowed for the develop-

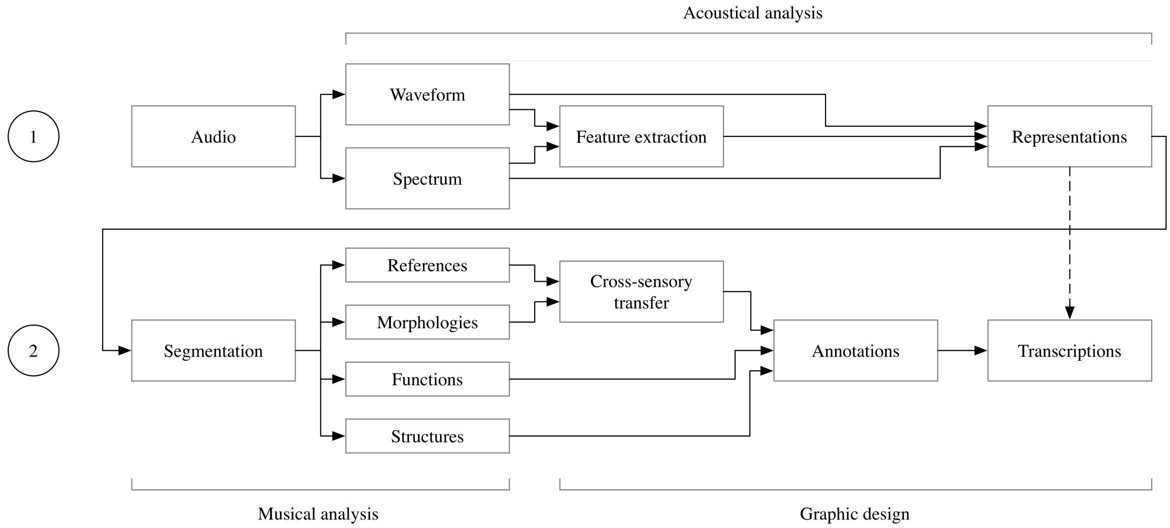


Figure 1. Representation and transcription framework.

ment of transcription in the same way as that used in structural linguistics. This filiation can be found in the work of several researchers such as François Delalande, who has used transcription as a pre-analytical step [8], or in that of François-Bernard Mâche, who has linked the units to their context in a phonological process [9]. Transcribing an electroacoustic work partially follows the steps developed for the structural study of languages. Partially, because some steps such as switching, or the concept of equivalence class are only rarely applicable in a systematic way. The segmentation of the musical flow remains one of the most problematic steps in the analysis of electroacoustic works. Thus presented, transcription remains a primarily descriptive tool.

In addition to transcription as a pre-analytical step, its use in a pedagogical or presentation context remains the most widely used. I have developed a graphical code that is attractive and easily understandable by a novice audience, for instance the transcriptions which I carried out for the CD-ROM *La musique électroacoustique* [10]. Colors, shapes and their arrangement on the graphical space have been chosen to enhance the understanding and memorization of works. This coding generally corresponds to the origin or context in which the sounds are used. The origin of the sounds is often imaginary – in this way, the approach is close to the concept of sound-image proposed by François Bayle [11]. On the contrary, transcription, which is often absent from theoretical writings on electroacoustic music, offers a wealth of possibilities. Finally, transcription is very commonly used to exemplify analytical discourse.

Whether it is as a pre-analytical step, an educational tool or as an example accompanying a talk, transcription is generally an essential step. This analytical step can also become the first act of creation, for example by using the structure or elements of an existing work in the creation of a new work.

2. THE ANALYTICAL FRAMEWORK

Transcription is usually a complex exercise (Figure 1). The method proposed in this article is divided into two steps.

2.1 Representations

This section concerns the realization of one or more representations which serve as visual support to musical analysis. The waveform and spectrum (linear, logarithmic or wavelets) facilitate the segmentation of the musical flow into sound materials. If the analytical objective is to study slow evolution or to opt for an analysis method without segmentation, then it is interesting to extract audio descriptors. One or more representations are realized during this step.

2.2 Transcriptions

During transcription creation, the type of segmentation is selected from musical analysis:

1. By identifying the origin of sounds (causal analysis).
2. By segmenting the musical flow into morphologies based on an analysis of the acoustic parameters of sounds (morphological analysis).
3. By identifying the musical functions of musical discourse (functional analysis).
4. By identifying elements of structures ranging from large temporal divisions – musical form – to the most finely divided – microstructures (formal analysis).

The analyst can choose between these types of segmentation or mix them. The next step is to convert these analytical elements to annotations that will be assembled to create the transcription. Then, transcriptions and representations can be combined to form analytical or composite representations [1] .

[1] Analytical or composite representations combine several transcriptions and/or representations to create a complex visualization of analysis.

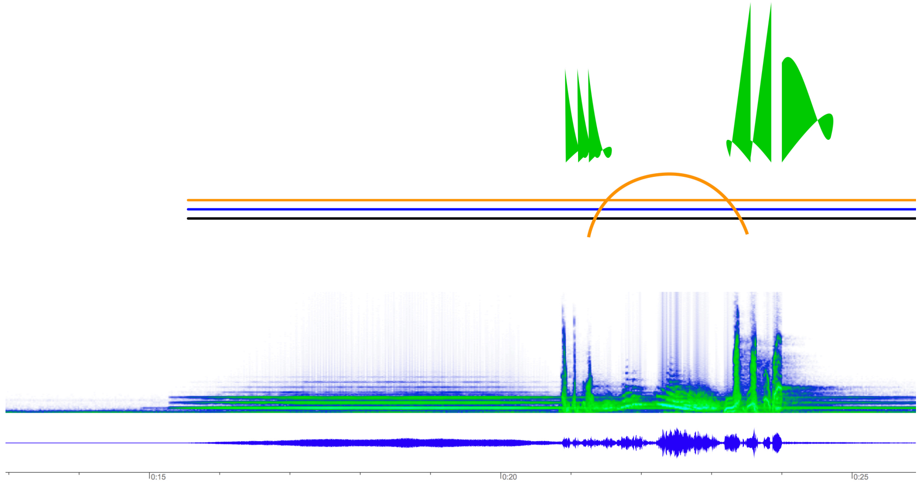


Figure 2. Extract of transcription of “L’oiseau moqueur” (*Trois rêves d’oiseau*) by François Bayle.

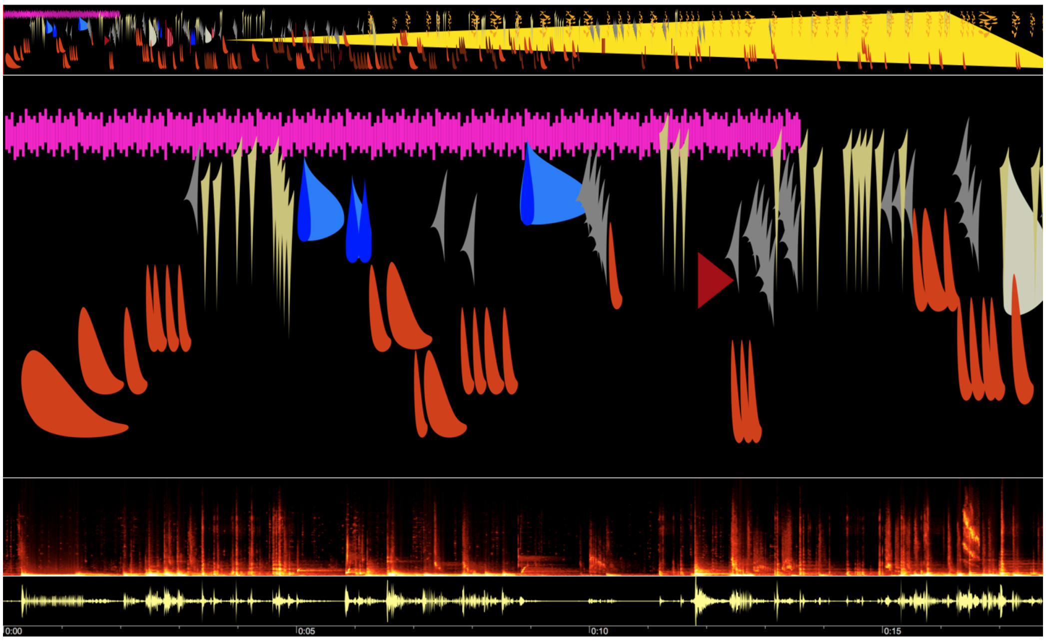


Figure 3. Extract of transcription of “Ondes croisées” (*De Natura Sonorum*) by Bernard Parmegiani.

2.3 The transcription space

The transcription space is the frame that will contain one or more backgrounds and graphic annotations. This space contains several dimensions organized into four categories : the graphical plane, background, annotations and other analytical parameters [2] .

2.3.1 The graphical plane axis

The horizontal axis always represents time and the vertical axis can represent approximate pitches. In each of the three short pieces of *Trois rêves d’oiseau*, the composer François Bayle uses harmonic sounds from musical instruments or from natural sounds such as bird songs. While the real pitch of these sounds has very little importance in the composer’s language, their representation in the form of a single line positioned in a specific register (Figure 2) makes it easy to distinguish them from other sounds [12].

It is also possible to represent spectral heights on the vertical axis. In my analysis of “Ondes croisées”, a movement of *De Natura Sonorum* by Bernard Parmegiani [13], I highlighted, first, the link between spectral heights and categories of sound and, second, the spectral evolution of the material from the bass to the treble (Figure 3) by drawing the spectral thickness and the approximate position of the sound on the frequency scale.

The transcription of other parameters such as the stereophonic position of the sounds highlights the importance of one of the dimensions of the space as an element of the musical form. Alain Savouret’s extract from *Don Quixote Corporation* has the particularity of being built on sounds that are easy to segment with a theme and variations musical structure. Moreover, each sound is positioned on the

[2] I briefly present the first two categories in this section. The creation of annotations is presented in section 2.4 and I do not mention other analytical parameters which are beyond the field of this article.

panoramic in an easily perceptible way, and there is no doubt that the composer used this criterion as one of the variation parameters.

2.3.2 Background

When I realized this transcription [10], I positioned the annotation shapes vertically in relation to the panoramic space (Figure 4). The transcription highlights gestures or articulation between this space parameter and the evolution of the musical structure.

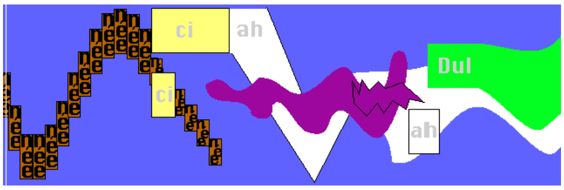


Figure 4. Extract of transcription of *Don Quixote Corporation* by Alain Savouret.

The transcription presented in Figure 4 only includes a colored background. However, it is also possible to add a waveform or a spectrum. In this case, the representation contains several other dimensions:

1. In the case of a waveform, the vertical axis represents the amplitude of the sound-signal mirrored waveform with a linear scale.

2. In the case of a spectrum, the vertical axis represents the frequencies, and the colors represent the intensities in grayscale or with pseudo-color (or false color).

It is pretty rare to use the background for legibility reasons, but it is quite common to juxtapose a waveform or a spectrum to a transcription. In Jean-Claude Risset's *Sud* transcription [14] (Figure 5), I used this technique to ensure that representations complete the transcription.

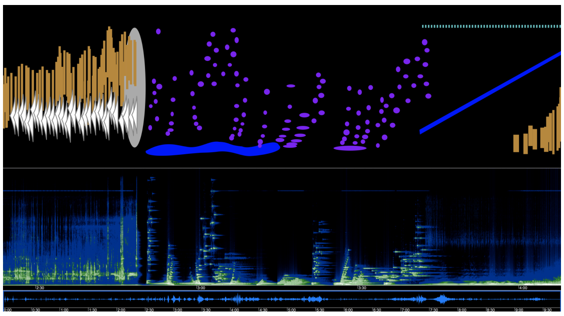


Figure 5. Extract of transcription of *Sud* (part 1) by Jean-Claude Risset.

2.4 Annotations

2.4.1 Semiotic correspondences

Creating a transcription is often complex. The choice of graphics and the link between analysis criteria and graphical properties alone leads to a concentration of many problems. The graphical characteristics of a transcription [15] always sit along an axis which ranges between iconic and symbolic representation. The terms "iconic" and "symbolic" are used here in their semiotic sense. Following Charles S. Peirce's example [16], an icon (shape) refers to an object (the segmented sound unit) by its resemblance relationship, while the relationship between a symbol (shape) and an object is based on a social convention. A drawing of waves to transcribe the sound of the sea, or what looks like an aquatic sound, as I employed in Jean-Claude Risset's representation of *Sud* [14], is relevant to the concept of icon. Greimas and Courtés propose broadening the term of iconicity to define it "as the result of a set of procedures to produce the effect of meaning 'reality'" [17]. Thus, an icon is a "referential illusion" and this is how we analyze these graphic annotations, which might, for example, take their form from the evolution of the intensity of the sound units. Cécile Régnault takes advantage of these referential illusions to describe sensory or factual correspondences [18] between, for instance the granular qualities of visual and sound textures, or the length of a graph and the duration of a sound. Correspondences based on gestural analogies are very efficient in a pedagogical situation or for the performance of an electronic representation of the electronic part of a mixed work.

Alain Savouret's transcription of *Don Quixote Corporation* is largely symbolic. Some of the segmented units are represented as rectangles, the color of which denotes the effects of sound transformation and manipulations. Lasse Thoresen developed a graphical transcription system based on Schaefferian typomorphology [19, 20]. The same remark can be made for the symbols used by Roy in his system for transcribing musical functions [21].

However, there are few totally iconic or symbolic transcriptions, as analysts generally use a medium-term approach and do not hesitate to move along the axis which links iconic to symbolic during the same transcription. The predominance of iconicity makes graphics easier to understand by using analogies between visual representations and sound. On the other hand, the symbolic character offers a wider range of transcription possibilities by allowing different types of sound parameters to be superimposed on the same graphical shape. Learning the meaning of symbols is then often essential, but the analyst can also rely on cultural conventions which can be comprehended by everyone, such as the height of sounds and the vertical position of a graphical shape, the spectral richness or the thickness of the line. Therefore, the choice of graphic shapes depends on the purpose of the transcription and the target audience. When I carried out the transcriptions for the CD-ROM *La musique électroacoustique*, I opted for iconic graphics or for a simple symbolization to allow their use in a pedagogical context while avoiding the pitfall of redundancy.

2.4.2 Links between sound and visual

Drawing on Bertin's work, Cécile Régnault has proposed a table of the correspondences [18] between the visual parameters of annotations and Schaefferian sound object criteria. This table lists a set of common uses which began to spread in the early 2000s. I have experimented with these different correspondences in my representations:

1. Graphical shape: Causality, gait, background-figure, typology.
2. Graphical thickness: Intensity, spectral width.
3. Vertical position: Real or approximate pitch, panoramic, spectral structure, formal structure (formal diagram).
4. Color: Typology, effects, sound layers.
5. Texture: Granulosity.
6. Animation: Space motions.

Most of these correspondences come from inter-sensory transfers or cultural habits related to musical notation: for example, the position or the vertical thickness used to transcribe height or spectral width.

However, I also experimented with transcription at the borderline of graphic art to assess the point when foreground and background exchange their roles. Usually, the background is used as a space which represents the space of time and frequency. The shapes drawn on this background stand out and are represented by units segmented during the analysis. The transcription I have made of *Hétérozygote* and Luc Ferrari's series of *Presque rien* [22] tends to attenuate the separation between form (foreground) and background. The background becomes a part of the forms, or guides a non-linear temporal navigation and the forms no longer stand out in the background. The transcription, usually done on at least two planes, is rendered in a single plane. These transcriptions also provided an opportunity to experience the minimum elements to be included in a representation. Luc Ferrari's music, based on anecdotal and minimalist soundscapes, is perfect for this kind of experimentation. The three transcriptions made of *Presque rien n^o 1, le lever du jour au bord de la mer* contain only one or two graphical shapes whose morphology corresponds to the sound amplitude of the foreground. The background, broken down into several parts, represents the background sounds. In order to simplify reading, the timeline is represented by the horizontal axis. In the third extract (Figure 6), the background and the foreground tend to merge, and it is difficult to say whether there are one, two, three or four forms in the background. The uniform color removes the sound space captured and created by the composer, guiding listening towards the main form and giving an immediate and synoptic vision of the whole.

2.4.3 Synoptic transcriptions

In my analysis of Bernard Parmegiani's "Ondes croisées" for the CD ROM *La musique électroacoustique*, my transcription was published in two forms: the paginated transcription and the enlarged synoptic transcription. These two versions of the same transcription were placed under each other (Figure 3). I had not anticipated the importance of this layout to explain a musical form based on a crossfade. This short piece is built from two types of sound

Figure 6. Transcription of an extract of *Presque rien n° 1, le lever du jour au bord de la mer* by Luc Ferrari.

materials: held flicker sounds and short sounds organized in a complex texture (a mixture of pizzicato, double bass, elastic, water drop and zarb musical instruments). The scintillating flicker of the beginning turns is the end of the previous part ("Matières induites") and has no role in the construction of the form of this piece. It is therefore based on the progressive appearance of a white noise (from a fire recording) at 0:25, which increasingly attenuates the complex texture of short sounds. The result is a fade that lasts about 1:30. I decided to represent the white noise by a large yellow triangle in which all the other sounds were gradually dissolving. This transcription, presented in a synoptic way, only retained this effect of fading graphically, revealing its musical equivalent. From my point of view, this form is one of the most striking gestures of *De Natura Sonorum*. The sound of white noise is thus at the origin of the musical form of this piece. However, this was not considered to be very important by the authors of *L'envers d'une œuvre* [23]. They considered it as a regression compared to the previous parts. However, the composer insists on presenting this sound as a desire for formal coherence.

The synoptic transcription is ideal for reporting formal processes. In my analysis of François Bayle's *La fleur future* [24], I have made formal transcriptions with linear and formal diagrams. They reveal the evolution of typologies of timbre, the amplitude of envelopes, and the role of silences in this short piece.

3. SOFTWARE

3.1 Available technologies

There are four types of software to carry out representations and transcriptions of electroacoustic music [25]:

1. Software to manipulate the sound spectrum: Audiosculpt [3], SPEAR [4]. The modification of the gain (enforcement or filtering) of regions of the spectrum facilitates the analysis of complex textures or

[3] Audiosculpt is based on SuperVP technology to analyze and manipulate temporal and spectral properties of sounds. It is distributed through the Ircam forum (`http://forumnet.ircam.fr`).

[4] SPEAR is a piece of free software developed by Michael Klingbeil (`http://www.klingbeil.com/spear`).

the study of mixing in which masking effects are used by the composer to orchestrate his material.

2. Sound information retrieval tools: Audiosculpt, Vamp plug-ins[5] inside Sonic Visualiser[6]. For several years, musicologists have used audio descriptors to analyze music.

3. Annotation tools: Sonic Visualiser, ASAnnotation[7], Metascore[8], Acousmographe[9]. From simple temporal annotation to morphological transcriptions, this software is essential for the analysis of electroacoustic music.

4. Musical analysis software: an Aural Sonology plug-in[10] can be used with the Acousmographe, Acousmoscribe[11], and TIAALS[12]. These technologies take a step forward in computer-assisted analysis by offering specific functions designed for musicology.

Unfortunately, these software packages have limitations:

1. They cannot analyze audiovisual files, they only use sound files, and most of these are only stereophonic files. Video music and multitrack works are very common in electroacoustic music. Moreover, video is a good support to analyze performance.

2. Several of these packages cannot export their data to readable files or import data from other software. There is no format to exchange analyzed data between them but nevertheless, analyzing electroacoustic music requires the use of several software applications from the extraction of data to creating representations.

3. The interface is often limited and not adapted to musical studies. For example, there is no possibility to navigate inside a file and to compare different moments of a work or of different works.

4. While they have interesting features (such as the Timbre Scope of Acousmographe or the drawing of audio descriptor values on the sonogram with Sonic Visualiser), most of them are difficult to use in some contexts (e.g. with a long work, without the possibility to filter data, or to synchronize with a graphic representation).

3.2 An example: EAnalysis

In October 2010, Simon Emmerson, Leigh Landy, Mike Gatt, and myself began the New Multimedia Tools for Electroacoustic Music Analysis research project at the Music, Technology and Innovation Research Centre at Leicester's De Montfort University. As part of this project, I developed the EAnalysis[13] software to gather the essential tools for the analysis, representation and transcription of non-written music. EAnalysis is a workspace where the user can create representations, import[14] data from other software or recorded during performance, analyse them, and export in different formats[15].

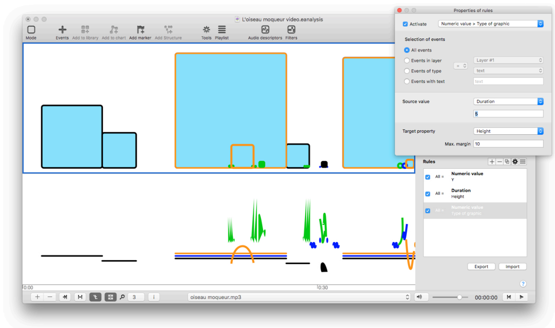


Figure 7. EAnalysis: Style sheet to create different transcriptions from the same analysis.

3.2.1 From ideas to software

The first idea of EAnalysis is to disconnect the graphical rendering from the musical or sound analysis. This feature uses a simple style sheet. One of the main difficulties when making a graphical transcription is to be able to experiment in different directions. Unfortunately, no existing software allows you to quickly modify several graphic parameters. The analyst realizes his transcription by drawing on a view and the graphical parameters are fixed once and for all. To transform this transcription, he must modify graphical shapes one by one. EAnalysis contains an additional layer: each graphic parameter can be associated with an explicit (intensity, grain, space, etc.) or neutral (keyword, text, number, etc.) analytic parameter (Figure 7). The correspondence between the graphic parameters and these analytic parameters is recorded locally on each view. A style sheet is used to create new rules for linking analytical and graphical parameters. Thus, it is easier to change a graphical transcription without affecting the parameters of the graphical form. This system also allows the user to

[5] Vamp is a plugin format dedicated to sound analysis (`http://www.vamp-plugins.org`).

[6] Sonic Visualiser has been developed by the Centre for Digital Music at Queen Mary, University of London (`http://www.sonicvisualiser.org`).

[7] ASAnnotation is free software based on Audiosculpt and has been developed at Ircam (`http://recherche.ircam.fr/anasyn/ASAnnotation`). Unfortunately, its development has been halted for several years and its compatibility with recent systems is not assured.

[8] MetaScore has been developed by Olivier Koechlin for the multimedia library of the Cité de la Musique (Paris). This software combines text, images, audio-visual files and animation to realize listening guides.

[9] The Acousmographe has been developed by INA-GRM (`http://www.inagrm.com/accueil/outils/acousmographe`) since the 1990s.

[10] The Aural Sonology plug-in was developed from Lasse Thoresen's research by the INA-GRM (`http://www.inagrm.com/aural-sonology-plugin-0`). It contains basic shapes to transcribe parameters of sounds based on an augmented schaefferian typomorphology.

[11] The Acousmoscribe has been developed by the Scrime (Bordeaux) from Jean-Louis Di Santo's research (`http://scrime.labri.fr`), and uses a spectral typomorphology.

[12] TIAALS is developed by the universities of Huddersfield and Durham (`http://www.hud.ac.uk/research/researchcentres/tacem/`). This software is still in its beta version and allows typological or paradigmatic charts to be created from extracts of the spectrum.

[13] EAnalysis is free software (`http://eanalysis.pierrecouprie.fr`).

[14] EAnalysis supports importation from audio, video, CSV, Pro Tools information session, and XML Acousmographe

[15] EAnalysis supports exportation to image, video, and CSV

generate different types of representations (for example animation, graphical curves, and out-of-time visualization).

The second idea is to help the user in his analysis by providing an analytic events library. EAnalysis contains fifteen preformatted analytic parameters (sound objects, Spectro morphologies, language grid, space, etc.) and users can add their own parameters and group them into a list, as well as to create a library to share with other users. The interface to edit events and manage their properties is simple and flexible.

The third idea is to experiment with new forms of representation by breaking with the traditional time-frequency view. One of the problems of graphical representation is the limitation of the dimensions and consequently of the functions or musical parameters which are represented.

A typical two-dimensional graphic representation can represent only three or four parameters at the same time. It is possible to add more, but this can complicate reading of the graphical output, and limit functionality. I have already presented in a previous article why a 3D analytical representation would be an error [26] because it would confuse the readability of the analysis. EAnalysis provides a simple solution to this problem as it allows for the simultaneous use of several types of views (Figure 8).

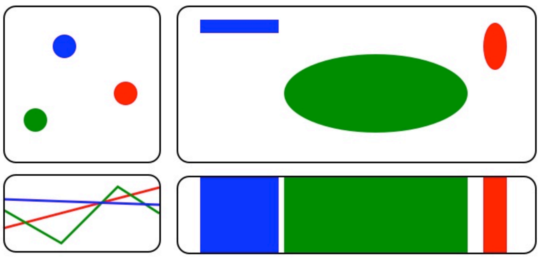


Figure 8. A representation combining several types of views (transcriptions and representations).

This multiple-view system (or composite representation) allows one to view the properties of an event from different points of view. I have already successfully experienced this type of representation in one of my previous analyses [27].

3.2.2 The architecture of EAnalysis

The architecture of EAnalysis contains three main user elements.

The **multimedia player** allows for the use of one or more audio and/or movie files inside the same project. It is at the heart of the software. This player contains all the functions useful for the analyst: loop playback, speed variation (without changing the pitch), etc.

The library of events is divided in two parts: graphical events and analytical events (Figure 9). These two types of events are different presentations of the same object. As shown in Figure 10, an event contains a set of properties divided into three categories:

1. Main properties: The name of the object and its temporal and frame coordinates.
2. Graphical properties: The type of shape and all other graphic properties.
3. Analytical properties: The list of analytical parameters.

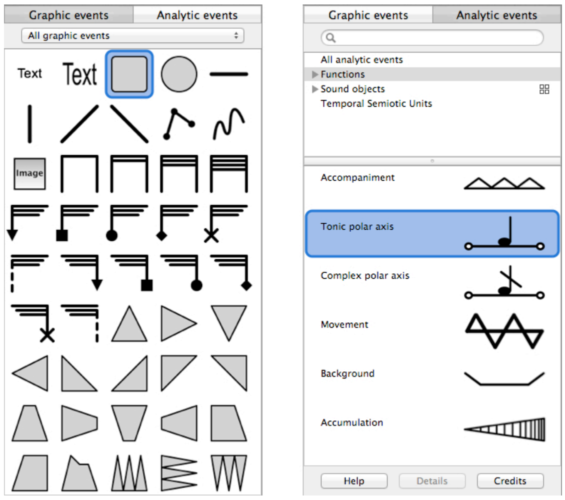


Figure 9. EAnalysis, library of events: Graphic events (left) and analytic events (right).

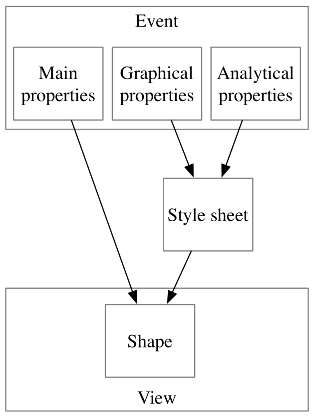


Figure 10. EAnalysis: Architecture of events (annotations).

Because events contain three types of property, they can be used for different strategies and with different levels of complexity:

1. Graphic events are very simple shapes such as are available in every drawing application: rectangle, ellipse, text, polygon, image, etc. This level is adapted to first annotations of the piece before analysis, working at listening with children, or creating beautiful graphic representations.
2. Analytic events are preformatted shapes for analysis. Each event contains a graphic shape and one or more analytic parameters. Working with preformatted analytic events is a good starting point for students to learn musical analysis or for specialists to apply existing theories.

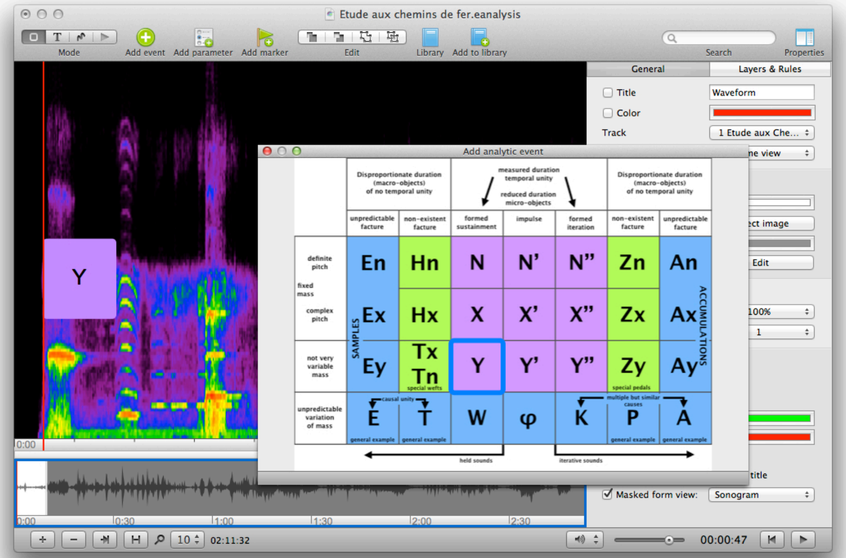


Figure 11. EAnalysis, library of events: Add a sound object from the Schaefferian typology.

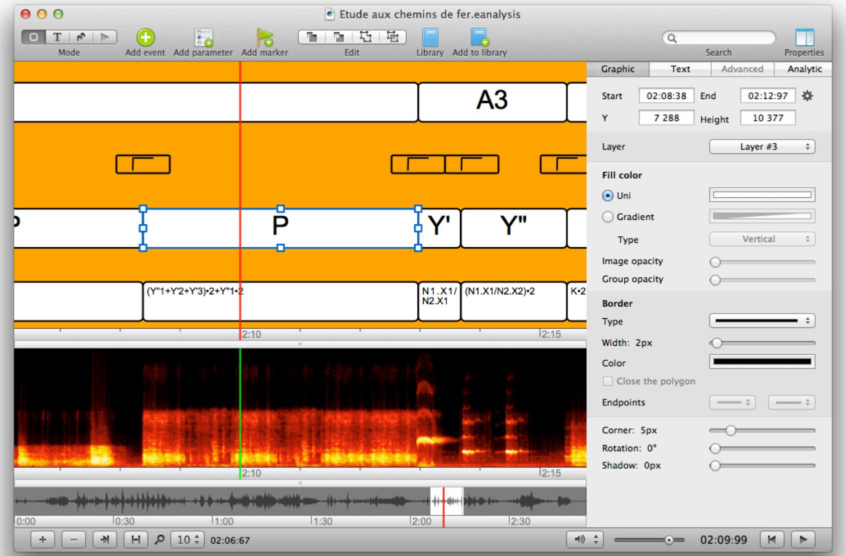


Figure 12. EAnalysis: Views to draw events or display sound representations.

3. Users can also create their own analytic events with personalized analytic parameters. This level is highly flexible, allowing the user to adapt representation and analytic segmentation to the analyzed work or to a personalized analytical theory.

A graphic event does not contain any analytical parameters but it is possible to add one, while an analytic event contains the definition of all properties related to the object. These properties can obviously be supplemented or modified by the user. They are presented through a list sorted by categories on several levels.

In practice, the navigation between the different analytic events presented through a list was not necessarily very practical or explicit. Consequently, I added a floating window to display the events in a different way (Figure 11). This window contains a clickable image from which the user can drag and drop the different events in views.

The third element is a set of **views** to display the graphical and analytical properties of events. These views are displayed in a single window. Figure 12 shows the main window with an example of three overlapping horizontal views. The user can add as many views as he wants to this window. The playback heads of the individual frames are synchronized, but the frames can be independent in their

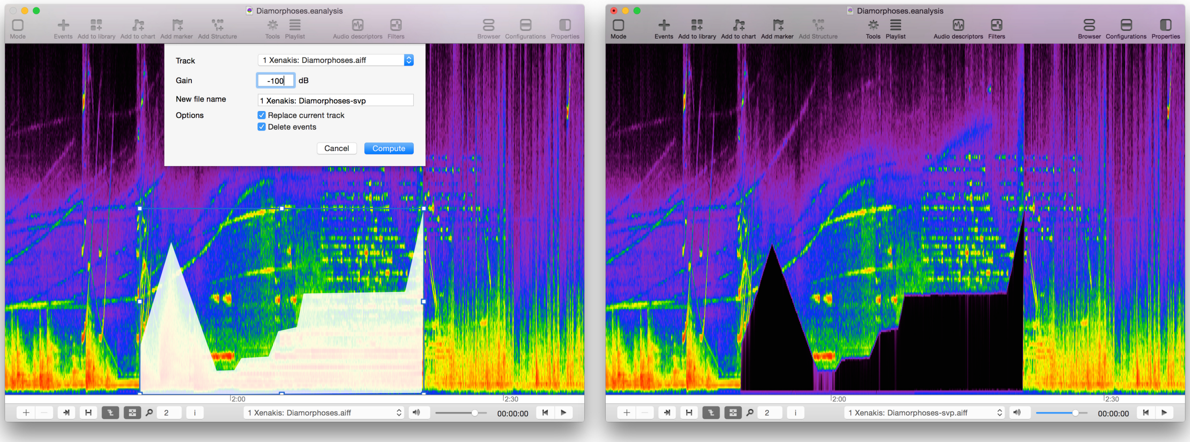


Figure 13. EAnalysis: SuperVP filter from a graphic shape.

zoom level. Thus, in this figure, the bottom view, representing the waveform, displays the entire audio file in a synoptic mode, while the two upper views are set to a zoom level to show details. Each view also has a set of parameters (background, color, playhead, timeline, mask according to another view, etc.) which can be easily modified.

3.2.3 Modes and markers

EAnalysis integrates three modes: normal, add text and drawing. These modes allow the user to create events with different tools. Normal mode is the default mode. The user adds an event by 'drag and drop' from a preformatted list or from his own library to the view. With add text mode, the user enters text during playback and can annotate audiovisual files with words or sentences. Each part of the text is an event and the user can switch to normal mode to change its graphic properties. This mode has been realized for analysts who prefer to work with text or for simple annotations of ideas during the first listening. The drawing mode is for users who prefer to draw with a mouse, graphic tablet, or interactive whiteboard. This mode is very useful to create very simple annotations on a white page, to highlight a spectrum, or to work on a whiteboard while listening with children. Moreover, if users use a graphic tablet, pressure will be detected and could be used to create artistic drawings like calligraphy.

These three modes were the first features that were developed to respond to various users' needs. They were not created as individual elements but as part of a global architecture.

Annotations (events) are also completed with markers. Markers are just time positions with simple graphic properties. They can be used to annotate ideas on first listening, or to mark breaks or structure parts. Events and markers are editable in time view, making this the default view to visualize, listen, and edit analyses. Other views are to display other data.

3.2.4 From events to filter

Since version 1.1.1, EAnalysis has been able to communicate with SuperVP to calculate gains changes in graphical annotations. SuperVP (Super Phase Vocoder) is a technology which has been developed by the Ircam Analysis/Synthesis team and is available as a command-line tool or through Audiosculpt.

Users choose any rectangle, polygon or freehand annotation to apply a gain modification (by filtering or reinforcing) and immediately display the result without exiting the software (Figure 13). This function can be used to suppress a part of the sound in order to improve the perception of the rest of the spectrum or to facilitate the perception of a low intensity spectral area.

4. CONCLUSION

This article presents the method used to create transcriptions of electroacoustic music. This method is based on several techniques drawn from acoustics, semiotics, design and musical analysis. The proposed framework relies on the practice of transcription and representation in musical analysis.

Based on a critical study of existing software, the article also presented EAnalysis software, developed since 2010. This has fixed some musicological problems encountered with other software. Moreover, EAnalysis also contains new features for musical analysis such as the use of audio descriptors or the realization of charts.

The success of EAnalysis now allows us to imagine evolution towards software which is open to the influence of all techniques used in musical analysis for the creation of transcriptions and representations. In 2018, EAnalysis will be merged with other software which I have been developing since 2006 for the creation of listening guides (iAnalysis). This fifth version of iAnalyse enhances the workflow used for transcriptions and covers the different steps of musical analysis, from the recording of a performance to the realization of listening guides through the realization of complex representations.

5. REFERENCES

[1] P. Schaeffer, *Treatise on Musical Objects: An Essay across Disciplines*. University of California Press, 2017 (1966).

[2] V. Tiffon, "La représentation sonagraphique est-elle une aide pour l'analyse perceptive de la musique électroacoustique ?" *Lien*, pp. 3–15, 2006.

[3] P. Schaeffer, *A la recherche d'une musique concrète*. Paris: Seuil, 1952.

[4] G. Tissot, "La musique électroacoustique au GRM : un art plastique ?" *Filigrane*, vol. 16, 2013. [Online]. Available: http://revues.mshparisnord.org/filigrane/index.php?id=543

[5] F. Delalande and J. Vidal, "Partition animée," INA, 1970. [Online]. Available: http://www.ina.fr/video/CPF03008561

[6] F. Delalande, "L'analyse musicale, une pratique et son histoire," in *Pratiques et objectifs des transcriptions des musiques électroacoustiques*, R. Campos and N. Donin, Eds. Geneva: Droz-HEM, 2009, pp. 131–153.

[7] S. Arom, *La boîte à outils d'un ethnomusicologue*. Montréal: Presses de l'Université de Montréal, 2007.

[8] F. Delalande, *Analyser la musique, pourquoi, comment ?* Paris: INA, 2013.

[9] F.-B. Mâche, *Musique au singulier,*. Paris: Odile Jacob, 2001.

[10] P. Couprie, "Transformation/transmutation. Analyse d'un extrait de Don Quichotte Corporation d'Alain Savouret," in *La musique électroacoustique*. Paris: INA-GRM-Hyptique, 2000.

[11] F. Bayle, *Musique acousmatique propositions... ...positions*. Paris: INA-GRM-Buchet Chastel, 1993.

[12] P. Couprie, "Three Analysis Model of L'Oiseau moqueur on of the Trois rêves d'oiseau by François Bayle," *Organised Sound*, vol. 4, no. 1, pp. 3–14, 1998. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01265026/

[13] ——, "Comment c'est fait ? Analyse graphique d'un extrait de De Natura Sonorum de Bernard Parmegiani," in *La musique électroacoustique*. Paris: INA-GRM-Hyptique, 2000. [Online]. Available: https://hal.archives-ouvertes.fr/medihal-01305248

[14] ——, "Représentation de Sud de Jean-Claude Risset," in *Portrait polychrome Jean-Claude Risset*. Paris: INA-GRM, 2001.

[15] ——, "Graphical Representation: An Analytical and Publication Tool for Electroacoustic Music," *Organised Sound*, vol. 9, no. 1, pp. 109–113, 2004. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01265027

[16] C. S. Peirce, *Écrits sur le signe*. Paris: Seuil, 1978.

[17] J. C. Algirdas Julien Greimas, *Sémiotique. Dictionnaire raisonné de la théorie du langage*. Paris: Hachette, 1993.

[18] C. Régnault, "Correspondances entre graphisme et son : les représentations visuelles de l'objet sonore," in *Du sonore au musical. Cinquante années de recherches concrètes (1948-1998)*, S. Dallet and A. Veitl, Eds. Paris: L'Harmattan, 2001, pp. 307–337.

[19] L. Thoresen, "Spectromorphological Analysis of Sound Objects: An Adaptation of Pierre Schaeffer's Typomorphology," *Organised Sound*, vol. 12, no. 2, pp. 129–141, 2007.

[20] ——, *Emergent Musical Forms: Aural Explorations*. London: University of Western Ontario, 2015.

[21] S. Roy, *L'analyse des musiques électroacoustiques : modèles et propositions*. Paris: L'Harmattan, 2003.

[22] P. Couprie and D. Teruggi, "Hétérozygote et les Presque rien," in *Portrait polychrome Luc Ferrari*. Paris: INA-GRM, 2001.

[23] P. Mion, J.-J. Nattiez, and J.-C. Thomas, *L'envers d'une œuvre. De Natura Sonorum de Bernard Parmegiani*. Paris: INA-GRM/Buchet Chastel, 1982.

[24] P. Couprie, "Quelques pistes pour analyser la structure de La fleur future de François Bayle," in *Die Klangwelt des François Bayle*, M. Erbe and C. von Blumröder, Eds. Vienne: Verlag Der Apfel, 2012, pp. 205–209. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01261202

[25] ——, "EAnalysis: Developing a Sound Based Music Analytical Tool," in *Expanding the Horizon of Electroacoustic Music Analysis*, S. Emmerson and L. Landy, Eds. Cambridge University Press, 2016, pp. 170–194. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01290982

[26] ——, "Dessin 3D et système immersif pour la représentation de la musique électroacoustique," in *Proceedings of the Electroacoustic Music Studies Network confernce (EMS'07)*, Leicester, 2007. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01258371

[27] ——, "(Re)Presenting electroacoustic music," *Organised Sound*, vol. 11, no. 2, pp. 119–124, 2006. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01265028/

AUDIOVISUAL SCORES AND PARTS SYNCHRONIZED OVER THE WEB

Jonathan Bell
Aix Marseille Univ, CNRS, PRISM
Perception, Representations, Image, Sound, Music
Marseille, France.
jonathan.BELL@univ-amu.fr

ABSTRACT

Typing **smartvox.eu** into the address bar of the browser of several phones, tablets and/or computers simultaneously is a simple way to understand what this web application is about: the different parts of the same musical score are being synchronized over the internet, through a remote server. This form of music making falls under the category of *networked music performance*, and addresses questions regarding what becomes a live performance of chamber music when musicians are distanced from each other, when sheet music is replaced by screens and headphones, or when the form of the piece is generated live by algorithms. The scores, composed in the *Bach* environment, display a scrolling playhead in proportional notation, bypassing conventional bars and beat rhythmic notation. Providing the performer with audio-scores and animated musical representation undoubtedly simplifies and speeds up the rehearsal process, but large forces (such as 80 simultaneous connections) and the constraints of a concert situation still leave numerous technical problems unsolved (mainly concerned with synchronization between devices, and server saturation), to which the present paper will attempt to partially formulate a response.

1. INTRODUCTION

This article presents the continuation of the research published in *SmartVox, a web-based distributed media player as notation tool for choral practices* [1]. SmartVox is developed within the *Soundworks* framework [2]. It is a single page web application dedicated to the delivery and synchronization of polyphonic multimedia (audio + visual) scores, themselves composed in the *Bach* environment [3] for Max/MSP. In a performance involving, for instance, thirty singers, each performer hears and sees his/her own part displayed in the browser of his/her smartphone, and the whole is synchronized through the distributed state of the web application. The application has enjoyed increasing success since its conception in 2015, which leads to two questions:

- For the performer, how does audiovisual animated notation differ from paper music?
- What is transformed from a chamber music perspective, and from an audience point of view?

Following animated notation with a cursor/timeline moving from left to right is often experienced, from a performer's perspective, as far simpler than deciphering a rhythmically complex score and following a human conductor. Dynamic notation [4] and audio-scores [5] can also facilitate the learning process of unfamiliar idioms, such as microtonality. However, the use of screens can be a potential drawback, as it prevents performers from making notes on their scores, such as breath indications, fingerings, etc... If James Bean's DENM environment attempts to solve this issue with an elaborate GUI [6], some composers judge this (extreme) sight-reading situation to be something worthwhile exploring [7][8][9]. This "slow but steady shift away from textualization in digital media" [10] only concerns the way the information is presented to the performer—not its content—, and yet the score *medium* is of such importance for the composer that it presents a genuine change of paradigm in his/her craft. With audiovisual information distributed to the players through smartphones during a performance, the interaction between audio-scores (musical material sent through earpieces to performers) and visual input (musical notation) changes the traditional relationship between composer, conductor, performer, and listener.

Networked music performances are usually thought of as way to play together from distant locations: "Networked music performance is often intended as a practice of remote performance, with musicians displaced far away at distant locations" [11], "A Network Musical Performance (NMP) occurs when a group of musicians, located at different physical locations, interact over a network to perform as they would if located in the same room" [12], or even across the globe: "Sonic and spatial aspects of networked music performance (NMP) or networked multimedia performance [...] will be explored from a particular perspective—that of remoteness or spatial distance. In NMP the performers (agents) are typically separated by distances, which in extreme cases add up to tens of thousands of kilometers." [13] With the growth of the internet at the turn of the century, the SoundWire group at Stanford CCRMA and other labs produced important research works and concerts which explored this global aspect of performance. Georg Hajdu had already achieved such a performance in 2000: "In the performance of my piece MindTrip [...], the five performers were located in different cities across the globe.

They were linked by the Quintet.net server running locally in Münster, connected via a 128 kbit dial-in ISDN connection to the internet" [10]. Networked performances, therefore, refer in no small part to the possibility to play together from distant locations. Some recent developments of NPM and distributed systems, however, show great interest in connecting terminals located in the same room: "In this book, the use of wireless technologies enables, instead, to connect musicians located in the same room, in a large indoor space or in outdoor spaces. We may, thus, distinguish between indoor local, outdoor local or remote NMP" [11]. In the work presented below, performances took place in the same room, but with performers distanced from each other – around the audience – for an immersive listening experience. From a performer's point of view, mutual listening is modified, or sometimes even impeded, in these kinds of concerts, and the codes of chamber music have to be conceived of in a different way, hence the reliance on a local area network.

2. MAJOR UPDATES

2.1 Go web

When transferring a file from one machine to another which is located in the same room, web technologies often reveal the paradox that it is often easier to use the internet (e.g. WeTransfer), rather than finding a local solution. The latest technical improvement of SmartVox therefore consisted of hosting the application on the web (in July 2017). Since its conception in March 2015, the SmartVox app required the physical presence of a server in the room where the performance took place. The server consequently required node.js (server-side javascript), and the application [1] . This practical limitation prompted remote web hosting, which made it possible for performers to rehearse the piece together without the physical presence of the server on a computer. Since then, performers can rehearse physically in the same room – but also remotely –, and all that is required from them is a tablet (or phone) with access to the internet, in order to type into their browser the domain name of the piece to be played (e.g. nuages.smartvox.eu). This feature considerably simplifies the task of the performer, and lets us foresee great potential in terms of dissemination of the application. Whilst increasingly used in rehearsals, local servers are still obviously considered to be the best practice in a concert situation. With a reliable internet connection and relatively recent devices, the synchronization between different devices seems almost equivalent to those realized with a local server (see Section 4, Table 2 and 3). The measurements of Section 4 will therefore seek to determine how several factors (the distance of the server, its architecture and configuration, the quality of the network) impact on the loading and synchronization of the different files (the scores), which are prerequisites to a successful musical performance.

[1] The SmartVox web application, open source, is available here: https://github.com/belljonathan50/SmartVox0.1

2.2 Algorithmic composition/open form

The second major improvement of the application consisted of generating the form of the piece in real time [2] , but with fixed events on a local level. The code below shows a basic algorithm permanently used by the server accessible through **smartvox.eu**, choosing randomly which section will come next.

```
let openform = function () {
let timesArray = [0, 84, 123, 173, 262, 292, 362, 403, 517, 535];
let sectionDice = Math.random();
let sectionNumber = Math.floor(timesArray.length*sectionDice - 1 );
console.log(‘chosen section is ${sectionNumber +1}‘);
console.log(‘its seek value is ${timesArray[sectionNumber]}‘);
console.log(‘its end value is ${timesArray[sectionNumber +1]}‘);
let startTime = timesArray[sectionNumber];
let endTime = timesArray[sectionNumber+1];
let duration = endTime - startTime;
console.log(‘its duration is ${duration}‘);
let thisTime = timesArray[sectionNumber];
experience.sharedParams.update('seek', thisTime);
experience.sharedParams.update('transport', 'Start');
function myFunction() {openform();}}
setTimeout(myFunction, duration*1000);
```

Sandeep Bhagwati defines four categories of real-time music scores [7]:

- Permutational, where sections can be performed in a different order each time.
- Parametric, where more parameters are left free to the performer.
- Auto-reflexive, where the performer's actions have an incidence on the unfolding of the piece.
- Co-creative, where audience and/or conductor can interact with the display of the notation.

According to this taxonomy, *And the Sea* (the piece whose score is accessible through smartvox.eu) is permutational, as is, for instance, Pierre Boulez's 3rd Sonata, 2nd movement (*Formant 2 – trope*), in which the four sections (*Commentaire, Glose, Texte* and *Parenthèse*), can be performed in different orders. As Freeman observes, these new forms of algorithmic notation closely relate to the aesthetic of the *open-form*, and even constitute a revival of the essential questions addressed in the 1960s by Umberto Eco, Alexander Calder, and Earle Brown, among others. As Freeman states: "Real-time music notation systems draw from a broad spectrum of algorithmic composition environments that produce music notation. They are also influenced by an open-form aesthetic in which a musical score is read differently in each performance of a composition" [8]. However the generation of the material can obviously go beyond the mere permutation: "I outline a new form of computer-assisted composition, in which the author, in the classical sense, recedes and his artifact, the score – dynamically generated from algorithms – exists only in the moment of its creation" [10]. The idea of an ephemeral piece that permanently generates itself is in itself extremely attractive, but, by putting the performer in an unpredictable situation, its benefits can also be called into question: "While this immanence has often been perceived as a force for the emancipation of performers and spectators, it can also give rise to unaccountability" [7].

[2] To see the generation in action, go to **smartvox.eu**, choose a different instrument (e.g. piano, flute, cello) on each tab (or each device), and press the play button to unlock the video. After few seconds, the videos start wandering semi-randomly along the timeline of the video.

2.3 Client-side synchronization

An recent update in SmartVox consisted of implementing a client-side synchronization algorithm, exposed in Section 4.3, allowing for unprecedented temporal accuracy between players/singers.

3. CHALLENGES IN PRODUCTION

Production and research have different goals. A public performance in a concert hall demands reliable technologies, whilst the development of a notational environment such as the one presently described can only improve by testing its limits. The use of smartphones in rehearsals, workshops, or in pedagogical contexts is generally accepted with enthusiasm by musicians. This distributed system, however, still presents several risks in performance (see Section 4.1), and demands that its technical limitations be overcome in order to succeed in forthcoming productions.

3.1 And the Sea

And the Sea, commissioned by the SKAM [3] collective, was written for voice, cello, flute, and piano. SmartVox was originally dedicated to vocal ensembles, sending an audio-score as well as visual information; for this piece however, the instrumentalists only received visual notational information. In spite of the three major updates discussed in Section 2, on the day of the performance, the piece had to run locally (the devices were not connected to the internet, but to a LAN – Local Area Network), and was played from the beginning to the end [4], i.e. not in its algorithmic form (unlike the smartvox.eu website, where the timeline/form is constantly being generated once the player unlocks his video, pressing the play button). All the rehearsals until the day before the concert were nevertheless practiced and synchronized through the algorithmic-score website (**smartvox.eu**). The animated notation was also sent to the performers in the linear (non-algorithmic) version [5], and the performers never expressed the need or desire for a printed version of the score. The system proved to be helpful and easy to use for musicians; they could read their score independently without having to rely on a conductor, and could be placed very far away from each other: the singer was freely walking around the church (Figure 1) during the whole performance, the piano was on the altar, the cellist in the middle of the audience, and the flautist was in the organ loft. The animated notation also helped especially for the synchronization to the 8-channel tape of electronics. The placements of the speakers, finally, was also greatly simplified by the setup, since it did not require any sound card, nor the use of a mixing desk, but only a few cables and four phones, connected to two mini-loudspeakers each, and placed all around the audience.

[3] Stuttgarter Kollektiv für Aktuelle Musik: http://skam.io/

[4] The trailer of a piece performed with this score is available here: https://youtu.be/prcXUbhd-ZY

[5] The parts were available as youtube links. The piano part, for instance, can be accessed at the following address: https://www.youtube.com/watch?v=QByxPXItxHs

Figure 1. *And the Sea*, SmartVox used in performance with SKAM.

3.2 SmartVox, the piece

The SmartVox piece/production, for 5 soloists, choir, and electronics, was premiered in Nantes in March 2017 [6]. Involving a local choir each time, this project has a participative aspect that makes it financially viable. SmartVox will therefore be sung in several French cities in 2018-19: Metz, Rouen, and Caen. In spite of its focus on web technologies, the piece relates to ancestral traditions, first because of its sacred text [7], and secondly because of its polychoral construction: several choirs are placed in different locations, around the church. One of the aims was therefore to highlight the creative act that involves "reading early music today" [14], or any form of interpretation of ancient texts. The use of audiovisual notation for this piece was also justified by its microtonal language, because of the confusion that the notation of such intervals may cause to some singers. This use of notation as an aid for unfamiliar idioms relates to the work of G. Hajdu [4], who proposes that dynamic notation can provide solutions to the learning of non-standard (e.g. microtonal) musical practice. In this piece, the composition workflow consisted of analyzing the frequencies contained in a recorded or synthesized sound [8], in order to subsequently compose melodic lines within this harmonic grid. [9]

3.3 Le temps des nuages

This piece, premiered in January 2018 [10], sets poems by French philosopher Michel Onfray. It used *SmartVox* on a much larger scale than in previous attempts: five singers (the *De Caelis* ensemble), five percussionists (the *Links* ensemble), four channels of electronics, and 74 junior high-school students were placed around the audience. The technical challenge here was to handle eighty connections simultaneously. For rehearsal purposes, each separate part was accessible through the address nuages.smartvox.eu. The size of the concert hall (600 seats) and the number of con-

[6] A live recording and animated full score of the piece is available here: https://youtu.be/8R4Twc1A7Ks?t=1

[7] The piece is based on the old testament, in Hebrew and in its French translation by André Chouraqui, often de-constructed using algorithmic processes.

[8] An example of a capture in *Audiosculpt* [15] shows the spectrogram of a synthesized sound: https://youtu.be/8OlkZa7cTl4

[9] The same electronic sound is then used as a harmonic canvas: https://youtu.be/Xh1Vxe_lQ-U?t=66

[10] A recording of the piece is available at the following address: https://youtu.be/SyFdR2HiF00

nexions required three powerful wifi antennas in order to irradiate the whole room (where the singers stood). Node.js had previously experienced difficulty when too many clients requested heavy files in a short period of time. On this occasion, *nginx* (see Section 4.3) was successfully tested as *reverse proxy*, in order to manage load balancing, cache static content (the videos) and manage port contention between clients.

4. TECHNICAL CONSIDERATIONS

The above-mentioned pieces need to overcome a major difficulty: the devices of the participants are always different from one another. Although the performances of mobile devices improve very rapidly, unexpected behavior can always occur. Section 4.1 lists the causes to the problems faced by SmartVox since 2015, mainly concerned with synchronization and network overload. Section 4.2 measures the delay between phones in different situations. Section 4.3 exposes a solution which highly improved synchronisation across devices, developed by Benjamin Matuszewski in January 2018.

4.1 Description of frequently faced problems

4.1.1 Standby/sleep mode

In the former version, most accidents (in concerts and in rehearsal) occurred when the performers' device switched to sleep mode. A standby on the client-side was in most cases likely to cause synchronization problems, or even sometimes interruption of the WebSocket connection, in which case the only possibility that remains is to reload the page. A standby on the server side evidently interrupted the connection between clients. Sleep mode was the most problematic behavior of smartphones for this application, since the '*start*' message was only received by the phones once the sleeping period was over, hence causing long delays between parts. [11]

4.1.2 Other Breakdown factors

At its previous state, the app was already reliable in rehearsals: if one device got out of sync or disconnected itself, it could be updated on the next 'start' message of the conductor in most cases. For concerts, based on the results discussed in Section 4.2.1, if all the devices started exactly together, there was no need to adjust timing, since all devices could keep in time with each other. Whilst this way of performing music has been the object of great interest from nonprofessional singers, in performance situations, with more than twenty singers on stage, a single user interaction was often likely to disturb the beginning of the piece, which often only run smoothly (e.g. with all the singers in sync) only three or four minutes after the piece had started. Among these user interactions can be listed:

- Idle mode: switching between applications may disturb the clock synchronization and/or pause the video.

[11] iOs devices seemed able, unlike Androids, to receive a '*play*' message while sleeping; an iPhone on standby could start the video approximately at the time the 'start' message was received.

Figure 2. Time differences between several devices were measured with snapshot pictures.

- Putting on/taking off headphones often causes the video to pause.
- When playing a video in full screen, phones' media players usually propose a 'done' button, which enables the user to escape to the fullscreen mode; this action causes the video to pause.

One frequent problem encountered in rehearsal (and in concerts...) consisted of a pause of the performer's media player. As a workaround solution to this issue, in the algorithmic piece *And the sea* for instance, the '*seek*' update is always paired with a '*start*' message. [12] This message allows the user to cancel the 'pause' behavior exposed above.

4.2 Measurements of timing accuracy

As a conductor working regularly with this application, the main difficulty so far has been to cope with temporal issues, i.e. when the phones are not exactly in sync with each other [16]. Extensive research has been made in the domain of "Synchronization for Distributed Audio Rendering over Heterogeneous Devices, in HTML5" [17], which shows that extreme timeliness can be achieved. In the present study, measurements were realized in order to understand where dysfunction might come from, so as to improve timing accuracy in the context of rehearsals and/or concerts. The application being constantly used with a variety of devices, the measurements were made with different types of smartphones: iPhone 5, iPhone 4s, Oppo r5, Huawei C8815, Miui 8.5, HTC 802w. Figure 2 shows a typical snapshot picture taken while the playhead is crossing the staff from left to right.

4.2.1 Drift

A hypothesis has been put forward that the media-player integrated into the browser's phone may experience latency while reading the mp4 file, and subsequently cause a delay after some time. To measure this, 5 devices (p_1, p_2, p_3, p_4,

[12] In the coding example Section 2.2, the '*start*' message corresponds to experience.sharedParams.update('transport', 'Start');

	t_1	t_2	t_2-t_1	Drift
p_1	1"8	14"10	12"3	0"02
p_2	1"95	14"4	12"45	0"13
p_3	1"85	14"15	12"3	0"02
p_4	2"03	14"35	12"32	0 (reference)
p_5	2"00	14"3	12"3	0"02

Table 1. Temporal drift from p_4

	t_1	t_2	t_3	t_4	t_5	t_6
p_1	1"45	1"85	1"5	2"7	2"2	3"95
p_2	1"5	2"1	1"3	2"8	2"35	4"1
p_3	1"45	2"0	1"2	2"75	2"25	4"05
p_4	1"6	2"1	1"25	2"8	2"35	4"15
p_5	1"30	2"10	1"3	2"9	2"4	4"15
sum	0"35	0"35	0"35	0"25	0"3	0"3

Table 2. Local server with sync module.

p_5) were photographed simultaneously twice, while reading and displaying a seven-minute file displaying a score with a timeline. The times displayed on p_1 were 1"8, and 6'14"10 (the 6 minutes are not displayed on the table for clarity). p4 was chosen as the reference from which the drift should be calculated. The results of the experience (see Table 1) showed that the drift from p_4, being lesser than 100 milliseconds, can be considered null for our purpose.

4.2.2 Local server

As stated in Section 2.1, one of the main recent improvements consisted of hosting remotely the server that was initially used locally. The following measurements will try to determine how much a distant server impacts on the synchronization of the setup. The sum row adds up the differences between the mean value [13] and the other devices' values (see Table 2).

4.2.3 Distant server

The same experience with a remote server (i.e. accessing the application over the internet) reveals slightly greater sum values (see $\mathbf{t_5}$ and $\mathbf{t_6}$ in Table 3), and therefore, less precise time accuracy.

[13] For instance in Table 2, the mean value for the sixth measurement t_6 is 4"1, the value displayed by the second phone p_2.

	t_1	t_2	t_3	t_4	t_5	t_6
p_1	4"65	10"8	6"9	12"9	7"9	9"3
$\mathbf{p_2}$	0"05	3"75	2"85	4"55	2"05	2"15
p_3	4"75	5"75	6"8	12"85	7"95	9"25
p_4	4"5	10"75	6"7	12"82	7'80	9"15

Table 4. Distant server – different networks – $\mathbf{p_2}$ failed loading the page correctly.

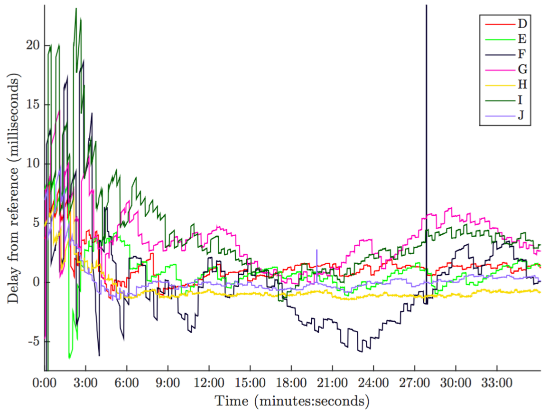


Figure 3. Synchronization of 7 heterogeneous devices. (Extract from J. P. Lambert [17]). The process still improves after 4-5 minutes.

4.2.4 Distant server – different networks (3G, 4G, Wifi...)

Each device is connected to the internet differently, i.e. only one of them is connected via local wifi. Phone N°2 is a recent iPhone (5s), the browser is Safari. The measurements were taken without reloading the page, highlighting a constant dysfunction of phone N°2: the intermittence of the 4G network on this phone may have contributed to the page load deficiency (see Table 4, $\mathbf{p_2}$). Once the page reloaded, the behavior was normal again.

4.2.5 Sync Module

The latest version of SmartVox has implemented a *sync module* [17] [18], which provides an elaborate solution that permanently computes an estimation of the reference time, in order to sync all the connected devices to the same shared clock. According to this reference time, the server can set-up a *look-ahead* scheduler, delaying messages (of e.g. 2 seconds), in order to leave enough time for all devices to receive this message at exactly the same time. Figure 3 shows that the synchronization of 7 heterogeneous devices gradually improves over time. A comparison be-

	t_1	t_2	t_3	t_4	$\mathbf{t_5}$	$\mathbf{t_6}$
p_1	4"1	4"55	3"35	3"30	7"60	5"0
p_2	4"1	4"6	3"35	3"25	7"55	5"0
p_3	4"0	4"45	3"25	3"20	3"90	5"9
p_4	4"15	4"6	3"35	3"25	7'65	5"05
p_5	4"05	4"5	3"20	3"35	7"35	5"0
sum	0"2	0"25	0"25	0"2	**3"95**	**0"95**

Table 3. Distant server – with sync module.

	t_1	t_2	t_3	t_4	t_5
p_1	6"15	5"85	5"45	9"2	1"4
p_2	6"4	5"9	5"4	9"45	1"4
p_3	6"3	5"9	5"55	9"05	1"4
p_4	6"5	5"85	5"9	9"55	1"4
p_5	6"25	6"4	5"75	11"05	1"75
sum	0"5	0"6	0"8	3"5	0"45

Table 5. Local server – without sync module.

tween table 2 and Table 5 demonstrates that the synchronization improves when the *sync* module is activated.

4.3 Synchronization update

The measurements of section 4.2.2 (performed on a local server) showed a satisfying synchronisation between devices. This confirmed the assumption that the important delay experienced between parts in rehearsals were most of the time due to user interactions or 'sleep mode' exposed in section 4.1. As a remedy to this issue, a solution was found by Benjamin Matuszewski in order to update dynamically the client's timeline, whenever he/she gets out of sync: every tickPeriod (for instance every second), on the client-side, the local time (syncTime) and the local timeline or transport (videoCurrentTime) are compared to the server's global time (triggerSyncTime) and global timeline or transport (transportTime). In the case presented here, if the difference (jit) exceeds 0.5, the local (i.e client-side) timeline is changed.

```
onUpdateTime(transportTime, triggerSyncTime) {
  if (!this.isReady)
    return;
  const syncTime = this.syncScheduler.currentTime;
  if (triggerSyncTime > syncTime) {
    this.syncScheduler.defer(() => {
      const videoCurrentTime = this.$video.currentTime;
      const jit = Math.abs(transportTime - videoCurrentTime);
      if (jit > 0.5) {
        this.$video.currentTime = transportTime;
      }
    }, triggerSyncTime);
  }
}
```

This new release of the application was used for two productions in 2018 (*Le Temps des Nuages* in January, and *Smartvox* in April), and gave promising musical results, with far greater clarity in the polyphony, and in homorhythmic responses between groups of singers.

5. GOING FURTHER

5.1 Dialoghi Spezzati

This piece, for twelve voices, twelve channels of electronics and *organetto*, was composed for the Mucem museum in Marseille and was performed with SmartVox. Since the application is essentially a multichannel video player, this piece explored the possibility of syncing live singers (each singer being guided by his/her audiovisual score) with filmed singers (displayed and accessed through the application, like the scores of the singers). An interesting dialogue could be perceived between live performers and recorded performers, displayed on screens. A natural continuation of this idea would be the implementation of WebRTC, adding visual and audio input and output to each terminal of the web application, to create a dialogue with remote performers.

5.2 Pedagogy in Classroom

SmartVox was tested this year (2017) on a weekly basis with 2nd year musicology students, in Aix-Marseille University. For this course, about Renaissance music, the application was particularly useful because it is mainly concerned with polyphony: each student could read and hear his own part on his device (phone, tablet, or laptop), with the full score [14] projected on the board of the classroom. The students were therefore able to sight-read up to eight-part complex polyphonies [15] with very little difficulty. [16]

5.3 Smartphones used as an instrument

The role of SmartVox is to turn the participants' devices into scores, but phones are often conceived as an orchestra of musical instruments rather than a notational tool [2][19]. With a similar architecture (a distributed web application), a wide range of user interactions can be imagined, mapping the user's gesture (such as screen-click, compass turn, accelerometer motion...) to a sample or a musical parameter. These types of musical experiments are strongly evocative of video games, and let us envisage playful forms of interactions with audiences. [17]

6. CONCLUSION

The present study concerns the realms of *networked musical performance* and *computer-aided performance*. The rapid evolution of smartphones, tablets and web technologies lets us hope that the technical problems listed above will soon be overcome. Musically, however, these limitations have strongly shaped the music I have written in recent years. In 2007, I realized experiments with a string quartet and four click tracks (a primitive form of wired networked performance), where the focus was put on extreme timeliness between players and electronics placed around the audience. [18] Years later, having accepted the temporal issues that can appear when working with web technologies, the focus was put on harmonic coherence, and tolerating a minimum of delay between parts, rather than on rhythmic exactitude. The present measurements have nevertheless shown that a more precise time writing can be achieved, thus allowing many different kinds of music or performative situation to be imagined.

Acknowledgments

I am very grateful to Norbert Schnell, who directed the COSIMA ANR research fund, and Benjamin Matuszewski, web developer at IRCAM, who coded SmartVox in the SoundWorks framework [2][20].

[14] For each piece, experience has shown that making a full score (conductor's score) available among all available parts is now considered best practice, since it is often very useful for performers.

[15] The scores are available on the internet, at the following addresses: tallis.smartvox.eu, josquin.smartvox.eu, dunstaple.smartvox.eu, dufay.smartvox.eu, avignon.smartvox.eu, canon.smartvox.eu; in order to conduct all the performers connected to the same score, the conductor should add the suffix /conductor on his device, e.g. canon.smartvox.eu/conductor

[16] The pieces studied in class were then recorded by the students in a church, with SmartVox, the result be heard here: https://www.youtube.com/watch?v=bofWvTCNNKI&t

[17] *Your Smartest choice*, by Huihui Cheng and Benjamin Matuszewski, was created at the Eclat festival 2017 in Stuttgart. https://github.com/ircam-cosima/your-smartest-choice. A demo is available here: https://www.youtube.com/watch?v=WKvhUJEE90k

[18] An extract of the piece is available here: https://www.youtube.com/watch?v=gOGMo_uwnlo

7. REFERENCES

[1] J. Bell and B. Matuszewski, "SmartVox. A web-based distributed media player as notation tool for choral practices," in *Proceedings of the 3rd International Conference on Technologies for Music Notation and Representation (TENOR)*, A Coruña, Spain, 2017.

[2] N. Schnell and S. Robaszkiewicz, "Soundworks – A playground for artists and developers to create collaborative mobile web performances," in *Proceedings of the Web Audio Conference (WAC)*, Paris, France, 2015.

[3] A. Agostini and D. Ghisi, "bach: An environment for computer-aided composition in Max," in *Proceedings of the International Computer Music Conference (ICMC)*, Ljubljana, Slovenia, 2012.

[4] G. Hajdu, "Dynamic Notation – A Solution to the Conundrum of Non-Standard Music Practice," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015, pp. 241–248.

[5] J. Bell, "Audio-scores, a resource for composition and computer-aided performance," Ph.D. dissertation, Guildhall School of Music and Drama, 2016. [Online]. Available: http://openaccess.city.ac.uk/17285/

[6] J. Bean, "denm (dynamic environmental notation for music): Introducing a Performance-Centric Musical Interface," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015, pp. 74–80.

[7] S. Bhagwati, "Vexations of ephemerality," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, Coruña, Spain, 2017, pp. 161–166.

[8] J. Freeman, "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Computer Music Journal*, vol. 32, no. 3, pp. 25–41, 2008.

[9] G. Hajdu, "Playing performers. ideas about mediated network music performance," in *Proceedings of the Music in the Global Village Conference*, Budapest, Hungary, 2007, pp. 41–42.

[10] ——, "Disposable music," *Computer Music Journal*, vol. 40, no. 1, pp. 25–34, 2016.

[11] L. Gabrielli and S. Squartini, *Wireless Networked Music Performance*. Springer, 2016.

[12] J. Lazzaro and J. Wawrzynek, "A case for network musical performance," in *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Port Jefferson, New York, USA, 2001, pp. 157–166.

[13] G. Hajdu, "Embodiment and disembodiment in networked music performance," in *Body, Sound and Space in Music and Beyond: Multimodal Explorations*, C. Wöllner, Ed. Routledge, 2017.

[14] J. Ferrando, "Reading early music today: Between reenactment and new technologies," in *Proceedings of the Computer Music Multidisciplinary Research conference (CMMR)*, Porto, Portugal, 2017.

[15] N. Bogaards and A. Röbel, "An Interface for Analysis-Driven Sound Processing," in *Audio Engineering Society Convention 119*, New York, USA, 2005.

[16] C. Chafe, "Living with Net Lag," in *Proceedings of the 43rd Conference of the Audio Engineering Society (AES)*, Pohang, Korea, 2011.

[17] J. P. Lambert, S. Robaszkiewicz, and N. Schnell, "Synchronisation for Distributed Audio Rendering over Heterogeneous Devices, in HTML5," in *Proceedings of the Web Audio Conference (WAC)*, Atlanta, USA, 2016, pp. 6–11.

[18] N. Schnell, V. Saiz, K. Barkati, and S. Goldszmidt, "Of Time Engines and Masters An API for Scheduling and Synchronizing the Generation and Playback of Event Sequences and Media Streams for the Web Audio API," in *Proceedings of the Web Audio Conference (WAC)*, Paris, France, 2015.

[19] J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang, "Evolving The Mobile Phone Orchestra," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Sydney, Australia, 2010, pp. 82–87.

[20] D. Poirier-Quinot, B. Matuszewski, N. Schnell, and O. Warusfel, "Nü Soundworks : using spectators smartphones as a distributed network of speakers and sensors during live performances," in *Proceedings of the Web Audio Conference (WAC)*, London, UK, 2017.

ELABORATE AUDIO SCORES: CONCEPTS, AFFORDANCES AND TOOLS

Sandeep Bhagwati
Concordia University Montréal
sandeep.bhagwati@concordia.ca

ABSTRACT

Over the past 18 years, I have repeatedly worked with auditive tools and audio scores that completely replaced any written score. The paper examines characteristics of the type of elaborate, autonomous audio score that I developed during this time, as well as attempts a preliminary classification of the compositional affordances that differentiate audio scores from visual scores. It describes the conveyance modes unique to audio scores; it touches on questions of control and context in elaborate audio scores, including on the question of whether such audio scores must necessarily be comprovisation scores; it details how, in the context of elaborate audio scores, the terms "practicing" and "rehearsal" describe other kinds of activities than they do in the context of visual scores; and it discusses unique problems of timing in the performance and composition of elaborate audio scores.

1. INTRODUCTION

1.1 Conveying Music through Sound

How do we make another musician make music - not any music, but a very specific musical *gestalt*, music that conveys a specific meaning, an adequate sensibility, an intentional emotion? In all musical cultures, this is a key question for music performance pedagogy. Not surprisingly, the answer usually is: anything that works – gestures, images, symbols, verbal descriptions. But most music performance teaching, even today, uses our ears: the teacher plays, the students imitate the teacher. Musical precision is conveyed most effectively through music itself.

The European practice of music notation, introduced into teaching as a mnemonic device among many others, a device initially well-suited to encode sonically abstracted pitch sequences but not much more, gradually evolved over a millenium to become the dominant channel for conveying eurological music from musician to musician. Over centuries, its always wildly heterogenous catalogue of signs and symbols expanded to encode many, but never all, of the gestures, images and auditory informations that previously had to be conveyed by personal contact.

But how many, precisely? No method transmits musical information free of loss or noise, especially complex niceties such as precise timing, dynamics or timbre. But music conveyance is not simply the transmission of information: each loss or misinterpretation significantly alters the aesthetic meaning conveyed. And musicking, while it may gainfully employ acoustic noise, is inimical to informational and structural noise.

European music notation has thus always relied on parallel, complementary channels of music conveyance: in teaching, the score is used as a support for the sonic and verbal conversations between students and teachers. In chamber music rehearsals, the score as a scaffolding saves time better used for discussions on finer points between musicians (and, if available, the composer), while in larger ensembles the role of the conductor has specifically evolved as a centralized music conveyor.

Conductors in performance, of course, exclusively use gestures and facial expressions to convey musical niceties, but in rehearsal they still often sing: the premise being that even a conductor's usually quite inadequate acoustic rendering of a musical passage can convey more specific musical information than a gesture, let alone words, could. Again, music itself, even a whiff of it, is experienced to be the best conveyance for music.[1]

1.2 Acoustical Cues

Acoustical cues, a feature of many musical practices around the world (e.g. colotonic gongs in gamelan, shouts in many African and afrological musics, cadential rhythms such as tihais in Hindustani art music), often function as mid-level temporal indices that shape structural features within a musical flow or coordinate ensemble phrasing. A special case of such acoustical cueing can be seen in click-tracks[2]: conceived initially to sync the inflexible time structure of tape(d) music with the unavoidably flexible timings of human performers, they quickly came to be used by composers who desired precisely coordinated control over the speed and the extent of tempo changes in an ensemble – or who wanted the musicians of one ensemble pursue individual tempo trajectories that would meet at specific moments: thus parametrizing time, as it were, both in its flux and in its synchronicities. It must, however, be pointed out that

[1] "Auditory models provide the only known method to develop an idea of how a specific instrument or passage should sound." [1]

[2] Mechanical Maelzel-type metronomes are a special case here: They indeed are acoustical prompts - but until Ligeti's 'Poème Symphonique for 100 metronomes' (1962), [2] they were primarily a rehearsal tool, not intended for actual performance. Also, other than the examples mentioned above, metronomes, with their inflexible, non-resettable pulse rate, do not offer kairotic cues: they offer a chronological framework. Click-tracks, initially used as metronomes for multi-track recordings, were much more flexible - they could be used in performance, and their pulse rate could be made to change over time.

while most acoustical cues in other practices are used as the best available solution to a problem of coordination, click-tracks need not actually be acoustical, and probably are not even an optimal solution: visual time cues would work as well (and might even be less disturbing to musicians). In live performance, the click-track was most likely adopted only because paper scores already hogged the visual channel.

Nevertheless, click-tracks - their technical infrastructure as well as many musician's familiarity with them - opened a window for the previously unknown type of score discussed in this paper: the elaborate audio score.

1.3 What is an Elaborate Audio Score (EAS)?

For the purposes of this paper, this term denotes a type of score that uses headphones as its interface to the musician and conveys musical information primarily via acoustical messages. If we accept the definition of a score as the collection of all composer[3]-defined, non-contingent aspects of a performance, audio scores, then, are scores that primarily use auditory communication to convey such composer-defined aspects to the performers.

These aspects can be conveyed in different modes and exercize various functions: information, instruction, imitation, inspiration, and instance (more on these terms below). These aspects will usually be conveyed in real-time, i.e. during the performance, although the last mode, instance, can be and has been used to complement a visual score.

In spite of their real-time bias, such elaborate audio scores need not necessarily be situative – they can be as fixed, and thus practice-able, as a written score. And yet, what is - and how it is - practiced will not be the same as in a written score: practicing such a score will tend more towards creative response than towards faithful execution, more towards exercising the imagination than exercising the fingers or the instrument.

Indeed, elaborate audio scores afford composers registers and opportunities of musical conveyance different from those possible in visual scores. They also exempt musicians from looking at a score, and thus free them to move around, and to use their eyes to take in other relevant information or to communicate, much as they do in improvisation or when music is played by heart.

Together with the possibility of conveying other registers of composerly intention to a musician, this unfettering of the musician's body and gaze may be the strongest motivation for composers to choose the audio score as their primary communication channel for their compositional ideas.

These ideas, based on a different interface and sensory mode, must therefore be different from those underlying a written or graphic score – it is my experience that composition for elaborate audio scores, especially for ensemble music, most likely will employ the compositional stance called "comprovisation", a complex intertwining of composition, structured improvisation and contextual improvisation – this, at least, has been the case in my compositions and comprovisations that use audio scores.

1.4 Developing an Elaborate Audio Score

My interest in audio scores already began with a very early score called "Music for the Deaf and Blind" (1985) written in my first year of composition studies at Salzburg's Mozarteum. In this piece, I had planned to let each musician in a classical piano trio play within a different sonic context – each would have a closed-concept headphone with different music, and they would be asked to play their written part along with the music in their headphones, not with their fellow musicians. This piece was never performed. Since 1999, however, I have been working with increasing frequency on progressively complex types of audio score. In *l'essence de l'insensible* [3] I used variable radio clicktracks enhanced with audio instructions to guide and coordinate 12 musicians through the sonically convoluted spaces of Richard Meyer's Stadthaus in Ulm (Germany), and to explore the aesthetic potential afforded by the difference between synchronicity and simultaneity. In *Nexus* [4] I used a continually reconfiguring live transmission network between five isolated musicians wandering in a cityspace to coordinate their musicking. In *Alien Lands* [5] I used a combination of animated score and audio score to enable the comprovisations of a spatially dispersed percussion quartet. In *Iterations* [6], I worked with live generated diverging and converging pulse paths, as well as with the "inspiration" mode detailed below that encouraged musicians to comprovise to a live DJ mix that the audience could not hear. During the gradual unfolding of a work cycle around a poem by Kabir, "I am a Bird from an Alien Land, my friend" (*Oiseaux d'ailleurs* [7], *Ham Pardesi* [8], *Fremde Vögel* [9], *On Nostalgia* [10], all for ensembles of 7-11 musicians), I finally developed elaborate audioscores that use all the conveyance modes listed below. Work on this elaborate audio score continued with *Villanelles de Voyelles* [11] for four singers a capella, and, at the time of writing, with "Ephémerides", a new project for large, distributed ensemble, to be premiered in 2019.

The work on all these projects is the primary source for the analysis outlined below. This paper, as my previous work on the scores themselves, does not refer to, rely on or relate in any decisive way to the work of other composers. While I was distantly aware of and sometimes, in media reviews, read about works such as Alvin Lucier's "Vespers" from 1968, which asks blindfolded performers to move in a space guided only by scholocation [12], Elisabeth Schimana's works that rely on what she calls "sounding scores" [13,14], the audio pitch and rhythm prompts for lay singers in Jonathan Bell's compositions [15], I never actually encountered these works live or studied them in detail during the years (1999-2015)

[3] Throughout this paper, the terms 'composer' and 'performer' signify roles, not persons. The role of the 'composer' can be filled by an individual or a collective, by a software or by a traditional method of intergenerational creation. The role of the 'performer' can be filled by a human instrumentalist, a singer, a programmer, a dancer or actor, and any combination thereof. Non-human sound producers, while sometimes regarded as performers in a wider sense of the word, either are usually not conditioned [animals] or not required or able [machines, natural phenomena] to parse and interpret verbal instructions conveyed by audio in a presentational performance context.

that I developed my elaborate version of an autonomous audio score.

If anything, I was more influenced and inspired by certain works of installation and performance artists such as by Sophie Castonguay, whose audio instruction score patch for "Le souffleur" (2010) [16] was developed by the same programmer who designed the audio score patch for my work *Oiseaux d'ailleurs;* by TC McCormack's performance project "Team Taxi" (2005) [17] where musicians sit in taxis who move around the city of Umea, Sweden, and create live music by emulating the sounds and events they hear on this trip; by Tino Sehgal's "This variation" (2012) where singers in a dark room at an exhibition take their cues and sonic material from the audience members coming to see the exhibition [18]; or by choreographers such as Xavier Le Roy, who upended the relationship between sound and the body in his "Mouvements für Lachenmann"(2005) when he asks the musician to just execute the movements that would be required to make Lachenmann's musique concrète instrumentale, but without any instruments – thus creating an inaudible, but mental music [18]; and finally Jerôme Bel whose "The Show Must Go On" (2001) [19] asks performers to only move in response different music's they can hear in their headphones.[4]

The reason, however, that none of these works had any real bearing on my research-creation towards an elaborate audio score is simple: with the possible exception of Castonguay, none of these projects was interested in repeatable, precise instructions – they all aimed to create ephemeral, improvisatory situations rather than the kind of repeatable and coherent constellations of sonic events that characterize polyphonic and multilayered music scores. These projects did not really care about any specific dramaturgical shape and/or sound of the resulting music, whereas my intention was to develop a conceptual tool that could precisely convey musical ideas, sonic materials and complex cochlear and temporal dramaturgies to musicians while they perform – albeit in a less abstract mode of representation than that of a traditional ink-on-paper score.

2. CONVEYANCE MODES

As mentioned above, in an elaborate audio score the composer's intentions may be conveyed to the musicians via different modes. It should be noted that all these conveyance modes are applicable to both real-time scores (when the audio messages are positioned, sequenced or even generated live) and offline scores (when audio tracks (i.e. parts) are prepared beforehand).

The difference between these score types will mainly impact production modalities, such as the nature of practicing and rehearsing (see section 4), or the preparation and integration of live vs. pre-recorded sonic materials. The sole difference they make to conveyance is quantitative: each score type will need a different set of conveyance modes and will weigh their importance differently.

[4] Ineed, while watching this show in Berlin in 2005, I was strongly reminded of my own abandoned "Music for the Deaf and Blind".

2.1 Conveyance Mode A: Information

Cues are the most basic of auditory signals. They usually inform the musician about their spatial or temporal embedment or their place within the dramaturgy of an evolving performance. They assume that the musician knows what to do with this information and do not usually offer specifics.

Cues can take the form of a variable /intermittent/continuous click-track, a count-down to the next change, or a kairotic cue-list ("Cue for your Solo: start NOW!"). Cues could also inform the performer about aspects of a performance that require no immediate action or reaction ("next pitch set in 10 sec", "spatialisation mode 3 is now active") or connect the performer to other participants ("Singer expects your cue", "Next cue from trombone").

A special kind of cue is the pitch cue: A musical pitch (played as a tone, not verbally denoted) which the musician does not imitate, but which informs the performance: the most obvious of pitch-cues would be a drone. Another example could be an upper-pitch limit that the improvising musician should not surpass, or a pitch-attractor, around which an improvisation should weave itself. While these tones themselves are purely informational, they, of course, must be pre-faced with an instruction that tells the musician how to extract this information from them.

Cues, while basic, can nevertheless decisively shape the music: most dramatically in the case of a click-track with varying speeds, or one in which individual tempi diverge and then re-unite again. They also can be essential for the performance of a live-generated auditory score, where a performer needs to be prepared in advance in order to be able to act on upcoming messages.

2.2 Conveyance Mode B: Instruction

Instruction messages, for a composer, will feel like the closest analogy to a visual score: they actually tell a musician what to do at a given moment. Nevertheless, the type of instructions that are possible in an audio score are quite different from those in a visual score. Visual notation affords the composer detailed control over fast-moving structural detail, especially with regard to pitch sequence and duration. Audio scores, mainly because inhabit the time of performance itself, and cannot be previewed, cannot specify temporal details in similarly fine detail: hence, their instruction set will always be limited to comparatively broad strokes.

Instructions come in several types: musical, interactional, para-musical and indexical. *Musical* instructions provoke musical structures that concern only the musician receiving the instruction; *interactional* instructions concern the musical relations between two or more musicians; *para-musical* instructions direct the performers to enact non-sonic behaviours; and *indexical* instructions point to, explain, and set up other conveyance modes.

2.2.1 Musical Instructions

While musical instructions in audio scores cannot shape musical structure in deep detail, they can provoke a more or less creative enactment of such structures. Such enactments can take different forms:

a) *recall:* instructions refer to material previously committed to memory (“Play Melody X”, “Play Rhythm Y”)
b) *adapt:* memorized musical fragments are used as material to be transformed into the current context (“Play Melody X to fit/counter the current tempo/time signature/register”, “Play Rhythm Y in triple time” etc.)
c) *create:* instructions describe the music to be played in a rather comprehensive fashion (“Play a sad / upward moving / triadic etc melody” , “Play a jerky / groovy / rigid beat” etc.). Perfomers must then invent a music that fits these descriptions.
d) *tune:* musicians can be given precise pitches to play. This can be especially useful in microtonal contexts, and indeed seems one of the more practicable and reliable scoring solutions for precise microtonal tunings. It, of course, will work only with slow moving pitch material. In live-generated scores, this format can also help tune the musicians to other sound sources, such as an environmental sound.
e) *conduct:* each musician can be given precise cues for starting and stopping, for the precise evolution of dynamics and pulse, and for the coordination with other musicians. These are tasks that usually are relegated to conductors. Audio scores, however, are a unique tool that can be used by composers to shape each of these musical parameters as they happen, and this separately for each musician or sub-ensemble.

2.2.2 Interaction Instructions

These instructions ask the performers to connect with other performers or with their environment – sonic or otherwise - in various ways. Such instructions can range from “Imitate performer x” to “Accompany performer Y” or even “Disturb performer z”, or other interactional behaviours. And they can focus the interaction on specific elements of another’s performance: “Follow the pitches of Z but in another rhythm” or “Match timbre with Y” or “Create a rhythmical dialogue with X”.

Similar interactions with the environment fall into this category, if they do not only reflect the sonic landscape (that would be more an imitative behaviour, see 2.3) but imply an interaction with it (“Trumpet: make the piano strings resonate” or “Accentuate/Satirize a conversation happening nearby”).

2.2.3 Para-musical instructions

Freeing the performer’s body and gaze implies new compositional parameters: directionality of body and gaze, body posture, the musician’s position and trajectory, etc. These can be integrated into a score in flexible ways previously difficult to define (“During the next 6 seconds: On a high pitch, quickly turn 180° while singing” or “When you hear a mordent from someone, slowly walk towards this performer”, “Turn away from the loudest among you.”).[5]

Such parametrizations can be used musically (mainly for flexible, improvisable, emergent types of sonic spatialisation as well as for re-configurations of the ensemble) as well as theatrically or choreographically.

2.2.4 Indexical Instructions

These are instructions that set up other conveyance modes: after all, the sound examples that are used as reference in the Imitation, Inspiration, and Instance modes (see below) are not self-explanatory – they need to be framed and defined by an instruction. (“Mimic the following sound”, “Accompany the following sound”, “Improvise like in the following sound”). Similarly, such instructions can set up and define a cue (2.1.1.) (“On next three cues: change timbre”).

2.2.5 Wording

A final remark on the wording of instructions: there is a musical necessity to be as precise, unambiguous and concise as possible. Musical time is so much more finely grained than verbal time - and the longer or complex a message is, the more music time it consumes – both on hearing and when it is processed by the performer. In addition, the longer an instruction the greater the risk that it is not fully retained or understood by the performer (who, after all, is usually playing while listening to the instruction). At the same time, in a comprovisation context, instructions do not really work effectively if they are commands that must be followed blindly – they need to be experienced as hints that open possibilities rather than constraints that close down options.

I have frequently found the wording of instructions to be a aesthetic/creative act in itself, not unlike writing poetry.

2.3 Conveyance Mode C: Imitation

Set up by the indexical instruction “Mimic the following sound” the performer aims to closely lock into a synchronized (or, if possible, responsive echoing) imitation of a sound example heard in the headphone. The composer is completely free to use any sounds as sound examples[6] – a part of the interest in this feature will be the actual, physical inability to exactly imitate the sounds presented on one’s instrument: e.g. when a flutist hears a waterfall’s bass rumble, or a keyboard player hears a microtonal glissando. The strain to imitate the impossible will pro-

[5] While musicians do move about in other types of music, such movements are either memorized (marching bands, choreographed performances) and optimized for the audience – or spontaneous and optimized for the performers. Audio scores allow musician movement to be developed further, into very specific configurations between choreography and spontaneity.

[6] As far as I can discern, there are no limiting constraints for the kind of sound example that can be used, beyond the insight that the more complex a sound example is, the shorter it should be for imitation, inspiration and instance to work at all: the musician must, after all, get a fair chance to absorb the example in its entirety and in its details before reacting to it.

duce music that the performer would not have used in the course of their usual idiosyncratic improvisations.

An interesting aspect of this approach to imitation is the insight that the sound example will never be imitated perfectly – and that embracing this impossibility opens another door: just as Chinese script characters enable the same thought to be communicated and spoken in widely different dialects and languages, the imitation mode enables musicians of widely different traditions and instruments to create the same sonic dramaturgy within their own sonic reference frame, even though their individual realizations of the sound to be imitated might differ wildly.[7]

A special case of this (and the two following modes) would be the invitation to mimic sounds and sonic structures *outside* the performer's headphones, in the immediate or mediated environment. This introduces even more contextual chance elements into the score, and seems to require a kind of default instruction that kicks in when, for any reason, the environment does not afford anything that the performer could use as a sound example.

2.4 Conveyance Mode D: Inspiration

Set up by an indexical instruction that specifies an interational relationship with a sound example such as "Accompany/accentuate/satirize/simplify etc the following sound" the performer uses the sound in the headphones (or outside) to orient her/his playing in the interaction mode defined by the instruction. This orientation is not mimikry in the sense of the previous mode, but rather a way of playing that takes off from the example, expands, comments, counterpoints it. This includes the possibility that the musician will play something that is not similar to the sound of the example, but emerges from a musical dialogue with it.

Interestingly, these interaction modes usually describe social or structural relationships rather than musical ones. In effect, the player treats the sound example in the headphone as if it emanated from another performer or other performers - and plays with these "other performers" according to their mutual musical and social positioning.[8]

One can, of course, ask a performer to be inspired by the sound example in a strictly musical, compositional manner (e.g. "play a floridus counterpoint to the example", "play the example as a New Orleans jazz phrase", "only play spectral overtones of this sound" etc.). This, obviously, will limit the choice of performers to those able to easily navigate such technical or stylistic constraints. But such a musical constraint can also be productive if used against the grain.

For example, I have found it musically interesting, in working with ensembles consisting of musicians from different traditions, to generalize such instructions to e.g. "play this example as it would be played in your tradition". In this way, aesthetical choices (here, an interest in composing with the differences between musical mannerisms) can determine and redefine the function of particular modes of conveyance.

2.5 Conveyance Mode E: Instance

In this mode, the sound example the musician hears in the headphone[9] is used indeed as an example, one instance of a particular style of musicking that the performer is expected to realize. These examples are, in a sense, seeds for a specific music to come: everything about them can be important and become a guide to improvisation.

As a composer, one can either rely on the performer's ability to both intellectually and intuitively grasp the specifics of this particular instance of possible musicking – or one can specify those aspects of the sound example that could become generative in the context of the current performance: "Take the rhythms and improvise with them", "Develop the example's melodic movement", "Like in the example, play with timbral changes" or a similar focus on other parameters.

Instances can be used as examples in the legends of visual scores, too (I have, for example, used them to specify and differentiate different types of glissando, or to show a specific desired voice quality). In an audio score, they become a powerful and enabling live comprovisation tool.

The three last approaches delineate three different interactions with any given sound example: *imitation* engages in sonic mimikry, *inspiration* engages in musical elaboration while *instantiation* is a process of analysis and continual re-construction.

3. SCORING

3.1 Comprovisation

Most music traditions arise from the fact that those aspects of a performance that need to remain coherent from one performance to the next and those that can be left to contingency, context and improvisation tend to converge on a stable, praxis-based mix: each tradition 'selects' a unique constellation from among all the possible permutations of performance parameters[10]. Further musicking in such a tradition is then determined by this constellation.

[7] Of course, it is possible that a composer really intends to have a performer imitate a sound example perfectly, down to the inflections and microtimings – then the performer should have the opportunity to practice this imitation beforehand – it effectively becomes a sonic *objet trouvé*.

[8] This reminds us that *all* sound examples could, in principle, also come live from other performers – whether they are in the same space or are telematically connected. Indeed, elaborate audio scores, and the elastic timing discussed in this paper, could be used as a powerful scoring tool in telematic performances.

[9] The sound examples used for imitation, inspiration and instantiation can be, of course, taken from existing music / field recordings – but they also can be newly composed and recorded specifically for the sonic context of this piece. This would mean that a significant part of the composer's sonic creation may be inaudible to the audience – if the composer does not decide to use this material in the performance, too – either as memorized performer scores or as part of an audio track played back in the space.

[10] i.e. pitch, duration, timbre, acoustics, spatialisation, but also conventions of the performing body (posture, dress, movement), social relationships between performers, signalling between performers and many more. Each of these performance parameters, in most musicking traditions, is set within such narrow ranges of acceptability that even minute deviations or tweaks can have huge aesthetical import – a fact often and strategically exploited, for example, by the avantgarde movements in eurological art music over the course of the 20th century.

For musicians within a specific tradition, its axiomatic constellation of performance parameters will over time become unquestioned and invisible. For example, western classical musicians usually not ask themselves why composers in their tradition (who mostly do not play with them) have readily provided them with pitches and rhythms and articulations - but often have left performers to figure out vibrato, portamenti, rubati or the kind of reed they use etc. They do not question this particular choice of parameters, but rather accept it as their baseline - and focus their creative energy on shaping those “surplus” parameters that their tradition leaves undefined.

Comprovisation, in contrast, is a creative mode in which composers, for each new piece, must decide the specific constellation of parameters that are to remain unchanged from one performance of the piece to the next, as well as those that are to be decided in the performance context [21]. Such decisions are often guided by several categories of constraints - cognitive (how many different and separate parameters can a musician control while playing), social (how much minute aesthetic control over a performer is socially acceptable, to what degree is a score perceived as an invitation for co-creation rather than as one where performers ‘execute’ the directions of an author) and, for a large part, technical/ structural/ organisational (available instruments and technology, players’ abilities and preferences, acoustics of available venues, can players hear/see each other, etc.).

The elaborate audio score, initially defined primarily as a specific interface and mode of conveyance, has already been shown to afford and privilege certain modes in which aesthetic or pragmatic information can be conveyed to the musician. There is, however, and for now, no particular school or aesthetic tradition based on audio scores, i.e. there is no “conventional” set of performance parameters, conveyance modes and sonic behaviours that performers and composers can regard as given when they embark on musicking with an audio score. This situation thus requires composers to constantly think about defining their own selection of performance parameters, almost anew for each artistic project: their creative mode for using audio score thus must be comprovisation.

As mentioned above, audio scores are not ideally suited to prescribe, describe or control fast-moving, non-repetitive details of pitch sequences, durations or articulatio. Instead they allow composers to inspire ensemble musicians to realize sonic behaviours that transcend the limits of written notation – and to coordinate them in ways impossible for improvisers. Many of the sounds and sonic behaviours resulting from audio scores will, of course, be familiar both from improvised and from composed music. But in an audio score, they can be sequenced and arranged in conceptually and/or dramaturgically elaborate musical relationships and ensemble constellations that transcend both the barely situative written score and the bare scaffoldings or the entirely emergent dramaturgies of improvised music – they enable complex architectures of ensemble comprovisation.

Moreover, audio scoring enables a composer to devise performances on the basis of <u>any</u> sonic behaviour whatsoever – including those that in the normal course of improvisation or sound production would require lengthy emotional/musical build-ups or that musicians would never use instinctively in their improvisations.[11] Such extra-traditional sonic behaviours can be coordinated and sequenced in utterly non-improvisational ways, while retaining their ontological openness for improvised sonic realization. As such, audio scoring is a creative mode that straddles both composing with conventional and graphic visual notation (imagining sounds, providing prompts to realize an imagined sound) and composing electroacoustic music (working with each sound as it is, without considering with its reproducibility or re-creation).

3.2 Timing

3.2.1 Precise timing

Tracing the advanced audio score back to click-tracks as one of its forerunners highlights one of the most obvious affordances of audio scores to the composer: perfect control over timing. Not only is it possible to enable groups of musicians to play in precisely coordinated variable tempi (rubati, accelerandi, ritardandi etc), but such variable tempi can also be composed polyphonically, allowing a different temporal evolution for each musicians while ensuring that all converge on a new common tempo at a later moment.

While such advantages certainly are useful, they are not applicable to all musical situations: Accelerandi and ritardandi often are more expressive when they are not precise, and rendered ad-hoc to fit the dramaturgical context. Diverging and converging tempi or poly-temporal rhythms, in order to become aesthetically perceptible, usually require the musical material itself to be restrained and concise – and such restraint may well run counter to stylistic or improvisatory affordances.

In many cases, click-tracks, whether pre-recorded or live-generated, simply are not optimal solutions for a desired outcome. For example, musicians in imitation mode will often be attracted to or perturbed by any timings in the sound example, and many instructions effectively generate their own temporal structure which may clash with the abstract pulsations of a click-track.

Lastly, audio scores, unlike visual scores, confront musicians with a score element, a message or instruction *in real-time*. One cannot, in an audio score, glance ahead towards things to come – rather, each instruction and example in the score arrives in the actual present, and must be processed (i.e. understood and musically realized) immediately. But this moment of immediacy has an indeterminate duration – each musician will react more or

[11] Another important affordance of the audio score is that it can be scored in ways that are culture/tradition-agnostic: Precisely because aesthetic intent is conveyed by a combination of natural language and recorded sounds, and not by culturally specific notation conventions, musicians from different traditions will, for the most case, understand and work with the audio score quicker, more reliably and with less stress than with other kinds of notation. The score itself requires no cultural adaptation or learning, once its basic functioning is understood. This, of course, is not to say that what musicians from different contexts will hear and how they interpret it will be the identical – the sonic realization of an elaborate audio score may vary not only from one tradition/culture to another, but also from individual musician to individual musician.

less promptly to an instruction, and may take a different moment to process it into actual sound. Often, especially in live-generated scores, such instructions may arrive at any moment in a musical flow, and in certain stylistic or musical contexts, the musician may need to "wind down" the current utterance before taking up the new instruction. In music, however, aesthetically relevant coherence coordination is a matter of split-seconds – and the slightest of such hesitations could thus destabilize a music that relies on precise click-track compliance for its aesthetical import.

3.2.2 Heterophonic Elastic timing

Audio scores are a tool suited particularly well to what I call 'heterophonic elastic timing', i.e. a mode of temporal ensemble coherence that is neither rubato (localized pulse variance) nor swing (localized variance in pulse/attack couplings) nor, of course, straight "playing-on-the-beat". It also is different from kairotic, inner timing in solo improvisations, because, although it may appear similar, heterophonic elastic timing can only really apply in an ensemble setting: the term describes a particular type of coherence between different musicians.

Heterophonic elastic timing occurs when a score is not only tolerant to the minute differences between individual performers in reaction time, processing time, and individually felt fit to the current musical activities – but when it actually embraces and expects such individual aberrations within the ensemble, usually in the interest of a larger goal: this could be a maintaining an emotionally/kinesthetically convincing flow, or an interest in perturbances and their effects on musical dramaturgy etc.[12]

Performers in my audio score pieces have likened the experience of playing in heterophonic elastic timing to the coordination of fish or birds in a swarm: a common trajectory is followed, but nevertheless each participant in this swarm has a certain leeway in seeking their way – for example if one encounters an obstacle, or if winds or currents require adaptation. In an audio score with heterophonic elastic timing, performers are effectively asked to coordinate *dramaturgically* (i.e. by ear), while the temporal flow weaves in and out of synchronicity.

A special case where precise <u>and</u> heterophonic elastic timing are both applicable in audio scoring is the situation in spatially dispersed, and maybe even spatially mobile ensembles: here, a precisely synchronized audio score can serve as the rigid conceptual scaffolding for a music that will sound quite elastically timed, simply because each listener will be at a unique location that is defined by a specific set of time lags for each musician, depending on the distance of the musicians. A composer could make use of this effect by writing exactly the same rhythm for all musicians, and then let the position and movement of the listener 'compose' a flexible spatial canon.

3.3 Situative and Fixed Audio Scores

Audio scores occupy a curious middle ground between situative and fixed scores. If we follow the definition of situative scores, as "scores that do not build on linear, pre-existing information structures. Information in these scores is only available ephemerally, i.e. while it is displayed or accessed in a particular context" [24] then audio scores are situative scores – during performance, every instruction or example is only ephemerally available to the performer around the time of its realization. And in the case of live-generated audio scores, this assumption holds water.

However, both in my work and that of others, the audio score has also been used in a fixed format – the individual performers' tracks, like orchestral parts of a written score remain the same for any performance, and can even be played on mp3 devices, their start synced by gestures. In this case, the individual part itself is no more situative than a written score – each performer can play it back to themselves and, if it helps, even learn it by heart. The audio comprovisation score is fixed and repeatable – which means it can be rehearsed, much like any other visual score.

4. PRACTICE AND REHEARSAL

In elaborate audio scores, the rehearsal is an important facet that guides their implementation and even composer's choices.

The performance of audio scores usually requires fewer ensemble rehearsals than a complicated chamber music composition and more than a free improvisation concert. And it usually requires more individual practice and exploration than both the chamber music concert and, most likely, also a free improv concert. What are the demands on a musician performing the kind of elaborate audio score discussed in this paper?

In any audio score comprising more than the most basic of elements (durations and pitches), the particular set of instructions first needs to be learned and understood. As mentioned in 2.2.5, the constraints on the wording of instructions are intense, and almost always will require the composer to use short-hand terms for more complicated ones, and explain them in the legend. In this, the first approach to an audio score is very similar to that needed for a conventional new music score that uses many non-standard symbols.

Once the musicians understand all the instructions, they might need to practice particularly demanding passages, just like in any other score. The difference, however, that these passages will only rarely be demanding for their fingers or larynx – rather, the difficulty in these passages will mostly pose a conceptual or creative challenge: How to create engaging and convincing music in imitation, inspiration or instantiation of a given sound example – especially when the score affords only a fairly short window of a few seconds to make such a musical statement? In my experience, the only truly virtuosic

[12] The concept of elastic timing itself is, of course, no invention of the author [22], [23]: several Asian traditions, such as *sanjo* and *p'ansori*, the music of *gagaku*, *gugak*, or *jingju* orchestras, as well as heterophonic chanting practices from Vietnam to Georgia are built on elastic timing as described above, as are drumming traditions in sub-Saharan Africa. The unique contribution of the elaborate audio score to elastic timing is the fact that each voice can be elastically timed in a different way, not only in a single temporal flow.

challenge in practicing an audio score tends to arise with complicated click-track led tempo changes and pulse-based improvisation.

The main questions that need to be addressed in subsequent ensemble rehearsals usually are again very different from usual orchestra, chamber ensemble or band rehearsals. Coordination in time and pitch, in phrasing and in musical inflexion, the great time devourers in usual rehearsals are almost absent from the audio score rehearsal process – as delivering exactly these parameters to the musicians is the great *forte* of such scores. Most rehearsals I have witnessed tend to use the available time to focus on the musical interaction between the musicians, on understanding one's role in a larger context and, as a consequence of this understanding, on exploring one's responses to the instructions and sound examples. In rehearsing an audio score, musicians, much like theatre actors, need to understand the musical persona their engagement with the audio score brings forth from inside themselves.

5. PERFORMANCE

5.1 Interface and Infrastructure

Audio scores, while using a comparatively recent technological interface, are not currently in dire need of ongoing technological development – they rely on existing technologies. In fact, today's audio and wireless technologies require between none and very minor tweaks in order to be appropriate for all kinds of audio scores for the foreseeable future.

All an audio score requires are interfaces to the musician's ear(s) (typically: open-concept headphones), a device providing the sequence of acoustic conveyances that make up the audio score, and, for some uses, a centralized, multi-channel audio dispatching system. If musicians are expected to move through space freely (after all, one of the primary motivations for using an audio score) then this dispatching system must be wireless. All these technologies have for some time already attained commercial viability and reliability, and are commonly used in commercial branches of the live entertainment industry as well as in a variety of non-artistic professions such as the military, police, or large construction sites.

Likewise, any software that would control the score or the multi-channel dispatcher is comparatively easy to come by: in many cases, basic functions of studio sequencing softwares are largely sufficient, and if not, multi-channel real-time composition software frameworks are comparatively easy to program. While it is conceivable that a specialized audio score composition software might emerge, there currently seems to be no need for one.

The only remaining source of technological uncertainty concerns the synchronization problems that may emerge in future, more evolved and data network-centric instantiations of the audio score[13] when many wireless data channels within close range must be kept in sync with one another. Interference, critical dropouts and unpredictable variations in latency can be assumed to remain vexing nuisances. Should the realization of an audio score therefore require split second coordination, analog radio transmission has so far proven to be the more reliable option.

5.2 Ensembles

As already mentioned above, the most obvious use for audio scores in music is an ensemble – in principle, of any size.[14] For the audience, the interplay of synchronicity and diversity, the joys of co-incidence and divergence, the seemingly unconducted and unexpected kairotic moment as well as the richness and tangibility of quickly changing, observable spatialisation through moving musicians are essential aesthetic assets of performances using an audio score, as can be the more choreographic or theatric possibilities such a score affords the composer.

All these would obviously remain absent in a solo score – the one exception being: a solo musician performing to a live-generated audio score that in a specific, artistically insightful and perceptible way connects the comprovisational solo to the audible or visible, but ostentatiously non-composed, contingent context, environment or situation of the performance.

6. CONCLUDING REMARKS

As we have seen, audio scores, at first blush merely a new type of interface, create new affordances for composers, require new approaches to playing with a score for performers and afford new aesthetic experiences for audiences. A widespread use of this interface would thus likely lead to new aesthetics of musicking. Competent and insightful reflections on such a sea change, however, would require detailed musical and theoretical analyses of actual comprovisation works that use audio scores.

This paper intends to provide some tools for such analyses, and for the ensuing aesthetic discussion. But most of all, it is a composer's invitation to other composers, a little manual of how to approach and think through composing with this relatively new and, as far as I can see, not yet intensively explored score interface for novel types of communications between composers and performers.

Acknowledgements

Research-Creation leading to this paper was financially supported by the Social Sciences and Humanities Research Council, the Canada Research Chairs Program, the Fonds Quebecois de Recherche – Société et Culture, the Canadian Congress for the Humanities, the Conseil des Arts et Lettres de Quebec, the Hauptstadtkulturfonds Berlin, and the Sociéte de Musique Contemporaine de Quebec. It was artistically supported by matralab (Concordia University), Stadthaus Ulm, Radialsystem V Ber-

[13] e.g. ones using sensor data and/or individual score processors on each musician's body etc.

[14] The limitations, as always, are mainly technical (wireless channels) and financial – but, also as always, one really needs good musical reasons for performances above a certain size of ensemble.

lin, Ensemble Supermusique Montréal, Bye Bye Butterfly Percussion Quartet, and Ensemble Extrakte Berlin. The author would also like to thank: Dr. Martin Scherzinger, my audio score software developers Matthieu Marcoux and Joseph A. Browne, as well as all the many musicians who tested the different versions of the Elaborate Audio Score for valuable insights and hints.

7. REFERENCES

[1] G. L. Duerksen. *Teaching Instrumental Music.* Music Educators National Conference, Ann Arbor: University of Michigan Press, 1972.

[2] G. Ligeti, "*Poème Symphonique" for 100 metronomes.* London: Boosey & Hawkes, 1962.

[3] S. Bhagwati, "*L'essence de l'insensible" for dispersed ensemble with clicktrack* unpublished score manuscript. Berlin, 1999.

[4] S. Bhagwati, "*Nexus' for 5 itinerant musicians with musical audio scores*, unpublished score and software, Montréal 2010. [Online]. Available: http://matralab.hexagram.ca/projects/nexus/

[5] S. Bhagwati, "*Alien Lands" for 4 distributed percussionists with animated and live-generated scores*, unpublished software score, Montréal 2011. [Online]. Available: http://matralab.hexagram.ca/projects/alien-lands/

[6] S. Bhagwati *et al.*, "*Iterations" for speaker, 8 musicians, 2 silent DJs, headphone installation and interactive clicktrack* unpublished score & software, Berlin, 2014. [Online]. Available: https://vimeo.com/120307891

[7] S. Bhagwati, "*Oiseaux d'ailleurs" for 11 musicians with written score and live-performed audioscore* unpublished, Montréal, 2011. [Online]. Available: http://matralab.hexagram.ca/projects/oiseaux-dailleurs/

[8] S. Bhagwati, "*Ham Pardesi" for 8 itinerant musicians with pre-recorded audio scores*, unpublished audio score, Montréal, 2014.

[9] S. Bhagwati, *"Fremde Vögel" for 7 itinerant musicians with pre-recorded audio scores.* unpublished, Berlin, 2015.

[10] S. Bhagwati, "On Nostalgia" for 9 musicians with pre-recorded audio score. in: *Ensemble Extrakte, Treatises on Trans-Traditional Musicking* [CD]. Berlin, 2017. Track 13.

[11] S. Bhagwati, "*Villanelles de Voyelles" for 4 singers a cappella with pre-recorded audio scores.* unpublished, Montréal, 2017. [Online]. Available: http://matralab.hexagram.ca/projects/villanelles-de-voyelles/

[12] A. Lucier, *"Vespers" (1968) for blindfolded performers with mobile echolocation devices.* In: A. Lucier and D. Simon, *Chambers: Scores by Alvin Lucier.* Middletown: Wesleyan University Press, 2012. p.15–27.

[13] E. Schimana. *"Virus #1.0-#1.7" for live generated electronic resonating body and diverse instruments.* [Online]. Available: http://elise.at/project/Virus_1

[14] E. Schimana, *"Vast Territory. Episode 1 Lily Pond" for violin, viola, cello, bass clarinet, flute and sounding score*. [Online]. Available: http://elise.at/project/Vast%20Territory

[15] J. Bell and B. Matuszewski, "SMARTVOX. A web-based distributed media player as notation tool for choral practices," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017 pp. 99-104

[16] S. Castonguay, "*Le Souffleur" performance for six performers with headphones.* [Online]. Available: http://www.sophiecastonguay.ca/index.php?p=lireprojets&idprojet=15

[17] TC McCormack, *"Team Taxi" performance for 6 taxi drivers and 6 musicians.* [Online]. Available: http://www.tcmccormack.co.uk/work/team-taxi.php

[18] C. Bishop and T. Griffin, "No pictures, please: Claire Bishop on the art of Tino Sehgal," *Art Forum*, vol. 43, no. 9, 2005.

[19] X. Le Roy, *Mouvement für Lachenmann. Staging of an evening concert.* [Online]. Available: http://www.xavierleroy.com/page.php?sp=e347f884fa37480bd0bd5dff79104483a8e284b5&lg=en

[20] J. Bel, *The Show Must Go On*, 2001. [Online]. Available: http://www.jeromebel.fr/index.php?p=2&s=6&ctid=1

[21] S. Bhagwati, "Comprovisation – Concepts and Techniques" in H. Frisk, S. Östersjö (Eds.), *(Re)Thinking Improvisation.* Malmö: Lund University Press, 2013.

[22] R. Polak, "Rhythmic Feel as Meter: Non-Isochronous Beat Subdivision in Jembe Music from Mali," in *Music Theory Online – A Journal of the Society for Music Theory,* vol. 16, no. 4, 2010.

[23] M. Scherzinger, "Temporalities," in A. Rehding, S. Rings, *The Oxford Handbook of Critical Concepts in Music Theory.* New York: Oxford University Press, 2019. (In press)

[24] S. Bhagwati *et al.*, "Musicking the Body Electric. The "body:suit:score" as a polyvalent score interface for situational scores," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16*), Cambridge, UK, 2016.

LINEAR (LIVE-GENERATED INTERFACE AND NOTATION ENVIRONMENT IN AUGMENTED REALITY)

Giovanni Santini
Hong Kong Baptist University
Kowloon Tong
Hong Kong
info@giovannisantini.com

ABSTRACT

Recent developments in Augmented Reality (AR) technology are opening up new modes of representation and interaction with virtual objects; at the same time, increase in processing power of portable devices is enabling a wide diffusion of applications until recently usable only in very specific situations (like motion-capture labs).

This study aims to describe an AR environment created for musical performance: LINEAR (Live-generated Interface and Notation Environment in Augmented Reality), where the author explored some perspectives made possible by the current state of AR technology applied to music.

In LINEAR, one dedicated performer using an AR iPhone app, can create virtual objects (rendered in real-time and superimposed to the real environment) according to the movement of the device; they are used both as virtual interfaces for electronics (sending OSC message to Max/MSP on a computer) and as forms of live-generated graphic notation. LINEAR allows, with some limitations, the representation of gestural movements with an exact 3-D placement in space: we can now have an analogic notation of gestures, rather than a symbolic one. For the iPhone performer, the act of notation corresponds to the notated act.

The resulting representations can be also approached as graphic animated notation by other performers (the iPhone screen is mirrored to a projector).

The multiple perspectives on the notation and the possibilities of interaction with virtual bodies allow a high level of flexibility, while introducing some almost unprecedented resources and foreseeing a very rich scenario.

1. INTRODUCTION

The idea of LINEAR came from a simple observation: no kind of existing musical notation can really represent a gesture. Even if a specific movement can be described through some kind of symbol or graphic representation, its trajectory can never be fixed in space. The importance of gestural notation in musical scores has been increasing with the overwhelming exploitation of Extended Techniques and the implementation of choreographies inside the compositional process, sometimes even detached from the need for a resulting sound.

The current state of AR technology allows the live generation of virtual entities that can keep track of the trajectory of one gesture in 3-D space. Those bodies can then be linked to arbitrary functions and data, thus constituting virtual interfaces. Furthermore, the trajectories can be interpreted as live-generated graphic notation.

The implementation of these possibilities inside LINEAR is aimed at developing open forms with an open instrumentation (including live signal processing); the real-time generation of the score and the possibility of change in perspective (thanks to AR technology) create a lively compositional ecosystem: the score is not pre-composed by the composer; instead, every performer has the possibility to intervene in real time on the notation, while interpreting it. This way, every player can influence the other ones' behavior. As explained in section 3, this is particularly true for the iPhone performer, who has the highest level of control on the notation.

The project described in this paper was not conceived to explore all the possibilities offered by AR applied to music. It is, instead, a work in progress, where some preliminary ideas are realized, revealing limits both in the still new, fresh and basically unexperimented practice of AR based musical performance [1] and in the technology itself.

2. BACKGROUND

The development of LINEAR is based on a very new evolution in technology and therefore the author could not rely on numerous similar experiences realized before. However, the artistic and technical panorama providing a background for this work is quite vast. In the next paragraphs, the different aspects of such scenario will be introduced.

2.1 Graphic notation on paper

Since the 50s (and in isolated cases even before [2]) musical notation has been pushed beyond a pitch-rhythm representation (as in Common Western Music Notation), in favour of an enormous amount of experimentations, depending on

[1] Visual augmentation of live music performances (as in [1]) cannot be assimilated to the intended outcomes of this project, since that kind of visual augmentation does not have direct consequences on sound and gestural behavior.

[2] E.g., L. Russolo, *Risveglio di una città, 2014*.

different aesthetical purposes, authors, environments and historical periods.

For this reason, a brief categorization of graphic scores or forms of graphic notation can never be really exhaustive or precise. However, we could roughly divide the use of graphic notation in five main categories:

- graphics are linked or linkable to specific parameters (for example durations or dynamics), even if in a "non-conventional" context (as in Cage's *Variation II*, 1961);
- electronic music notation (an example could be Stockhausen's *Studie II* (1954); however, depending on the aim of notational process and kind of composition, the notation could vary enormously);
- graphics are used to obtain some kind of intuitive reaction and forms of free association; they may result from particular re-combinations of traditional staves (as many pieces in Crumb's *Makrokosmos*, or in Bussotti's *AutoTono*, 1978);
- all the possible structures/trajectories of the work are resumed in one map (as in Kourliandsky's cycle of *Maps of non-existent cities*, 2012) or in a rhizomatic [3] representation (as in Haubenstock-Ramati's *Konstellationen*, 1976);
- graphics are used (often even in combination with traditional notation) in order to add indications of specific gestures/actions basically (although not necessarily) aimed at producing sound (as in Lachenmann's *Gran Torso*, 1971 or Laporte's *Dégonflement, 1978* or Yiran Zhao's Dirigentenquartett *Verwickelte Synästhesie)* (2013).

As showed later, the graphic notation generated in LINEAR may be referred to the last two categories.

2.2 Real-time scores and animated notation

As soon as the technology allowed it, the gain of a temporal dimension inside a score (i.e., time is not just codified on the x-axis of the paper, but really "passes" and modifies what the performer sees) became another way to push notation beyond its "traditional" boundaries.

Essentially, real-time scores make use of some forms of animated notation (i.e., graphic animation is implied, producing scrolling, permutation, transformative or generative scores) [3].

According to Freeman, real-time scores can be placed "in the context of algorithmic and computer-assisted composition and also within the aesthetic framework of open-form composition" [4]. Usually, they require a constant sight-reading since the performer cannot wholly foresee the following musical events.

For example, Gerhard Winkler's *KOMA* (1995) makes use of live-generated scores, visualized on a computer, where shapes related to micro-glissandos and dynamics are

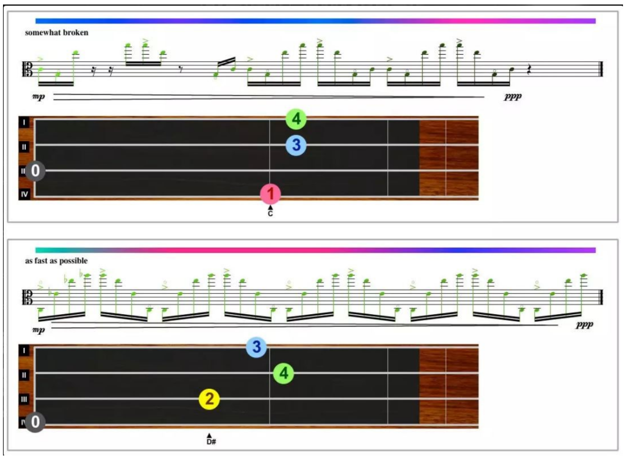


Figure 1. Extract from S. Shafer's *Terraformation*. The color-gradient line on top indicates bow contact position, "fret" notation indicates fingerings and coloured circles left-hand pressure over the strings.

continually moving in real-time, according to principles of real-time generation [5].

In Shafer's *Terraformation* (2017) chords to be performed on a viola are created during the performance (following specific rules set by a decisional algorithm) and translated, in real-time, into an action-based notation comprising three different layers (common notation, "fret" notation and two sets of color-gradient notation, see Figure 1).

The *Decibel Scoreplayer* is a tool for real-time scores, allowing the network-synchronized scrolling of graphic scores ([6], [7]).

In some cases, it is used for regulating the real-time changing transparency of different superimposed images forming a score; such changes in transparency allow the visualization of different trajectories/possibilities inside a rhizomatic score (as in Vickery's composition *trash vortex* [7], [2]).

Some compositions reveal a strong orientation towards an advanced use of graphics and animations, almost transcending the concept of score in favor of the idea of dramaturgy. For instance, in P. Turowski's *Genni* (2018, Figure 2), the score may be seen as the staging of a plot with geometric figures as characters. This piece also shows the expansion of animated notation to the third spatial dimension. In the next paragraph, 3-D notation will be presented more in detail.

2.3 3-D and VR scores

The possibility to access the third dimension in image rendering in real-time, made possible by the increase in processing power of computers and the diffusion of programming frameworks for real-time 3-D rendering (as *Jitter* and *Processing*) can be considered, in the opinion of the author, a real turning point in musical notation.

Kim-Boyle, in [8], presents two compositions using 3-D notation. In *16:16* for piano, in particular, the score is animated and nodes inside 3-D space are mapped to different

[3] For the concept of rhizomatic musical notation, see [2].

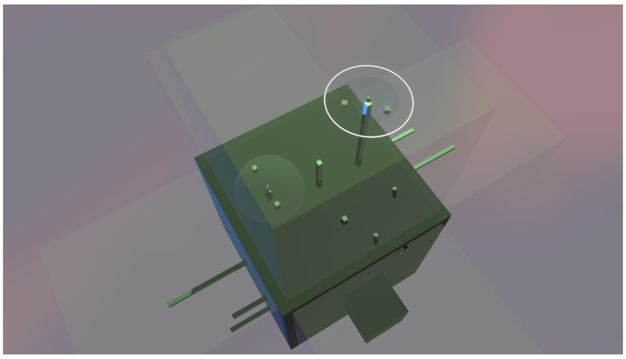


Figure 2. Screen-capture from P. Turowski's *Genni*.

pitches and kinds of piano strings preparation according to position and color. The author also proposes the visualization of the 3-D score by using red-cyan glasses; such an adjustment allows a true perspective vision of 3-D notation for more than one person at a time without the need of VR setups.

Another interesting form of 3-D notation, for playing drums, can be found in [9]: the score allows the representation of different layers of information about the same drum pattern, depending on the point of view on the 3-D structure obtained from pattern analysis; the author also shows 3-D printed scores obtained by those models. Finally, he introduces the use of VR for immersive visualization of 3-D models.

The first ideas about musical visualization in VR can be traced back to 2001 [10], with the proposition of a virtual representation of musical structures derived from form analysis in a VR environment; however, only recently some real experiences have being developed.

In *SpectraScore* [11] (Figure 3), elements visualized in a 3-D VR environment (rendered in real-time) transmit image data to Max/MSP for audio synthesis. Therefore, the sound environment changes depending on visual data extracted from the observed objects.

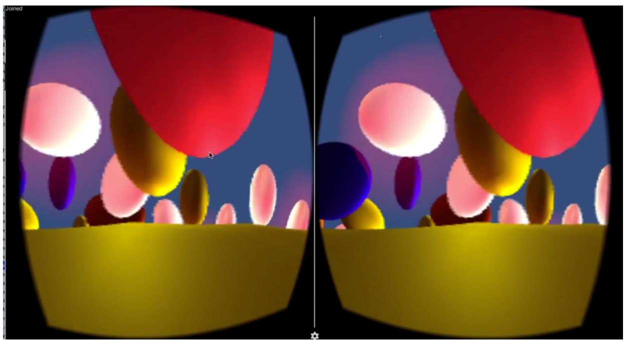


Figure 3. Screen-capture showing the stereoscopic vision for a session with SpectraScore VR [Available: https://www.youtube.com/watch?v=LK6rQFAmPDE].

In A. Brandon's *Hidden Motive* (2018), a graphic score is generated live by the composer (who may also be in another part of the world, sending it through wi-fi) and transmitted to a mobile device mounted on a VR headset (Figure 4). The score is also mirrored to a projector.

However, the use of VR is not necessarily aimed at visualizing 3D scores. In *[P.O.V.]* (2017) for saxophonist, VR glasses, electronics and video mapping by Oscar Escudero Romero, the performer uses VR glasses for visualizing a 2-D scoreplayer and some short animations used as markers for some musical details (like repetitions). The use of VR, in that case, is necessary because of the particular nature of the piece: lights should be turned off in order to deliver good quality projections; this solution lightens up another potential use of VR: scores can be visualized even in the absence of light.

All the experiences above, from graphic to VR scores, extend resources and aims of notation far beyond the Common Western Musical Notation. If a trend can be traced in the presented research progress, it consists in a process towards forms of 4-D representation (notation in the space-time continuum) and interactivity. Last developments in AR technology could constitute the most advanced peak in that direction.

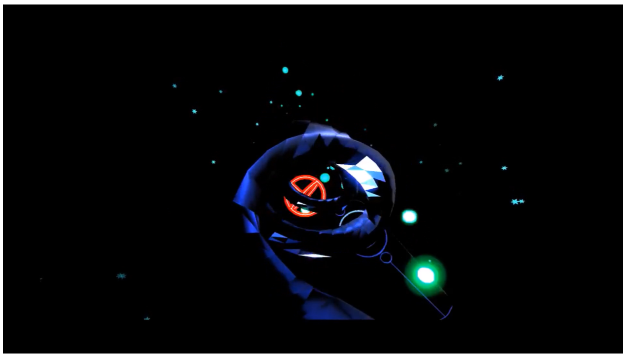


Figure 4. Screen-capture from Amy Bandon's *Hidden Motive*.

2.4 Augmented Reality

Augmented Reality (AR) is a term coined in the 1990s by Tom Caudell [12], for a technology born in the 1960s ([13], in [12]). AR allows the vision of virtual objects (with a precise position in space) superimposed onto the real environment; those "holograms" are made visible through the use of portable devices such as smartphones or tablets (one example is the famous *Pokmon Go!* developed by *Niantic* [4]). In different setups, virtual objects are visualized through screens or projectors connected to a computer (as in [14]). The use of head-mounted see-through devices (such as HoloLens [5]) may be referred to AR, although it is usually inscribed inside the Mixed Reality (MR) [6] field.

Another essential feature of AR consists in the real-time interaction with those items: 3-D virtual objects have a precise position in space and can be looked at from different perspectives. They can be manipulated, with some limits, according to their shape and position in space.

[4] https://www.pokemongo.com
[5] Microsoft HoloLens: https://www.microsoft.com/en-us/hololens
[6] MR is that technology that allows the representation of virtual objects inside the real environment through the use of specifically designed see-through head-mounted devices.

2.5 AR and music

The first experiments in the application of AR to music can be traced back to 2000 with the *Augmented Groove* [15]. The study proposed a marker-based [7] AR sequencer, where different people could insert or remove MIDI tracks by adding or removing cards from a table. Users wearing head-mounted displays (not see-through) could see virtual images rendered on top of the cards. Similar techniques were used by the same authors in 2001 [16], 2003 [17] and in 2007 [18] (implementing also voice and gesture recognition for interaction). A similar experience, yet more evolved and expanded to the simultaneous use of more than one setup (a sort of orchestra of marker-based AR instruments) can be found in [19].

The literature started to grow especially during the last years, following the increased technological possibilities and the continuously spreading interest in the market. Applications developed so far seem constrained, at least in most of the cases, to the imitation of already existing interfaces (as the holographic interface for Behringer DeepMind [8]) or to an aid for improving learning in already existing practices on traditional instruments (e.g., [20] and [21] for guitar, [22] and [23] for piano).

For example, in [21] AR technology was used as a support for studying different songs for guitar: virtual fingers were projected on the frets (visualized on a screen connected to a computer), in order to indicate positions for specific chords. The virtual fingering positions were changed according to the exact timing of the selected song.

Augmented Piano Roll [22] and *Pianolens* [23] (Figure 3) [9] have many similarities. While there are some differences in implementation (use of projectors or HoloLens), the functioning is almost the same. Some colored blocks, whose width corresponds to the keys' one and whose length is proportional to the duration of the note to play, are rolling towards the performer. When they come across a specific line (which is the indicator for "now"), the pianist has to press the corresponding key and keep it pressed until the end of the block.These systems also provide feedback on right and wrong notes and rhythms.

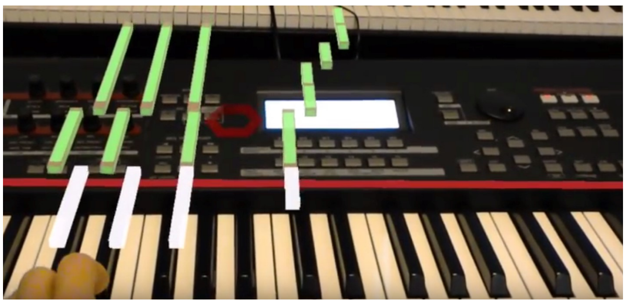


Figure 5. PianoLens demonstration.

[7] Marker-based AR renders virtual objects according to image recognition of specific markers. Each marker is related to one precise virtual objects. Usually, those markers are drawn on cards and the 3-D model is rendered on top of them.

[8] *Behringer DeepMind 12 Augmented Reality Launch*: https://www.youtube.com/watch?v=-9MTlsA-wi4

[9] https://www.youtube.com/watch?v=5TExa2L1rOM

Other studies have been focusing on the exploration of potentialities for real performance.

An interesting use of virtual objects in AR as control interfaces is explored in [14], describing an environment where controllers are visualized in the real world through the use of projectors. Such interfaces can be used thanks to the spatial tracking provided by the use of depth and RGB cameras.

GLASSTRA [24] allows the conductor of a laptop orchestra to visualize in real-time the status of the orchestra on Google Glasses.

An app created by the media artist Zach Liebermann [10] permits the recording of sound while generating a 3-D sound-wave representation. Different visual chunks of the virtual object are linked to correspondent audio chunks in the recorded sound (hence back-and-forth movement along the drawing corresponds to back-and-forth playback of the sound file). The 3-D virtual representation also becomes a 3-D virtual interface for playing back the recording.

The HoloLens AR interface for Behringer's *DeepMind 12* provides AR [11] controllers for the synthesizer to be used with bare hands.

All the experiences above present some form of interactivity and imply mostly real-time information delivery. As shown in the next section, LINEAR permits interaction with virtual objects, while allowing a new form of notation that it is not conceived as an aid for learning a previously existing score but as an autonomous musical representation.

3. LINEAR

3.1 Introduction

LINEAR is an environment designed for new forms of notation and new interfaces in Augmented Reality.

It is composed of the combination of different devices. Its core consists in an AR app for iPhone, developed by the author. It communicates with Max/MSP through OSC (Open Sound Control) connection. The iPhone's screen is mirrored to a streaming box connected to a projector. A dedicated router allows wi-fi connection between the devices.

LINEAR is conceived for the development of open forms in an electroacoustic context with an open instrumentation. An ensemble using LINEAR should include the following figures:

- one iPhone performer (using the AR app);
- one laptop player, controlling some parameters and presets of real-time DSP (Digital Signal Processing);
- at least one instrumentalist playing an acoustic instrument (with live processing).

Before presenting functioning and aims of LINEAR, a preliminary technical introduction is necessary.

[10] *AR app - recording sound in space and playing back by moving through it*: https://www.youtube.com/watch?v=ET2CKUqdPCo

[11] See note 8.

3.2 Technical framework

The software on iPhone is based on ARKit, the framework released by Apple in 2017 for developing Augmented Reality applications on iOs devices.

The AR technology provided by ARKit consists essentially in rendering virtual objects over the rear camera input (thus blending these bodies with the real environment); virtual entities have a precise position in space and, at each video frame (the usual frame-rate is 60 frames per second), they are rendered on the iPhone screen according to the perspective of the observer (more specifically, the device calculates its own position and orientation and therefore derives the observer's perspective). This rendering procedure gives the illusion of a precise positioning of virtual projections in 3-D space. Thus, the device's positional tracking is one of the core features of AR.

According to the Apple Developer Documentation [12] , positional tracking (fundamental for correct rendering of 3-D images) is performed through a Visual Inertial Odometry (VIO) algorithm. It is based on two different data sources: CoreMotion (the Application Programming Interface, or API, that delivers combined data coming from gyroscope, measuring orientation, and accelerometer, measuring acceleration) and the iPhone camera. Feature points are extracted from visual data contained in each video frame captured by the camera; they are compared to the contiguous video frames' feature points for understanding spatial movement. At the final stage, feature point analysis is combined with CoreMotion data to provide a stable [13] (as much as possible) positional tracking.

Rendering can be performed in three different frameworks: SpriteKit (2-D rendering, not suitable for the purposes of LINEAR), SceneKit (3-D rendering), and Metal (3-D custom rendering: the most advanced and efficient but requiring low-level programming).

The AR app for LINEAR is developed in Swift using ARKit and SceneKit. For the purposes of this app, one major advantage of SceneKit over Metal lies in the possibility to instantiate an object by coding only its position in space, its shape and its texture. The framework handles automatically the rendering pipeline and the use of projection matrices for providing a convincing spatial perspective.

The library *SwiftOSC* by Devin Roth [14] is included to handle OSC (Open Sound Control) messages.

3.3 The AR app on iPhone

3.3.1 Startup

On start, the app presents the camera view (i.e., the normal input of the device's rear camera). The screen orientation is locked on landscape mode.

A small green sphere is instantiated 50 cm in front of the camera, marking the center of the point of view. At each frame, the sphere's position is updated according to the device's position and orientation, so that it appears always in the center of the screen (i.e., the center of the camera view).

The app has two main functionalities:

- Creating virtual objects (divided into four categories) linked to stored information (including the name of each object, in order to recall precisely the memorized data);
- Sending to and receiving different sets of messages from Max/MSP via OSC according to specific events.

3.3.2 Creation of virtual objects - first three categories

Virtual objects are divided into four categories, each one with a different particle system [15] attached (linked to different colors: yellow, blue, red and dark violet. Figures 6 - 7).

The creation of objects of the first three categories (yellow, blue and red particle effects) is enabled when the iPhone performer presses the lower part of the screen. The device behaves then like a brush, painting virtual entities in space according to the trajectory during the drawing action. The resulting lines are formed by a succession of small, sphere-like virtual bodies (surrounded by a particle system) aligned along one trajectory.

The body category is chosen according to the speed of the device (depending on specific thresholds).

Every time a new body is created, the software gives it a name and links it to the desired set of information (changes may occur according to different setups and instrumentations for different performances).

When the iPhone performer is not pressing the lower part of the screen, no virtual body is created and the application detects collisions [16] between the green sphere and the painted trajectories.

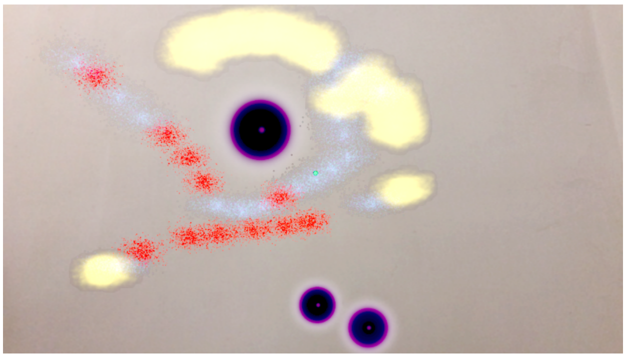


Figure 6. Screen-capture of the iPhone screen running the AR app for LINEAR. One possible graphic result.

[12] Apple ARKit: https://developer.apple.com/arkit/

[13] As better explained in Paragraph 4, the functioning of image data analysis is crucial, since the positional tracking is not stable in case of environments with scarce visual complexity.

[14] devinroth/SwiftOSC: https://github.com/devinroth/SwiftOSC

[15] A particle system (or particle effect) is a graphics effect making use of numerous copies of a small virtual object (particle); each particle can have different movements and behavior. However, the overall impression gives the idea of a single, lively body.

[16] Each object has a "physics body", used for detecting virtual collisions, attached to it. Every time the green sphere marking the point of view collides with one virtual object, the data linked to that object are sent to Max/MSP.

3.3.3 *Creation of virtual objects - fourth category*

The objects of the fourth category (dark violet) are generated according to OSC messages received from a Max/MSP patch running on a laptop. Those OSC messages contain the 3D coordinates of the position of the object to be instantiated; those coordinates are derived from a set of three sound descriptors (e.g., spectral centroid, spectral spread, spectral magnitude) referred to the analysis of the input signal of the Max/MSP patch; the sound produced by one or more instrumentalists participating to the performance is the audio input. The bodies of this category are not generated continuously; their instantiation is triggered by an envelope follower. Additionally, the laptop player can activate/deactivate this functionality.

The iPhone performer can delete every object created in the scene by tapping the highest portion of the screen. All the virtual objects are released from memory, particle systems associated disappear and the data related to the previously created virtual bodies are reinitialized. This function makes it possible to draw new sets of trajectories without preserving the old ones. Such processes are similar to the starting point of a new section in a composition using traditional notation.

3.3.4 *VR mode*

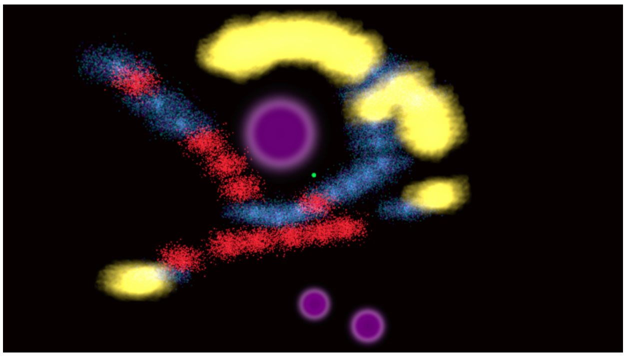


Figure 7. The combination of Figure 6 in VR mode.

The laptop player can trigger a VR mode, excluding the rendering of the camera input and leaving only the virtual bodies. The background can be in any color chosen by the laptop player. This functionality can be used freely throughout the performance. During the VR mode the positional tracking is still functioning, allowing a correct visualization of virtual bodies according to different perspectives.

3.3.5 *OSC communication with Max/MSP*

Information exchanged via OSC is of three kinds:

- data related to virtual objects, sent from the iPhone when bodies are instantiated or whenever a collision between the green sphere and a virtual body is detected;

- speed data sent out at each video frame;

- data related to sound descriptors applied to the input of the Max/MSP patch (acoustic instruments), sent from the laptop to the iPhone.

Figure 8. Rehearsal (F. Teopini iPhone, L. Y. W. Angus flute).

3.4 Production of sound in Max/MSP

Data sent from the iPhone on body collisions are used to play single samples from different libraries (linked respectively to the first three categories). Objects of the fourth category (violet) are linked to the sound they are generated by (the input from acoustic instruments analyzed and sent to the iPhone). Each body, once created, is related to a single sample; therefore, each trajectory drawn by the iPhone performer has a precise sounding identity and can be played in every direction (depending on how the point of view is moved: backward, forward, in small chunks). Objects of the fourth category are discrete points in space (they are tendentially not positioned along trajectories). They break the general continuity of the notation.

The iPhone performer can walk around or across virtual bodies, thus changing the perspective on (and somehow reshaping) the AR interface and the sounding gestures.

The Max/MSP patch also provides live DSP for all the instruments involved in a performance [17] .

Speed data are sent at each video frame from the iPhone to Max/MSP and used to regulate different parameters (such as loudness, DSP presets or parameters' values).

3.5 The laptop player

This performer handles volumes, presets, overall balance and spatialization. Additionally, he/she can also choose what the iPhone speed is going to control.

As explained before, Max/MSP sends messages to the iPhone app, in order to create virtual objects according to sound descriptors. This functionality is triggered by the laptop player and can be interrupted by him/her at any moment.

[17] The Max/MSP patch is modular and allows fast implementation of different techniques according to the needs of different performances.

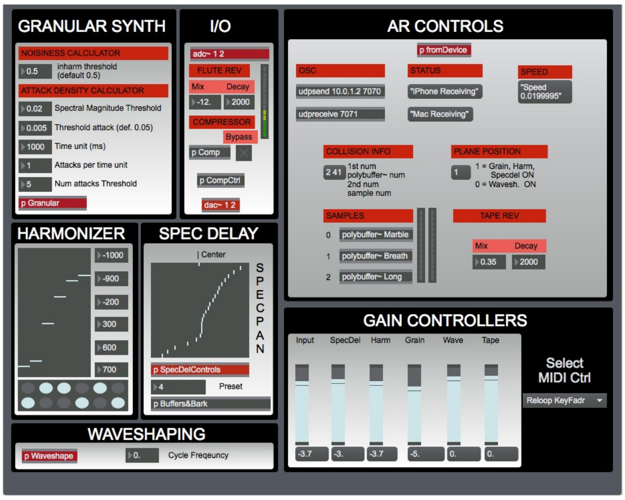


Figure 9. Max/MSP Patch for the laptop player LINEAR (may change for different performances and setups).

Figure 9 shows the UI (User Interface) used by the laptop player for a performance including electric guitar and cello. Techniques and layout may vary depending on the context.

3.6 The perspective of the iPhone performer: graphic gestural 3D notation and virtual tangible scores

The iPhone performer creates the highest number of virtual sounding bodies during a performance, each with a precise placement in space and each linked to a precise sound sample. Consequently, a specific result derives from a specific movement, and that movement is represented by a specific trajectory drawn thanks to the created virtual bodies.

The notation indicates a precise gestural behavior for the iPhone performer: it describes what gesture he/she has to perform in order to obtain a specific result. However, the notation is conceived to leave some decisional freedom to the interpreter, as it does not indicate how fast or how continuously the trajectory should be followed. Furthermore, the performer's movements are not necessarily constrained to the painted lines.

3-D drawings are, at the same time, a control interface for sample libraries. The "physical" interaction between the green sphere marking the center of the point of view and the other virtual bodies generates sounds (through the Max/
MSP patch). In short, from the perspective of the iPhone performer, virtual bodies have two different functions: they bring information about movements for generating sound and they are the "generators" of that sound.

Such a co-presence of notation and sound generator/ control interface in the same virtual objects, induces us to consider the existence of a new typology of scores which could be called (quite oxymoronically) *virtual tangible scores*, as a particular case of tangible scores (defined as "graphical scores [. . .] physically incorporated in the form of the instrument" [25]).

3.7 The perspective of the other players

3.7.1 Graphic animated notation

The other players (laptop performer and instrumentalists) cannot interpret the drawings the same way as the iPhone performer does. They cannot interact directly with virtual objects.

For them, those trajectories are part of a real-time animated [18] score that does not have immediate gestural implications . The score they read is intended as a means to convey creative energies during the performance. As Fischer writes:

> "An animated notation is an invitation for composers and performers to start their own so-called mapping process. They need to connect or map visual attributes with sonic attributes. In staff notation the mapping by composer and performer are basically congruent. In animated notation the mapping process is done individually, first by the composer and then by the performer." [26, p. 35]

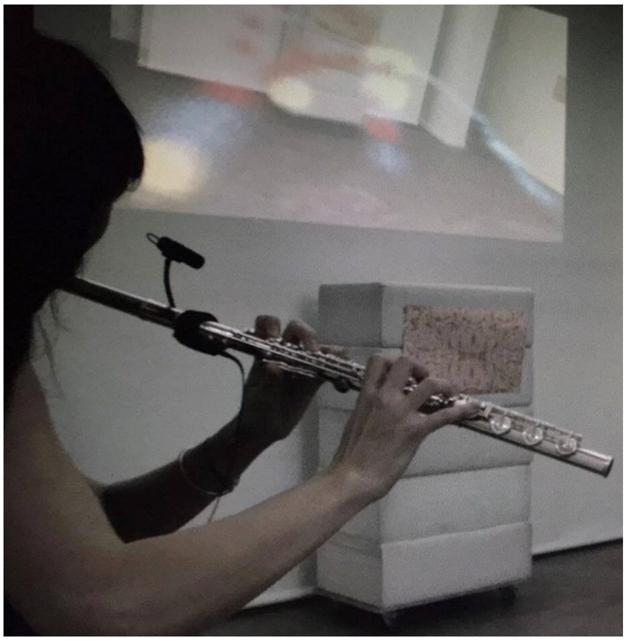


Figure 10. P. Pakiela reading the graphic notation from the projected image.

Such a continuous re-mapping process implies a particular form of creativity that other types of notation intrinsically exclude (e.g., Common Western Notation). The score is not written once and fixed. This kind of notation, derived from graphic scores developed since the 1950s, redefines the idea of composition in terms of continuous creative effort exploited by a group of performers; in LINEAR, in particular, the group operates exclusively in an *in-time* dimension, excluding any *out-of-time* structural planning [19] .

[18] The animation derives from the continuous movements of the iPhone performer, who is constantly changing the perspective on the AR shapes.
[19] Though with a strong simplification, the idea of the in-time/out-of-time dichotomy is derived from Xenakis [27].

Codification strategies applied so far by the author of this article require the identification of different body categories (particle system colors) with generic sound qualities (e.g., yellow = moaning sounds, red = inharmonic-distorted sounds, blue = lively high-pitched sounds, violet = static-low sounds). Instrumental techniques used for the realization of the score are discussed before rehearsal or performance, but there is no pre-decided path to follow. Performers are asked to "read" a single trajectory until they reach a point of conjunction with other trajectories. At that point, they can jump to another tone.

The definition of more refined strategies for the interpretation of the animated score is an open process: the interaction between the iPhone player and the score creates a high number of unpredictable situations, hard to codify in advance. However, some guidelines seem to emerge. The possibility of a change in perspective performed by the iPhone player is probably the most powerful resource: getting closer to (thus zooming-in on) a specific point inside a trajectory has wide repercussions. For instance, getting extremely close to one virtual body would fill the entire screen with the color and particle effect of that body, thus creating a sense of totality of the sound quality related to it. It is also instinctively translated into a *ff* (at least, according to the performers that played in LINEAR). On the contrary, finding a point of view that excludes almost every virtual object, except for some small, far bodies, could be interpreted as a perforated and quiet sound texture.

3.7.2 Notational feedback

The instrumentalists and the laptop player can partially modify the score.

The laptop player can trigger the VR mode: when it is active, the real environment is not rendered and is substituted by a plain color background. In this case, performers can concentrate only on the virtual score. According to the performers the author worked with, this functionality somehow changes the re-mapping process and, in general, makes it easier for the players to concentrate only on the score reading. However, the AR mode is considered the main one. Motivations for this are presented in 3.8. Another functionality is triggered by the laptop player: the creation of virtual bodies of the fourth category.

When instrumentalists play (interpreting the notation on screen), they generate notation (and virtual interfaces), as virtual bodies are instantiated according to sound analysis. Such a process produces a phenomenon that could be called *notational feedback*: the notation is created as an effect of its interpretation. The concept of notation can be pushed to some unknown boundaries, where the ideas of authorship, composer, form and improvisation can be seen under a new, slightly different light.

3.8 Compositional ecosystem

Going back to the definition of animated score by Fischer, we can suggest that, in LINEAR, AR scores go somehow a step beyond. The process of mapping is not done "first by the composer and then by the performer". It is rather reconstructed in real-time by the whole ecosystem formed by all the performers. The role of the "composer" is limited to the predisposition of the conditions for the ecosystem to be formed (software development, proposition of strategies). Beyond that, the notation and the details of formal development are completely in the hand of the real-time performing ecosystem.

There is an internal hierarchy, with the iPhone player on top (considering the privileged relationship with the score). To some extent, an AR drawing may be assimilated to a formal section (or to the whole piece) in a commonly notated composition: the "main idea" is the entire virtual painting, and the development lies in the different perspectives one can obtain (zoom-in, zoom-out, rotation, exclusion from the field of view, etc.).

3.9 Relation of the score with the audience and with the environment.

The essential feature of AR consists in blending the real environment with digitally rendered objects. The presence of an object in the real space clarifies its spatial existence and dimension (this is especially important if we recall the idea of virtual tangible scores presented in 3.5). The interaction with the real environment brings the score itself inside the space of the performance. The score has a 3-D inclusive nature (it can potentially include the entire venue of the performance). The audience is, in some way, part of the whole process of creation and can be surrounded by those virtual objects.

Figure 11. A performance (Liverpool, FACT 3).

Essentially, the projected score also acts as a visual part of a multimedia performance. While the VR mode presents only the score itself, AR also provides a perspective on venue and spectators. For this reason, the AR mode is considered the main one.

4. ISSUES AND LIMITATIONS

In its current form, LINEAR shows some limitations, ranging from the still preliminary stage of artistic development to the imperfections in positional tracking.

Regarding the latter issue, as explained in 3.2, ARKit makes use of feature points for performing its VIO algorithm. The absence of visual cues in the image detected by the camera will result in poor positional tracking. In worst cases, virtual objects move randomly around the scene.

Main causes for poor tracking are recognized as being bad light conditions and reduced visual complexity in the scene (therefore, lack of feature points) [20] .

Fast movements and sudden changes in the camera view easily lead to tracking issues (feature points must be compared between consecutive frames), reducing the iPhone performer's freedom of movement; at the current stage of development, the use of AR trajectories as choreographic indications, though promising, is not completely viable and presents risks for the stability of positional tracking.

Another problem is distance estimation. As [28] shows, users tend to underestimate distances, with obvious limitations to the flexibility and precision of interactions with virtual entities. Among many technical solutions, only shadows projected on the floor have a positive impact on distance estimation. At the current stage, downcast shadows are not implemented in LINEAR.

Even if the camera input is rendered on the screen, the device does not understand how the surrounding space is shaped, i.e., it does not understand depth data in the image. Therefore, the interaction of virtual and real world is still limited. For instance, virtual entities positioned behind a real object would not be hidden, as it would happen in reality (phenomenon called occlusion).

The possible notational solutions are currently constrained to only four body categories, each one emitting a particle effect. Even if an infinite number of different trajectories can be created, the look of a single virtual body or of a body category cannot change over time. The use of a VR mode makes the notational process more dynamical but does not overcome all the limitations.

There is also an intrinsic (and wanted) constraint: the system is not meant to create fixed, pre-composed AR scores. The author is currently working on another project aimed at filling this gap.

5. CONCLUSIONS AND FUTURE WORK

The current state of AR technology permits the exploration of unprecedented possibilities in musical notation and performance. While the technology itself existed for 50 years, only recently it has reached a level of flexibility and precision allowing a relative ease of implementation.

In this study, the author has presented a possible use of AR in LINEAR, where the OSC connection between an iPhone app, a Max/MSP patch and a streaming box produces an environment usable for performances based on live-generated animated scores and virtual interfaces. Its use sheds light on some concepts that have not been fully explored yet:

- virtual tangible scores (the iPhone performer plays virtual trajectories, i.e. the notation itself);
- notational feedback (some virtual bodies are created according to the analysis of the acoustic instruments' sounds; i.e., the notation is created by itself, as an effect of its reading);
- compositional ecosystem (all the performers have a direct influence on the notation and how it is interpreted).

In future works, notational process and performance strategies can undergo considerable improvements, especially with the design of a more complete and complex set of possibilities. Enhancements would include techniques of image processing for the camera input, such as distortion, frame differencing, tessellation as well as dynamic change of the visual features of virtual body categories throughout one performance.

The integration with different sensors could further expand the application functionalities. Major improvements in world tracking could be accomplished using 3-D ambient scanning sensors (as the *Structure Sensor* [21]). This implementation would allow a higher freedom in movement for the iPhone player (better positional tracking) and a better quality in the interaction between virtual and physical world. Introducing the use of a headset for mobile devices (*Bridge* [22] by *Occipital* for Apple devices) could also bring to a higher level of immersion for the iPhone performer and to a different approach with notation and perspective changes.

The idea of virtual tangible scores suggests the use of haptic devices in order to give the feeling of touch with virtual structures.

The continuous contact with performers (aimed at identifying limitations and at improving artistic and technical aspects of the system) is, and will be, an essential part of this research.

Acknowledgments

My PhD studies at Baptist University are funded by the HKBU UGC grant.

6. REFERENCES

[1] ARMusic, “Augmented Reality During Live Music Performances.” [Online]. Available: https://www.youtube.com/watch?v=nyVs_5TfN4c

[2] L. Vickery, “Rhizomatic approaches to screen-based music notation,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Brisbane, Australia, 2016, pp. 394–400.

[3] ——, “The evolution of notational innovations from the mobile score to the screen score,” *Organised Sound*, vol. 17, no. 2, pp. 128–136, 2012.

[4] J. Freeman, “Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance,” *Computer music journal*, vol. 32, no. 3, pp. 25–41, 2008.

[20] Apple – *Introducing ARKit: Augmented Reality for iOS – WWDC 2017*: https://developer.apple.com/videos/play/wwdc2017/602

[21] Occipital, *Structure Sensor - 3D scanning, augmented reality, and more for mobile devices*: https://structure.io

[22] *Bridge - Mixed Reality and Positional Tracking VR Headset for iPhone and iOS*: https://bridge.occipital.com/

[5] G. Winkler, “The realtime-score. A issing-link in computer-music performance,” in *Proceedings of the Sound and Music Computing conference (SMC’04)*, Paris, France, 2004.

[6] C. Hope, A. Wyatt, and L. Vickery, “The Decibel ScorePlayer: New Developments and Improved Functionality,” in *Proceedings of the International Computer Music Conference (ICMC)*, Denton, USA, 2015.

[7] C. Hope, L. Vickery, A. Wyatt, and S. James, “The Decibel Scoreplayer - A Digital Tool for Reading Graphic Notation,” in *Proceedings of the International conference on Technologies for Music Notation and Representation (TENOR’15)*, Paris, France, 2015.

[8] D. Kim-Boyle, “The 3-D score,” in *Proceedings of the International conference on Technologies for Music Notation and Representation (TENOR’17)*, A Coruña, Spain, 2017.

[9] J. J. Ham, “An architectural approach to 3D spatial drum notation,” in *Proceedings of the International conference on Technologies for Music Notation and Representation (TENOR’17)*, A Coruña, Spain, 2017.

[10] D. Hmeljak, “3D Musical Notation - Providing Multiple Visual Cues For Musical Analysis,” in *Proceedings of the XII Colloquium on Musical Informatics*, Gorizia, Italy, 1998.

[11] B. E. Carey, “SpectraScore VR: Networkable virtual reality software tools for real-time composition and performance,” in *International conference on New Interfaces for Musical Expression (NIME)*, Brisbane, Australia, 2016.

[12] H. Ling, “Augmented Reality in Reality,” *IEEE MultiMedia*, vol. 24, no. 3, pp. 10–15, 2017.

[13] I. E. Sutherland, “A Head-Mounted, Three-Dimensional Display,” in *AFIPS’68 Proceedings of the Fall Joint Computer Conference, Part 1*, 1968, pp. 757–764.

[14] F. Berthaut, C. Arslan, and L. Grisoni, “Revgest: Augmenting Gestural Musical Instruments with Revealed Virtual Objects,” in *17th International Conference on New Interfaces for Musical Expression (NIME)*, Copenhagen, Denmark, 2017, pp. 180–185.

[15] I. Poupyrev, R. Berry, J. Kurumisawa, K. Nakao, M. Billinghurst, C. Airola, H. Kato, T. Yonezawa, and L. Baldwin, “Augmented Groove: Collaborative Jamming in Augmented Reality,” in *ACM SIGGRAPH 2000 Conference Abstracts and Applications*, New Orleans, USA, 2000.

[16] I. Poupyrev., R. Berry, M. Billinghurst, H. Kato, K. Nakao, L. Baldwin, and J. Kurumisawa, “Augmented reality interface for electronic music performance,” in *Proceedings of the International Conference on Human-Computer Interaction (HCI)*, New Orleans, USA, 2001, pp. 805–808.

[17] R. Berry, M. Makino, N. Hikawa, and M. Suzuki, “The Augmented Composer Project: The Music Table,” in *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR’03)*, Tokyo, Japan, 2003, pp. 338–339.

[18] F. Farbiz, Y. T. Ka, K. Wang, W. Ahmad, C. Manders, J. H. Chong, and K. T. Yeow, “A multimodal augmented reality DJ music system,” in *International Conference on Information, Communications and Signal Processing*, Singapore, 2007.

[19] T. Laurenzo, “YARMI - An Augmented Reality Musical Instrument,” in *International conference on New Interfaces for Musical Expression (NIME)*, Pittsburgh, PA, USA, 2009, pp. 268–269.

[20] F. Liarokapis, “Augmented Reality Scenarios for Guitar Learning,” in *EG UK Theory and Practice of Computer Graphics*, L. M. Lever and M. McDerby, Eds. The Eurographics Association, 2005.

[21] C. Kerdvibulvech and H. Saito, “Vision-Based Guitarist Fingering Tracking Using a Bayesian Classifier and Particle Filters,” in *Advances in Image and Video Technology – Second Pacific Rim Symposium (PSIVT)*, Santiago, Chile, 2007, pp. 625–638.

[22] V. Chouvatut and W. Jindaluang, “Virtual piano with real-time interaction using automatic marker detection,” in *International Computer Science and Engineering Conference (ICSEC)*, Nakorn Pathom, Thailand, 2013, pp. 222–226.

[23] E. Strasnick, A. Chambers, L. Jiang, and T. Xiaonan, “Pianolens: An Augmented Reality Interface For Piano Instruction,” 2017. [Online]. Available: https://cs.stanford.edu/people/estrasni/research/pianolens/

[24] I. Bukvic and S. Lee, “Glasstra: Exploring the Use of an Inconspicuous Head Mounted Display in a Live Technology-Mediated Music Performance,” in *International conference on New Interfaces for Musical Expression (NIME)*, Copenhagen, Denmark, 2017, pp. 313–318.

[25] E. Tomás and M. Kaltenbrunner, “Tangible Scores: Shaping the Inherent Instrument Score,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, London, UK, 2014, pp. 609–614.

[26] C. M. Fischer, “Understanding Animated Notation,” in *International conference on Technologies for Music Notation and Representation (TENOR’15)*, Paris, France, 2015.

[27] I. Xenakis, *Formalized Music. Thought and Mathematics in Composition*. Pendragon Press, 1992.

[28] C. Diaz, M. Walker, D. A. Szafir, and D. Szafir, “Designing for Depth Perceptions in Augmented Reality,” in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Nantes, France, 2017, pp. 111–122.

JOHN, THE SEMI-CONDUCTOR: A TOOL FOR COMPROVISATION

Vincent Goudard
Sorbonne Université , Collegium Musicæ
Paris, France
goudard@lam.jussieu.fr

ABSTRACT

This article presents "John", an open-source software designed to help collective free improvisation. It provides generated screen-scores running on distributed, reactive web-browsers. The musicians can then concurrently edit the scores in their own browser. John is used by ONE, a septet playing improvised electro-acoustic music with digital musical instruments (DMI). One of the original features of John is that its design takes care of leaving the musician's attention as free as possible.

Firstly, a quick review of the context of screen-based scores will help situate this research in the history of contemporary music notation. Then I will trace back how improvisation sessions led to John's particular "notational perspective". A brief description of the software will precede a discussion about the various aspects guiding its design.

1. INTRODUCTION

1.1 From traditional to graphical score

The score is generally considered as a tool for the composer to create a musical work for an interpreter. It describes the expected sonic result and prescribes the gestures to perform[1]. It thus stands as mnemonic mean to keep track of what is independent from the context of the performance[2] and often, is assimilated to the artwork itself in Western musical tradition.

Scores fulfill yet many other functions. It allows in particular to transpose the musical time into a visual space, enabling the composer to arrange musical elements "out of time" in order to produce pieces that could not be conceived without this visual support[3].

If the Western notational system invented by Guido d'Arezzo in the 11th century has continuously evolved, improving with new symbols and techniques until the early 20th century, the technological and cultural revolutions that followed subverted both the means of production and the range of musical expression, now extended to noise and the whole sound spectrum.

We can notice the development of so-called "graphic scores"[4] in the middle of the 20th century, that reflects this musical evolution for which the traditional notation is insufficient. For reasons that might seem opposite, the graphic score helped to push both the limits of what was possible to "fix" in a composition — by specifying it entirely on a synthesis system, and the limits of what it was conceivable to vary — the part entrusted to the performer's interpretation. The scores for Iannis Xenakis' *Mycene Alpha* and Earle Brown's *December 1952* highlight both these directions (see Figure 1).

This apparent opposition between a totally fixed work and a work that is totally subject to the performers' creativity seems more like the outcome of complementary approaches that aimed at exploring the new sound and musical domains, in their manifestations as well as in their potentialities, whether reified or fantasized.

Within this continuum of possibilities between fixed work and free improvisation, that Sandeep Bhagwati called "comprovisation" in [1], various notational perspectives can be considered. The various purposes of musical representation hitherto integrated in the traditional score gain independence and take a variable importance, adapting to the musical work and the performance contexts. The score defines the playing field, which is not necessarily linear and which, thanks to the possibility of producing animated images in real time, is no longer necessarily fixed.

1.2 Screen Scores

The increasing availability of digital devices led to the development of several applications meant for the creation of scores on-screen. As Lindsay Vickery notes in [20]:

> *These developments suggest a trend, particularly amongst young composers whose practice has developed exclusively on computer, to take the logical step to present notated materials on screen.*

[1] Eric Maestri proposes the terms "phonographic" and "ergographic" to describe both these aspects [14].
[2] Acknowledging here that the interpretation belongs to the contextual.
[3] A notorious example is the rondeau "*Ma fin est mon commencement*" (14th century) from Machaut in which the two voices are each other's retrograde.
[4] … that is, using graphic signs other than the usual symbols of the conventional notation of notes on a staff.

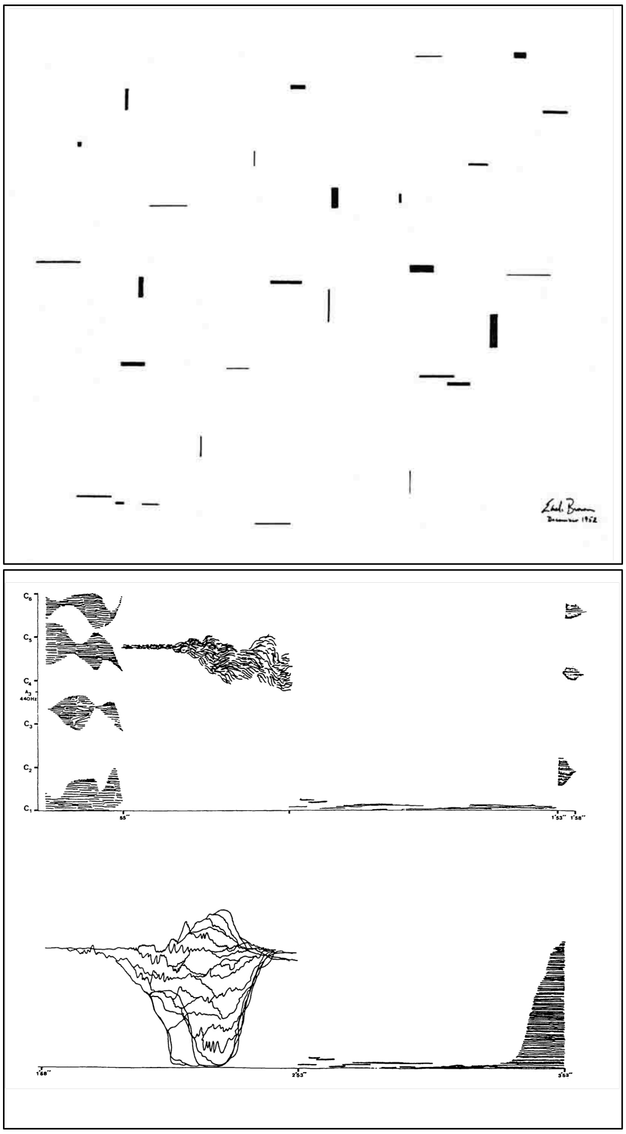


Figure 1. Extracts from *December 1952* by E. Brown (top) and *Mycène Alpha* by I. Xenakis (bottom).

Cat Hope summarizes the main features offered by this new medium with the following terms in [10]: *scrolling, permutative, transformative, generative and networking capabilities of the digital medium.*

Using computer graphics for the purposes of musical representation seems a medium of choice to enrich the possibilities of writing graphic scores. Especially, the fluidity of adaptation of the virtual medium makes it possible to envisage multiple "views" of the same score according to the contexts for which it is intended. Thus the composition, the performance, or the analysis of a musical work do not necessarily require the same representations. In terms of musical performance, we can add a distinction between the interpretation of a score by a human and a machine, these two types of "interpreters" affording relatively different abilities.

In the same way that digital technologies have atomized the musical instrument by decoupling its various constituents (gestural controller, mapping, synthesis, etc. becoming modular), they have also atomized the score into its various functions, supporting composition, performance or analysis. It is then necessary to specify which use case is at play and Cat Hope defines for this purpose the term "screen-score" as the medium presented to the musicians for a performance in [11]:

> *Screen-scores are notated music compositions devised to be performed; and are not to be confused with visual representations of music or the musical interpretation of visual art.*

The concept of screen-scores has been investigated in depth by several authors, composers and musicologists, (in papers by Winkler [18], Clay [3] or Lee [12]) who discussed the advantages and drawbacks of using digital technologies for musical representation, both in its technical aspects and in its musicological consequences. Lindsay Vickery offers a very detailed review in [17] of critical latencies allowing an instrumentalist to read musical material displayed in real time and provides advices on what the composer should pay attention to when composing with this medium.

These studies offer relevant and valuable descriptions to the composer who wishes to work with screen-scores. However, it seems that they can be supplemented by a different approach to the score than those considered in most of the literature, in which the point of view is often that of the composer. The design of a screen-score system is consequently polarized by the central importance of the score, itself considered as a prerequisite for musical performance, a situation that also reflects a strong tradition of Western classical music[5].

In the case of ONE's performances (Figure 2), which are based on a practice of free improvisation devoid of prior composition, the focus moves towards the instrumentalist's side. The central element is not the score but the listening and understanding of sound and other musicians. The score (if it is still possible to call it so) often emerges after the improvisation sessions and its presence should not be at the expense of mutual attention. From this perspective, it is possible to envisage that the musician him/herself adapts the musical representation to his/her own needs, depending on the parts s/he has to play, her/his personal preferences, the various movements of the score, etc.

In the particular case where the instruments are digital and programmable, the use of a networked score system finally offers the possibility of delegating certain parameters of the instrument to an outsourced control supported by the score. In a situation of improvisation, the negotiation between this automated control and the choice of the musician implies a mediation that I will discuss later.

[5] A notable exception is the contribution of Georg Hajdu [9] who proposes the concept of "disposable music" to qualify musical forms "*that rely on a lesser degree on fully notated scores, such as "comprovisation" or laptop performance"*. However, even as "*disposable*" as it is, the score plays here again a prior role to the performance and remains central to the attention, differing from our approach.

Figure 2. The members of ONE with their digital musical instruments.

1.3 Improvisation in ONE – birth of a notation

The seven musicians of ONE[6] are all deeply involved in the field of computer music with varying specialities applying in the fields of instrumental practice, composition, instrument making, research in music sciences and education. All of us practice digital musical instruments of which we built the software parts and sometimes also the hardware parts, to some extent. At the origin of our collaboration, there was no other project than that of attempting the experiment of playing a sound-based music together with such heteroclite digital instruments, without grid, without music theory, without prior agreement on the form and content.

Several improvisation sessions were opportunities to discover our sounds, our playing styles, our musical vocabulary. These moments of rehearsal were first and foremost the occasion of anarchic performances, guided only by the thread of our listening, to confront, to merge, to burst, to collide spaces, objects, soundtracks, along with moments of discussion and adjustments of our musical setup.

These sessions were also subject to classic improvisation exercises: searching for timbral fusion and counterpoints, fugal passages among musicians, accompanying a soloist, working on the pianissimo nuances, or playing "in the style" of a piece we knew. Eventually, audio recordings allowed us to play back the sometimes long and uninterrupted improvisations to extract interesting ideas.

The issue of large musical movements appeared before ONE's first public concert. The lack of a score structuring the concert's duration led us to follow a narrative scenario inspired by a novel by Jules Verne. Thus, the concert consisted of a series of chapters, simply identified by intertitles in lieu of exotic and imaginary soundscapes to be explored.

Little by little, these experiences gave rise to the emergence of a more atomic musical vocabulary representing atmospheres and movements collectively defined, that we named "*karmas*"[7]. The various moments of play and discussion brought us to the development of other conceptual objects that were partly implemented in the form of a software nicknamed "John, the semi-conductor".

The origin of John's development is related to the desire to find a way to structure musical time in different movements within the perspective of freely improvised concerts of fairly long duration. Another motivation lies in the ability to vary the improvisations so as not to always repeat the same textures and formal structures such as sequences of ascending-descending cycles.

In addition, we were looking for ways to stimulate the exploration of unusual combinations and musical ideas pushing us out of our "comfort zone". The proposal to mathematically divide time into sequences to allow all possible ensembles of solo, duet, trio, up to the tutti, was the first impetus for the development of a score generator able to automatically produce such distributions.

As the opinions diverged within the band on the balance between rules and absence of rules, a key principle did find a ground of agreement: John is a semi-conductor. This means that scores created using John are just a proposition that each member of the group is free to follow or not, depending on the musical context that only takes shape during the very moment of the performance. Listening remains the primary rule of the game, prevailing over a blind follow-up of the score. In particular, the articulation between the different parts of the score, whether they are tiled or disjointed, or the act of playing when not supposed to, etc. is left to the appreciation of each musician.

This principle has the direct consequence of a streamlined design whose purpose is to allow each player to situate oneself within the score at a glance, without monopolizing her/his attention to the detriment of the other musicians. The goal is therefore very different from the one pursued in other screen-based musical notation systems such as those explored in works involving (extreme) sight-reading [8].

Essentially, John allows collective time management, whether during rehearsals, composition, or performances, providing a shared representation support. A brief description of the software to capture its outline will precede a discussion of the different aspects related to this group management.

[6] Performance except: https://youtu.be/lBVNwGeTxFA

[7] The relationship with this Indian concept is distant, but it does include an appealing meaning that echoes how we view them in performance: the set of actions represented by the karma influences the future of the individual. In the same way the musical interpretation of a *karma* (as we define it) is subject to the actions of the musicians and any accident, bifurcation with respect to the score will prevail on the musical evolution more than the score itself.

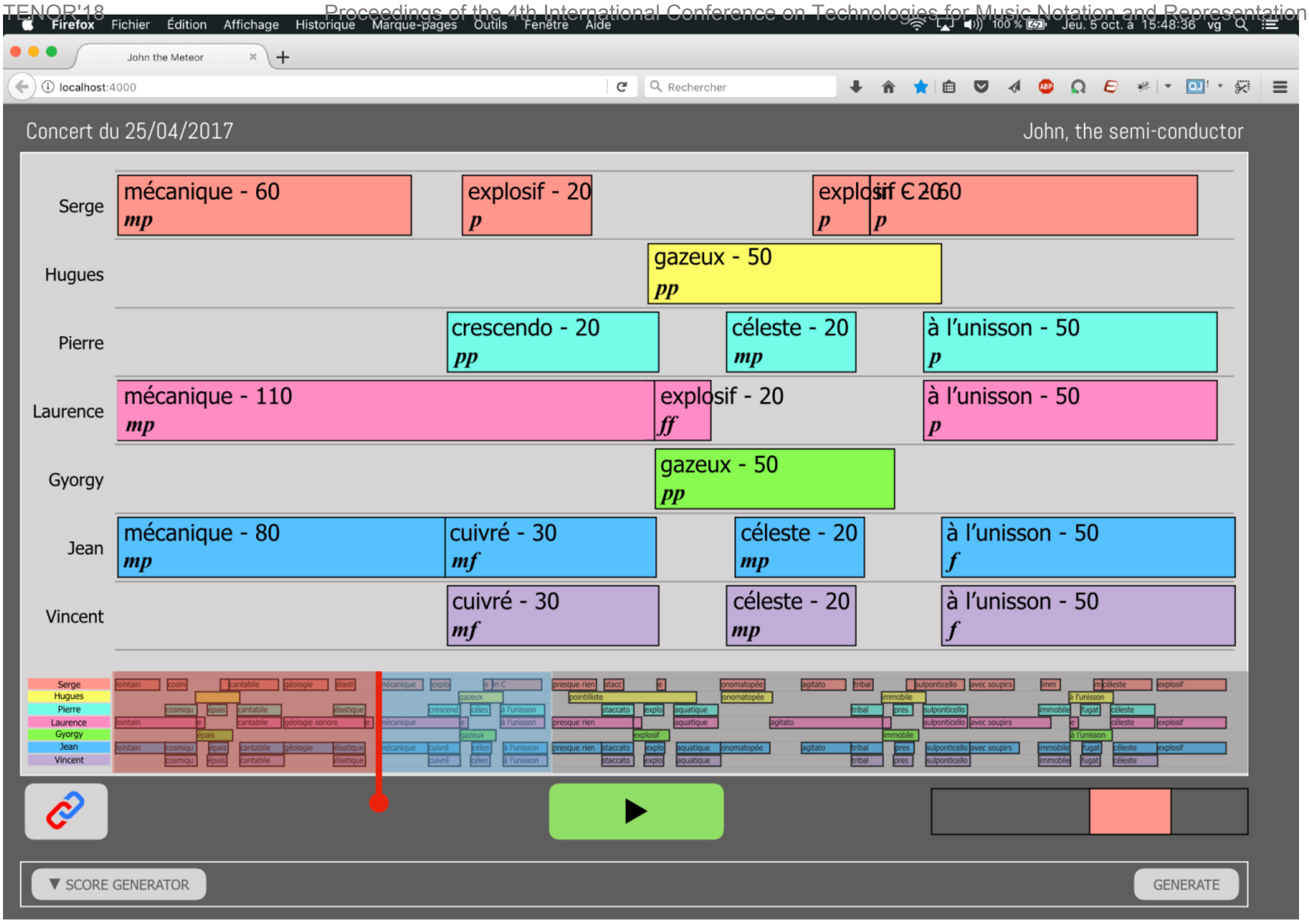


Figure 3. Snapshot of John's client interface.

2. ABOUT JOHN

John relies on a client / server architecture, in which each musician is visualizing a client interface in a web browser, on which s/he can act. This interface comes in two parts, a score generator on the one hand and an interactive visualization of the score on the other hand.

2.1 Score generator

The score generator makes it possible to very quickly create musical propositions by specifying only global constraints:

- overall duration of the score;
- minimum and maximum number of players;
- minimum and maximum duration of blocks;
- a list of *karmas* that identify a particular mood according to a common vocabulary established by the musicians during improvised practice sessions;
- nuances from pianississimo to fortississimo.

Once these constraints are specified, the score generator produces a random proposition that respects these conditions, that is composed of a sequence of time blocks associated with a *karma* and a nuance. This proposal can then be adjusted in the editing / viewing interface.

2.2 Interactive visualization

This interface represents blocks on a timeline. It consists of a reduced *global view* on one hand, giving a shared overview of the whole score, and a zoomed *local view*, located above the global view. On the global view is a playhead that is common and synchronous to all clients (in red on Figure 3), as well as a window (in blue on Figure 3) defining the time span displayed in the local view. This window is defined individually by each musician on their client and is typically ranging from ten seconds to a few minutes depending on the temporal granularity of the score and the preference of each.

All controls are accessible in all clients, so that anyone can edit the score: generating a new score, moving and changing the duration of the blocks and their content (*karma* and nuance), starting playback, changing the playback speed, moving the playhead to start at a given moment of the score. These changes will be immediately reported to all other John's clients.

The user can also define local parameters which will only affect her/his own client interface: the various tracks visibility, the duration of one's *local view* and the synchronization (or not) of one's local view to the reading cursor with the *link* toggle-button.

2.3 Implementation details

After a first version developed with Max[8], the application was brought to reactive HTML5 using the Meteor framework[9]. This allows collective editing on any platforms (including mobile platforms) connected to a LAN, via a simple web browser. The visualization was implemented using the D3.js library[10].

Scores are saved in JSON format as a list of events with a unique identifier, a track index, a start time, a duration, and a number of properties such as karma and nuance. During playback, time and score events are sent as MP messages [9] over the network using OSC[11].

3. JOHN IN PRACTICE

3.1 Generative composition

The score generator saved a lot of time during the rehearsals, giving us an immediate possible musical structure like an empty shell. As arbitrary as it is, its main function is to stimulate the musical performance with the most minimal prescription: play (or don't). Thus the proposals are often tried as they come before being adjusted collectively according to what members of the band find interesting or not. We can then evolve this musical structure, with apparently more efficiency than if we were to start out from nothing.

3.2 Distributing participation

The fact that John explicitly proposes a distribution of playing time among each musician has led to situations of performance that we would not necessarily have tried, especially in reduced configurations (solo and duet), each of us having a tendency to play too readily to actually leave room for these minimal configurations to settle.

Moreover, having “out of the game” moments makes it possible to better anticipate one's appearances. Indeed, digital musical instruments often have a “meta” dimension[12] and more generally a huge number of settings. These planned moments of rest make it possible to better manage the time we have to reach less accessible settings.

3.3 Tight synchronisation

At a micro-temporal level, synchronisation is impeded by the lack of idiomatic rules[13]. In particular, the absence of pulse or metric system makes the synchronization among musicians ever more difficult as their number increases and often deprives freely improvised musical forms from clear and tight transitions in large ensembles.

The conductor, when there is one, provides accurate cues, beats, and potential directions for play. Beside the ethical issues raised by the role of a leader in an improvisation band, raised by Canonne in [2], entrusting the conducting to a person[14] remains limited by the fact that s/he can only act in the present, and that it requires the almost permanent attention of the musicians, to the detriment of the attention they could bring to their peers. In this respect, the representation offered by John condenses in a certain way the score and the conductor in a single visual medium. The animated score (“scrolling score” in our case) offers indeed visual cues that indicate the simultaneity of several musical events, and its scrolling under the playhead allows a precise synchronization among the musicians at transitions.

3.4 Visual support for musical landmarks

Despite the availability of analysis tools[15] and a certain lexicon to describe sound and musical objects in electroacoustic music[16], there is no standard of prescriptive notation for digital musical instruments. The lack of a unanimous vocabulary, the singular nature of the instruments and the tremendous sound palette they provide can make it a nightmare to identify and discuss what has just been played during a long improvisation session (somehow failing here to use the word “rehearsal”). A minimal score such as that proposed by John facilitates this identification and allows to re-work specific moments after a long performance. The reduction that symbolic notation carries out on the complex sonic outcome of a performance allows everyone to quickly find one's way in the temporal space of an improvisation, faster than it would if we had to refer to the sound recording.

3.5 An ecology of attention

Free electroacoustic improvisation involves strong musicians' attention to other musicians, to their instrument and, obviously, to sound. In this respect, digital instruments often present the additional drawback, as compared to acoustic instruments, of capturing some of the visual attention due to the frequent presence of a screen, many interaction parameters, and an interface sometimes lacking tactile feedback or touch marks that would allow to access them without needing to look at them. Furthermore, digital musicians will often prepare their instrumentarium just prior to the performance with a chosen set of ad-hoc musical elements[17] (when not re-coding the whole thing) which further complicates a perfect knowledge of the ergonomics of the instrument, which would do without the visual.

John's design has been driven by an economy of cognitive load for musicians. Being able to partly customize one's visualization interface thus does not mean to add more visual data to it, but to see only what is necessary for one to gain collective awareness.

[8] https://cycling74.com/
[9] http://meteor.com/
[10] https://d3js.org/
[11] Open Sound Control: http://opensoundcontrol.org
[12] That is, it can be totally reconfigured during the performance to offer a whole other set of sounds, processes and playing modes.
[13] such as chord grids, time signature, scales, etc.
[14] such as using Walter Thomson's Soundpainting or in a composition like John Zorn's *Cobra*.
[15] Such as E-Analysis [5] or the GRM Acousmographe [7].
[16] In the work of Schaeffer [18], Bayle [2] or MIM [17] among others.
[17] In an informal discussion, Thor Magnusson used the term "pre-gramming" for this particular work that precedes a concert.

3.6 A score for humans and for (digital) instruments

During the score playback, the server sends data to clients as events begin or end (Figure 4). This information can be used by the musician's instrument (according this DMI is connected to the network). But, as John is only a “semi-conductor”, it may as well be subject to the musician/client approval to allow some flexibility in the way the musician sticks to the score. Thus, it could have been devised that a specific *karma* recalls a corresponding preset of sounds in the musician's instrument, suitable for the *karma's* mood. But if the musician is still playing a previous other *karma*, s/he probably will not want this notification to automatically change the preset before s/he finishes one's current phrase. This “loose control” makes John's usage a little different from traditional sequencers.

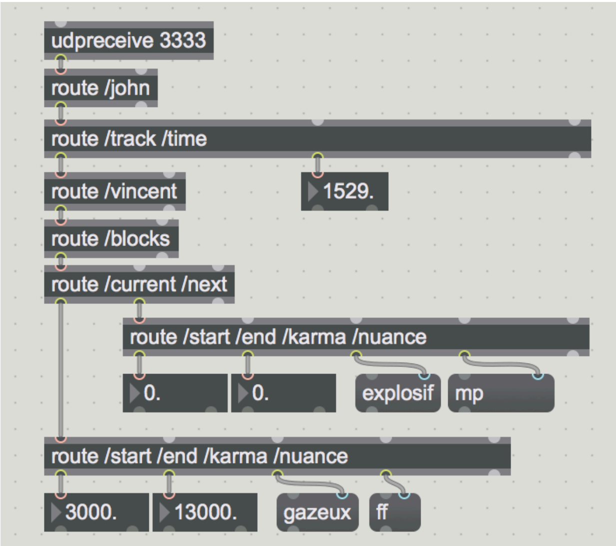


Figure 4. Receiving data in a Max instrument.

3.7 Showing the score?

Being able to read the interactions among the musicians in improvisation performances can contribute to the performance overall appreciation. Yet with DMIs, the spatial and energy decoupling between the instrumentalist's gestures and the resulting sound energy and location (on a possibly remote loudspeaker) confuses this reading. Screen-scores systems allow to share the score display with the audience more easily than printed scores and could help in this situation with the risk however, that it may “*detract from the dramatic performative aspects of the work*” among other reasons suggested by Cat Hope in [11]. Although John's score was never shown directly to the audience for this very reason, it has been used to control visual effects and lightings, meant both for stage design purpose and for helping listening and understanding of the music[18].

4. PERSPECTIVES

It has been acknowledged by the members of ONE that John was helping our creative process. However, there remains open issues like collective synchronization over rhythmic passages. Especially, anticipating a dynamic process is no trivial task and would probably require specific tools for that purpose, such as the animations proposed by Ryan Ross Smith in [19].

The concept of *local* and *global view* could probably be generalized to other shareable parameters. For example, being able to start a local playback in order to practice or prepare one's instrument on one's own. Similarly, it would be useful to work on another score than the ones loaded on others' clients. This de-synchronization raises however issues of versioning conflicts.

John's porting to a web technology is partly motivated by the possibility of future concerts involving a large number of musicians and where ever musician would be able to see his part with a simple web-browser. More developments will be needed to achieve such a performance, but there should not be technological locks.

Overall, computer-based scores give way to many possible interactions during performance time. Maybe the score should be considered as a collective instrument, which every musician and possibly the audience too, could play.

Acknowledgments

This work was partly carried out within a doctoral program supported by the Collegium Musicæ[19]. ONE is produced by Puce Muse[20]. Also, I would like to thank my colleagues from ONE for their valuable collaboration in the development of John : Laurence Bouckaert, Pierre Couprie, Hugues Genevois, Jean Haury, György Kurtág Jr. and Serge De Laubier.

5. REFERENCES

[1] S. Bhagwati, “Notational Perspective and Comprovisation”, in *Sound & Score. Essays on Sound, Score and Notation*, Leuwen University Press, pp.165–177, 2013.

[2] F. Bayle, *Musique acousmatique : propositions... positions*, INA-GRM/Buchet-Chastel, 1993.

[3] C. Canonne, “Improvisation Collective Libre et Processus de Création : Création et Créativité au Prisme de la Coordination”, *Revue de Musicologie*, vol. 98, no. 1, 2012.

[4] A. Clay, “Inventing Malleable Scores: From Paper to Screen Based Scores” in *Transdisciplinary Digital Art. Sound, Vision and the New Screen*, Springer Berlin Heidelberg, pp. 255–269, 2008.

[18] Examples include switching spotlights on musicians supposed to play, changing light hue according to the karmas, projecting aggregated sound waves as traces of the score, synching video, etc.

[19] http://collegium.musicae.sorbonne-universites.fr/

[20] http://pucemuse.com

[5] P. Couprie, "EAnalysis: developing a sound-based music analytical tool", in *Expanding the Horizon of Electroacoustic Music Analysis*, Cambridge University Press, pp.170–194, 2016.

[6] D. Fober, Y. Orlarey, and S. Letz. "INScore An Environment for the Design of Live Music Scores", in *Proceedings of the Linux Audio Conference,* Stanford University, USA, 2012.

[7] E. Favreau, Y. Geslin, and A. Lefèvre, "L'acousmographe 3" in *Actes des Journées d'Informatique Musicale (JIM'10)*, Rennes, France, 2010.

[8] J. Freeman, "Extreme sight-reading, mediated expression, and audience participation: Real-time music notation in live performance", *Computer Music Journal*, vol. 32, no. 3, pp. 25–41, 2008.

[9] V. Goudard and H. Genevois, "Mapping modulaire de processus polyphoniques", in *Actes des Journées d'Informatique Musicale (JIM'17)*, Paris, France, 2017.

[10] G. Hajdu, "Disposable Music", *Computer Music Journal*, vol. 40, no 1, p. 25-34, 2016.

[11] C. Hope, L. Vickery, "Screen Scores: New Media Music Manuscripts", in *Proc. International Conference on Music Computing (ICMC'11)*, Huddersfield, UK, 2011.

[12] C. Hope, L. Vickery, A. Wyatt, and S. James, "The DECIBEL Scoreplayer – A Digital Tool for Reading Graphic Notation", in *Proceedings of the International conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015.

[13] S. Lee, J. Freeman, and A. Colella, "Real-Time Music Notation, Collaborative Improvisation, and Laptop Ensembles", in *Proc. International Conference on New Interfaces for Musical Expression (NIME'12)*, Ann Arbor, Michigan, USA, 2012.

[14] E. Maestri, "Notation as Temporal Instrument", in *Proceedings of the International conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, UK, 2016.

[15] T. Magnusson, "Algorithms as Scores: Coding Live Music", *Leonardo music journal*, vol. 21, pp. 19–23, 2011.

[16] T. Magnusson, "Scoring with Code: Composing with algorithmic notation", *Organised Sound*, vol. 19, no. 3, pp. 268–275, 2014.

[17] MIM, *Les Unités Sémiotiques Temporelles – Éléments nouveaux d'analyse musicale*, MIM, Eska, 1996.

[18] P. Schaeffer, *Traité des objets musicaux,* Le Seuil, 1966.

[19] R. R. Smith, "An Atomic Approach to Animated Music Notation", in *Proceedings of the International conference on Technologies for Music Notation and Representation (TENOR'15),* Paris, France, 2015.

[20] L. Vickery, "The limitations of representing sound and notation on a screen", *Organised Sound*, vol. 19, no. 3, pp. 215–227, 2014.

[21] G. E. Winkler, "The Realtime-Score: A Missing-Link in Computer-Music Performance", in *Proc. Sound and Music Computing Conference (SMC'04)*, Paris, France, 2004.

EXPLORING WITH ILÉA ENSEMBLE: SHAPING FREEDOM IN IMPROVISED MUSIC

Kevin Gironnay
Université de Montréal
gironnayk@umontreal.ca

ABSTRACT

This paper approaches the topic of experimental improvised music within an ensemble, and will first present several techniques used in the field of non-idiomatic improvised music, especially in the case of collaborative improvisation such as within Cobra (J. Zorn) and Ensemble SuperMusique. After discussing the limitations of these techniques, the method of Ensemble ILÉA will be introduced along with its techniques and solutions to guide an ensemble without restraining the expressivity of the improvisers or limiting the experience of the audience.

1. INTRODUCTION

When an improvisation is performed between several musicians simultaneously, the formal direction of the performance is at a risk to be diluted by the many ears and brains involved in the creation process. In comparison, it is much easier to decide on a direction and the path in which to achieve this direction when one improvises alone. How can several musicians improvising simultaneously achieve this connection while they each possess their own different visions of the direction to follow? For example, if a crescendo is being built by the musicians, the decision of when and how to finish it can be problematic: the usual result being some form of crossfade toward somewhere different. However, if a musician decides to stop the crescendo abruptly, there is a risk of disappointment due to the lack of possibility that every other musician decides to cooperate. This abrupt silence of one musician may go unnoticed while the others decide either to continue the crescendo or begin a decrescendo, for example. This scenario may not be uninteresting and may indeed lead to new dynamics in the music, since improvisation is, by nature, potentially infinite, and the ability to react and to adapt quickly is an advantage that allows improvised music to be enjoyable by more than just the musicians playing. The risk of dilution of the decision-making in the musical direction remains, however: a lack of any clear intervention creates a homogeneous performance. Any initiative taken by a musician takes time to be registered and followed by the others, thus running the risk of a music made up of successions of crossfades. Decisive musical moments, that apparently only written music can generate, are unavailable. The researcher Anne Robineau summarizes this dichotomy of written and improvised music, stating: “In a derogatory way, improvisation is often associated with a lack of consistency, and even with an absence of shape. Composition is criticised for the opposite. It would be too rigid since it implies the writing of the music before the execution.”[1] [1] This sparked the search for an alternative method.

With an aim to shape improvised musical performance, and to avoid the situation where each decision drowns in the continuum of the other improvisers, some methods have already been invented and explored with success; two of which will be discussed here: the game-piece *Cobra* by John Zorn, and the gestures for conducting improvisation by the Ensemble SuperMusique. The method of Ensemble ILÉA, created in response to these two specific examples, will be presented as a solution to avoid both biasing the audience’s listening experience and constraining the musical expressiveness of the improvisers.

2. CONDUCTING AN IMPROVISATION ENSEMBLE: FROM JOHN ZORN TO SUPERMUSIQUE

Cobra is a musical piece composed in 1984 by American composer and musician John Zorn. Considered by the composer to be a game-piece featuring improvisers and a “game master” [2], *Cobra* is flexible, restricted neither by specific instrumentation nor by size. *Cobra* is a direct continuation of the principle of indeterminacy in music developed by the New York School (led by John Cage, Earle Brown, and Morton Feldman) in the 1950s and 1960s. Open form pieces such as Earle Brown’s *Twenty-Five Pages,* or Terry Riley’s *In C*, influenced younger composers, particularly in the downtown music scene of New York. This new approach to musical form naturally appealed to improvisers, as did soundpainting and graphic notation which are both highly interpretative methods of guiding musical performances.

The soundpainting technique, created by Walter Thompson in the 1970s and consisting of a set of gestures to trigger and modulate interventions of musicians, is a revealing example of the role a conductor bares in improvised music. Some open form pieces needed a conductor, Earle Brown’s *Available Forms*, for example. *Cobra* is another example of conducted improvised music.

[1] “De façon péjorative, l’improvisation est souvent associée à un manque de cohérence, voire à une absence de forme. C’est tout le contraire qui est reprochée à la composition. Celle-ci serait trop rigide puisqu’elle suppose l’écriture de l’œuvre avant son interprétation.” (Translation: K. Gironnay).

During a performance of *Cobra*, the conductor or "game master" has a set of cards expressing musical directions. Set gestures made by the musicians prompt the conductor's discretion to display one of these cards. The musicologist John Brackett defines the role of the conductor as a "prompter": "The prompter responds to requests made by the players by relaying information to the other members of the ensemble and while the prompter often functions as a conduit of information, she/he can choose to ignore requests by the players" [3]. The gestures, as fixed in the score, consist of a combination of pointing to a part of one's body (mouth, ear, head, and nose) and showing a number with one's fingers. The musical directions expressed on the cards are versatile and can control parameters such as volume, speed, or instrumentation (by creating duets, making silent improvisers play, and etcetera). The cards can also save a musical state that can be recalled upon later, and can even create an ending to the improvisation. Another feature of *Cobra* is the guerilla system, which allows certain improvisers to play at will with or without consideration of the director's instructions. Guerilla improvisers can be "terminated" by another improviser under certain conditions. *Cobra* can thus be classified as a musical role-play game where there is a harmonisation of the improviser's musical decisions. This harmonisation is made by the conductor, an outsider, whose interactions allow the musical content to be more dynamic.

Following this same method of a non-playing entity conduct a group of improvisers, the Montreal-based Ensemble SuperMusique created their set of gestures for conducting improvisation. Like *Cobra*, the conductor takes on a full-time role where their only task is to lead: an outside perspective. The conductor's gestures control the same kind of musical parameters as in *Cobra* with two main differences, the first of which being that the improvisers cannot ask for directions and instructions, it is the conductor alone who chooses the path. Therefore, the only musical input from the improvisers concerns the near future: they can decide how to express the given directions. Of course, these directions are influenced by what the improvisers play, but the improvisers do not hold the power to redirect the piece. The concept here is to follow the rules and to trust the conductor: non-compliance with directions (i.e. playing loudly when the conductor asks for a *pianissimo*) does not occur. The second main difference from *Cobra,* is that the gestures used by Ensemble SuperMusique are easily interpretable by the audience, providing clues about what is to come (unlike the coded gestures and cards used in *Cobra*).

In both methods, the moment when decisions in the musical direction are taken and applied is visible to the audience. The audience is thus drawn to these gestures and their attention is most likely to fixate on the relationship between the gestures and their musical effects. The discussion following performances using these methods tends to center around the significance of the gestures, the rules, and what was or was not understood, resulting in limited comments on the music itself. This is due to the gestures that are guiding the audience's listening during the improvisation: the audience tries to categorize the gestures and identify them, in order to recognize their effect on the resulting musical events. More importantly, the audience begins to anticipate gestures, and they live the musical phenomenon only with these expectations. The audience and its ears become biased.

3. ON THE MODEL OF ENSEMBLE ILÉA

3.1 The creation of the Ensemble

Ensemble ILÉA was created with the following intentions: first, to avoid a shapeless improvisation due to decision-making becoming silent when diluted between too many improvisers; and second, to avoid a distortion of the listener's experience due to an analytic and causal relationship between the music and ostentatious conducting gestures. In this objective, programs of improvisation guides were developed that are only visible to the improvisers.[2]

Figure 1. Example of a combination of members for an ILÉA show: 1 vibraphone, 1 flute, 1 clarinet, 2 laptops. The monitors visible only to the improvisers display improvisational guiding programs.

Creating Ensemble ILÉA was a way for me to continue the development of these guiding programs and to put them into use at the center of an improvisation ensemble, especially while completing my master's degree at Université de Montréal. My research focused on the use of improvisation in both improvised and fixed music.

My intentions as I continue to develop these guiding programs remain consistent and are inspired by both the conception of the form by Earle Brown as a "result of people's actions responding immediately to an environment shaped by possibilities…"[3] [4], and by Cornelius Cardrew's interpretation of indeterminacy, summarized by artist Matthieu Saladin as "a means to free what someone else thinks is constrained"[4] [5]. This undoubtedly influenced the relationship I tried to create between the guiding programs and the improvisers of ILÉA: shaping a direction, but not a strict path, so any improviser can explore and feel others exploring around the given direction.

---

[2] These programs were originally thought and developed within the improvised music collective Unmapped in 2012.

[3] "La forme comme résultante des actions de gens répondant immédiatement à un environnement décrit de possibilités…" (Translation: K. Gironnay).

[4] "L'indéterminé comme moyen en vue de libérer chez l'autre ce qui lui paraît contraint." (Translation: K. Gironnay)

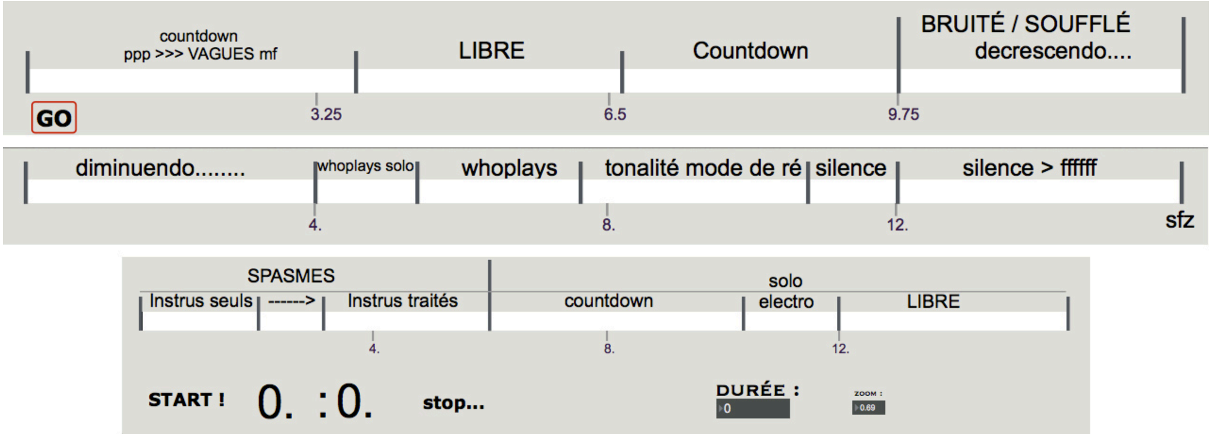


Figure 2. Examples of three different timelines used in three different concerts.

While less graphic, these programs are similar to animated scores developed by composers such as Cat Hope, Ryan Ross Smith, or Guðmundur Steinn Gunnarsson: although they graphically incorporate the time passing, they show conceptual suggestions as opposed to abstract indications.

Ensemble ILÉA itself consists of twelve musician-performers, including myself, of acoustic and/or electronic instruments, while their structure for performances (rehearsals and concerts) is variable.[5] The guiding programs were explained at the first meeting of the ensemble, and their use was introduced as an alternative to the methods of *Cobra* and Ensemble SuperMusique to create conducted or, more accurately in this case, "guided" improvisation. The resulting goal is to create improvised music where improvisers can musically evolve together through concise concepts and follow similar ideas, while eliminating potential confusion caused by deciphering the common direction that is being drawn. Therefore, with a common musical context, improvisers are more inclined to focus on the musical parameters to develop a relationship with the rest of the ensemble. They can also be less concerned about the significance of their involvement. Creating a frame of reference for an improvisation makes every idea relevant, in the way that ideas will be interpreted according to the frame of reference itself. For example, if when improvisers are responding to the concept "sporadic" and most are playing short musical events here and there, one improviser decides to play a sustained note: this decision will be interpreted by other improvisers according to this "sporadic" frame of reference. Is it a way to color the silence that is in between all the sporadic events? Or, is it a way to underline the briefness of the other musical events? Then, maybe the improvisers will start to play with the parameter of duration, and it might result in inverting the sound/silence ratio with a continuous sustained note where silence is being sporadic. In a totally free and not guided improvisation, a sustained note while everyone is playing sporadic musical events would often be directly considered as a proposition to go against the flow, or at least somewhere else. Here lies the risk of limiting improvisers' musical expression by making them uncertain of the actual flow, or of the musical direction.

These guiding programs, of which we will see in more detail in the following section, can additionally suggest who should play, and can also allow for synchronized musical events. It has been made clear to the ensemble since its conception that these guiding programs display suggestions that they may choose whether or not to follow. This is pertinent, since it was never my intention to minimize the improvisers' field of action, but to increase the consciousness of their actions: if they decide to not follow a suggestion, they should know the musical implication of this action. In the same way, if another improviser apparently does not follow the given suggestion (as in the "sporadic" example), the others should trust this decision as a conscious action. Gaston Bachelard stated in *L'intuition de l'instant* that "an accident is at the root of any attempt to evolve" [6], which is exactly how every action against the flow is observed in Ensemble ILÉA: as an attempt to evolve. In this way, suggestions are sufficient to shape an improvisation while keeping it free. Orders inhibit freedom, and the lack of common direction can easily lead to shapeless improvisations.

Since these guiding programs are displayed on onstage monitors that are visible only to the improvisers, the audience does not view any ostentatious signs of conducting (which can lead to a causal listening of the music, to attempts to try to understand what is happening, and to expectations such as when a conductor is about to make a gesture). With Ensemble ILÉA, abrupt changes are truly abrupt because they are not previously revealed to the audience.

3.2 Guiding programs

These programs are currently divided into three distinct parts: the timeline, the countdowns, and the "Who plays?" program.

The timeline program (Figure 2) shapes the whole improvisation. The duration of the improvisation can be set, and the time passing is illustrated in red inside the white bar, which also indicates the current section of the improvisation, and signals the sections to follow. The time-

[5] Every rehearsal and concert has its own combination of members. From duets to tutti, the ensemble has a total of 4083 potential combinations.

lines that Ensemble ILÉA creates can be quite complex with many constraints, or fairly simple with rather free indications. These indications are sometimes explicit (such as "crescendo", "acoustic instruments only", or "in D", for example), or other times full of imagery (such as "convulsions" or "blue"). It is also common to include "free" sections, to avoid the feeling of constraint. The timeline program can be viewed as the meta-program, as the two other programs are included within: they are "called" inside the timeline. A section of the improvisation can then be called "countdown", it then automatically opens the other program.

The countdown program (Figure 3) is one of the most used programs because of its efficiency and simplicity: it creates the possibility of synchronized musical events between improvisers without being seen or even predicted by the audience. This program creates successive countdowns, and is therefore versatile since the synchronizations that it creates can be used in many ways by the ensemble. It can be used to create complex impacts, made up of the different impacts played by the improvisers when the countdown reaches zero. Impacts may vary in length, which can create interesting sound materials: for example, a synchronized electronic impact with a saxophone slap can then slip into a resonance made from a flute's high note and a bass clarinet's low note. These complex sound objects surprise the improvisers and can lead to new directions until the following zero. In between these synchronized moments (that can last from anywhere between several seconds to around one minute), improvisers are free to do whatever they want with the new direction given by the last impact. Another feature of this program is that since improvisers have a visibility of the time before the next synchronization, they can create a *crescendo* into it and to shape the upcoming impact. This tension is, in my opinion, an effective way to engage the audience into an active listening, by presenting perceptible breaking points and by giving a shape to the performance.

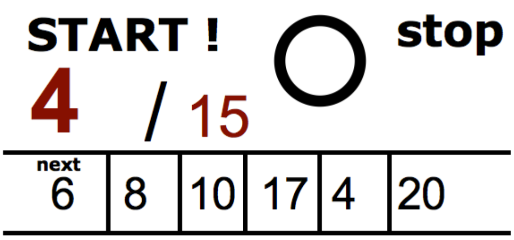


Figure 3. Example of a countdown: a synchronization point arrives in four seconds (bigger, red number), on that last countdown that lasted fifteen seconds. The following countdowns will be of six, eight, ten, seventeen, four, and twenty seconds in length.

Different uses of the countdowns are constantly created within the ensemble: in the first timeline presented in Figure 2, the performance starts with a "dynamic waves countdown". This is another common use of the countdown program where the synchronized point (i.e. reaching zero) should be a peak of the dynamic (*mezzo forte* in Figure 2), while the rest of the time improvisers should play within another dynamic (*pp* to *ppp*, in Figure 2). The timing of the *crescendo* toward the peak and the *decrescendo* to come back to the original dynamic is absolutely free, and can be articulated quickly by some improvisers and slowly by others. The result is a series of dynamic waves.

The third program is called "Who plays?", and generates a portion of the ensemble invited to play (Figure 4). It can be programmed to generate soli, duets, trios, and so forth. It can also suggest a *tutti*.

Figure 4. "Who plays?" is generating a portion of the ensemble invited to play.

This program allows chosen improvisers to develop a new direction (or a new way of continuing an ongoing direction) with a reduced size of players. As always in Ensemble ILÉA, other improvisers are never prohibited to play: they can join the improvisers selected by the program while keeping in mind that they might be perceived as intruders. Once again, it is also about setting up a common context so that anomalies (i.e. improvisers going against suggestions) can be noticeable and so that others can react. Even when followed as instructed, this program brings musical changes and shapes the performance without disturbing the audience from concentrating only on the musical phenomenon.

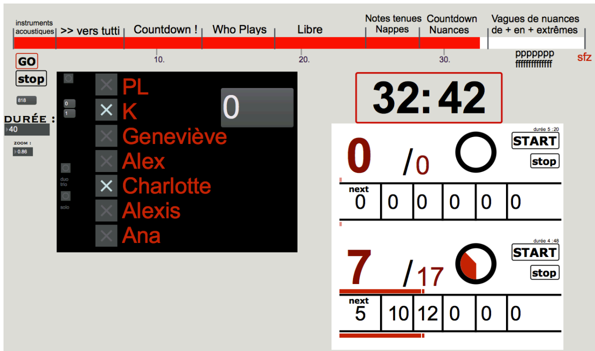


Figure 5. The guide program with all the programs shown (the timeline, the countdowns, and "Who plays?")

The specific guiding program for a performance is discussed between the members playing before said performance. Mainly, we decide the form of the improvisation, the sections of the timeline, their duration, and their theme (keywords, concepts, countdowns, or "who plays?"). Then, as shown in Figure 5, all the programs are linked to each other so only a simple push of the "Start" button is required to begin the performance.

4. CONCLUSION

Although Ensemble ILÉA is an improvised music ensemble, it is not its purpose to make improvisation visible as an aesthetic. Improvisation is, within the Ensemble, more of a creation process. The whole reason for my research on improvisation through these programs is to create music that can have the effectiveness of written music, and the freedom and innovation of improvisation.

With two albums released and over a dozen shows performed, Ensemble ILÉA is keen to continue to produce music while developing new guiding programs.

5. REFERENCES

[1] A. Robineau, *Étude sociologique de la musique actuelle au Québec: le cas des Productions Supermusique et du Festival International de Musique Actuelle de Victoriaville*. Ph.D. thesis, Université de Montréal, Canada, 2004.

[2] J. Zorn, “The game pieces,” *Audio culture: readings in modern music*, pp. 196–207, 2004.

[3] J. Brackett, “Some notes on John Zorn’s *Cobra,*” *American Music*, vol. 28, no. 1, pp. 44–75, 2010.

[4] E. Brown, “Sur la forme”, *Musique en jeu*, no. 3, pp. 29–44, 1971.

[5] M. Saladin, “La partition graphique et ses usages dans la scène improvisée,” *Volume!*, vol. 3, no. 1, pp. 31–57, 2004.

[6] G. Bachelard, *L’intuition de l’instant*. Stock, 1931.

IMPROVING SIGHT-READING SKILLS THROUGH DYNAMIC NOTATION – THE CASE OF COMPROVISADOR

Pedro Louzeiro
Universidade de Évora, Évora, Portugal
Centro de Estudos de Sociologia e Estética Musical, Lisboa, Portugal
Fundação para a Ciência e a Tecnologia (FCT), Lisboa, Portugal
pedrolouzeiro@gmail.com

ABSTRACT

This paper proposes an approach to sight-reading improvement using a dynamic notation system – Comprovisador. The system was created with the goal of coordinating musical performances in which a soloist improvises and an ensemble of musicians sight-read a staff-based dynamic score. This situative score is therefore generated by Comprovisador's algorithms which feed on the soloist's improvisation. Musicians read the score from computer screens, in a local network. This kind of musical practice requires performers to be good sight-readers. A good sight-reader (of traditional notation) often relies on pattern recognition, understanding of musical structure and other abilities which come from being familiarized with certain repertoires – but when dealing with situative scores these abilities are seldom relevant. With this consideration, a Practice Tool was developed as part of Comprovisador to allow musicians to get acquainted with the system's notation interface and to learn (not the notes, but) how to deal with not being able to predict patterns or structure. After further development, this tool was tested by music students and teachers in order to assess its applicability in an educational context regarding improvement of sight-reading skills. A study with those participants is presented to validate the utility of the system and identify areas for further development.

1. INTRODUCTION

The motivation on addressing issues related to sight-reading has evolved from two directions: 1) as a qualified solfege teacher with over fifteen years of experience, the author has been interested in ways to help students improve their skills, and 2) as a creator, while developing a real-time notation system and putting it into action during rehearsals and performances, the author has worked in collaboration with competent sight-readers, looking into ways of improving the system's notation interface in order to meet and expand their abilities.

The system – Comprovisador – was originally designed to carry out comprovisation performances using real-time algorithmic composition and dynamic staff-based notation. To engage in such musical practice, performers are expected to have excellent sight-reading skills as well as the ability to adapt to new performance situations. A Practice Tool was created within Comprovisador to help performers improve those skills while getting acquainted with the system's notation interface. Later, this was seen as an opportunity to broaden the system's application and adapt it as a tool for music students.

In order to assess the system's applicability in this new educational context, a user study with quantitative and qualitative data is discussed herein.

2. BACKGROUND

As described in recent publications [1, 2, 3], Comprovisador is a system designed to enable mediated soloist-ensemble interaction using machine listening, algorithmic compositional procedures and dynamic notation, in a networked environment. As a soloist improvises, Comprovisador's algorithms produce a score in real-time that is immediately sight-read by an ensemble of musicians, creating a coordinated response to the improvisation. This interaction is mediated by a performance director who does so by manipulating algorithmic parameters. Implementation of this system requires a network of computers in order to display notation (separate parts) to each of the musicians playing in the ensemble. More so, wireless connectivity enables computers – and, therefore, musicians – to be far apart from each other, enabling space as a compositional element.

Comprovisador consists of two applications – host and client. Both are developed in Max 7 [4] using Bach library [5] for its notation features and computer assisted composition tools. To this date, the system has been used in eight public performances, which are documented in the project's website: comprovisador.wordpress.com [6]. The website [1] also contains video examples of the dynamic score in action.

The name "Comprovisador" derives from the term comprovisation, which has been used by several authors such as Lawrence D. "Butch" Morris [7], Richard Dudas [8] and Sandeep Bhagwati [9], among others. Bhagwati has

[1] Furthermore, it is possible to download and install the client application. Musicians can install it if they are to perform in a future "Comprovisação" or simply if they wish to practice sight-reading.

proposed an eloquent definition of the term [2]. A simplified version according to my reading of his definition, would be: a musical performance context where both composed and improvised elements coexist in aesthetically relevant interdependency.

Comprovisador was indeed conceived as a tool to enable such musical performance contexts where solo improvisation and composed response are, in fact, interdependent: thanks to real-time composition algorithms, the composed response is highly dependent on incoming improvised material; and by virtue of a feedback loop, the improviser's decisions are affected by composed elements. One can say it forms a dialectical relationship, for a composed response could not exist without the improvisation and the improvisation could not be the same without the composed response. This interdependency is further extended by the presence of a mediator.

Aesthetic relevance is the main concern when tailoring composition algorithms [3]. Likewise, it is of utmost importance when making choices in notation type and notation interface design [4]. In Section 3, these choices will be examined in order to better understand what the system demands from the performer in terms of sight-reading skills and, consequently, the original goals of the Practice Tool which was later adapted for use in an educational domain.

3. DEVELOPMENT

3.1 Notation Type

Although there are several different approaches to real-time notation (many of which are featured on animated notation dot com [10], a website run by Ryan Ross Smith), most choices fall into two broad categories: staff-based notation and non-staff notation [5]. The latter has many advantages: it encourages performers to be creative in translating non-conventional signs into sound and music, it exempts performers from the responsibility of not playing wrong notes and, potentially, it embodies an aesthetic value as a visual or multimedia experience.

On the other hand, while it is true that staff-based notation may put performers in a less creative and less forgiving situation (and the idea of projecting the score so that audiences may follow performers' mistakes might create additional anxiety to the situation), it is also true that it enables a greater compositional control over certain musical parameters – namely, pitch and harmony. Wrong notes as well as timing discrepancies and other audible mistakes are bound to occur. But it is possible to take this failure expectancy into account and somehow incorporate it in the aesthetics of the piece.

A good example of this incorporation is Nick Didkovsky's "Zero Waste" [11, 12], for sight-reading pianist and real-time transcription algorithm. In this piece, the performer sight-reads two initial measures of software-generated music while the algorithm transcribes the performer's rendition. The transcription is immediately displayed to the performer and the process repeats itself. Both performer and algorithm are expected to fail in order for proliferations of the initial gesture to take place. As Georg Hajdu points out [12], the abstract, chromatic quality of the material selected for the opening bars prevents an error from being perceived as such. Instead, error becomes the shaping force of the piece.

During early development stages of Comprovisador, the concept of "extreme sight-reading" proposed by Jason Freeman [11] had an influence on the choice of using staff-based notation. The influence came from the concept expressed in the title itself rather than from a particular example found in the article. Strategies were conceived towards the design of a functional notation interface, considering the problem of error and all its surrounding issues. The element of time was found to be crucial in this conception, as will be exposed in Section 3.2.

3.2 Notation Interface Design

3.2.1 Synchronized Attacks

In a hybrid type of performance such as comprovisation, it is presumable that the listener will be looking for clues as to what is being improvised and what is being composed – and even how effective is the notation system. Regarding listener's ability to discern between composed and improvised music, Lehmann and Kopiez propose that " 'togetherness' and precision of an ensemble may indicate composition, while a higher degree of entropy could signal improvisation" [13]. In this line of thought, we find synchronization to be an effective way to let the listener perceive organization as opposed to chaos, hinting at what is being composed in real-time.

In a synchronized attack, even if a few notes are false or missing, there is no way the listener can tell. And, as we have seen, it is plausible to have a mistake becoming a shaping force – in this case, by influencing the improviser's playing.

In order to have synchronized attacks in an extreme sight-reading context, the issue of time is of great importance. Firstly, musicians need time do recognize each note or group of notes (or, as John Sloboda would phrase it, to register pitch symbols in memory [14]); secondly, they need time to prepare the notes on their instrument; lastly, they need to be precisely cued – and effective cuing involves very specific timing. And motion (see [15]). In any of these three steps, problems may arise leading to delays and jeopardizing synchronization. Hence, establishing a reading time window and implementing a visual cuing device (consisting of a bouncing ball – see Figure 1) were our first design choices. Both would have to be time-adjustable, according to musical goal and/or technical difficulty.

[2] "[M]usical creation predicated on an aesthetically relevant interlocking of context-independent and contingent performance elements" [9].

[3] The compositional procedures used in Comprovisador are explained in [1]. A discussion on performance mediation using the system's control interface is made in [3].

[4] A thorough description of the notation interface can be found in [2].

[5] One could say graphic notation, but that term would not encompass works where notation goes beyond the scope of graphical signs. Such is the case of Jason Freeman's "Glimmer" where colored LED light tubes convey pitch and loudness information to performers. [11]

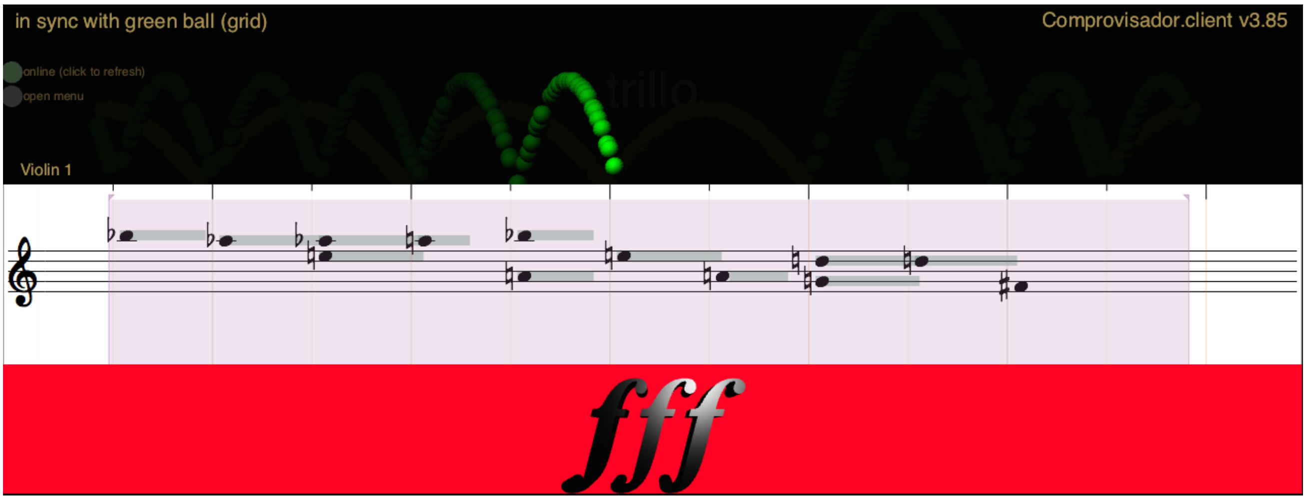


Figure 1. Comprovisador.client – notation interface – proportional notation and bouncing ball.

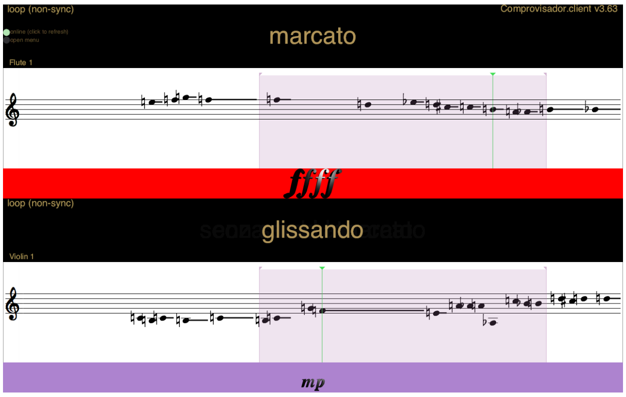



Figure 2. Comprovisador.client – notation interface – “loop (non-sync)” – dual instrument configuration.

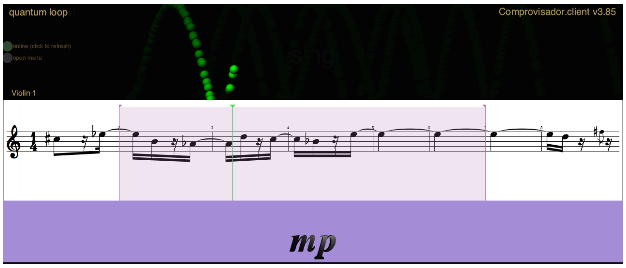


Figure 3. Comprovisador.client – notation interface – standard rhythmic notation and loop region.

3.2.2 Motivic Exploration

Apart from synchronization, another perceptual evidence of compositional process could be motivic exploration. If the listener is confronted with a melodic fragment being played simultaneously by various instruments and/or transformed in a coherent manner, he or she might perceive it as composition. Here, simultaneity refers to a given short time interval we perceive as present (specious present – see [16]). It does not imply unison or homophony but rather polyphony (and even micropolyphony).

This textural procedure, if done with no regard to synchronization and no special attention to meter or rhythm, allows musicians to serenely read the score and render the melody with far less mistakes than otherwise would be possible. At the same time, a dense texture will help in disguising the occasional missed note. This led to the use of proportional notation, a looping melody, a linear cursor and the verbal instruction: “non-sync” (see Figure 2).

3.2.3 Standard Rhythmic Notation

It should be interesting to provide rich and cleanly organized textures, made of melodic, rhythmic and harmonic elements, as well as formal ones, like repetition and variation. Standard notation (see Figure 3) allows all that while adding two new levels of time: meter and rhythmic durations. The problem lies in the fact that the more elements are added, the more difficult sight-reading becomes and the more exposed musicians feel.

A progressive approach to the various elements could conceivably be the answer. We can compare it to when a musician is learning a new piece of music. If they encounter a difficult passage, they might focus solely on the notes, repeating the passage several times until they are sure to play all the correct pitches. And only then will they try and play those pitches in precise rhythm and tempo. Emulating this process, when in Comprovisador standard rhythmic notation is activated, the notes that were previously displayed in proportional notation will be kept the same, enabling the performer with the chance to focus solely on the new element: rhythm.

3.3 Practice Tool

Development and enhancement of these and other features of Comprovisador was only possible thanks to the feedback of musicians who tested the system in rehearsals and performances. As a way to enable performers to get acquainted with the system's notation interface and its idiosyncrasies, a Practice Tool was developed featuring an elementary graphic user interface (GUI) for parameter control. This way, even before the first rehearsal, they were able to experience sight-reading in a simulated performance context, being subject to unpredictable note patterns (thanks to a random walk algorithm) and to a specific cuing strategy – the bouncing ball.

The Practice Tool was especially useful in situations where musicians and developer were in different locations. The tool allowed to obtain valuable feedback from a distance and perform bespoke enhancements in time for the first rehearsal.

4. A POSSIBLE NEW DIRECTION

Carlos Guedes [17] states that one of the goals on the development of real-time composition applications is "to open a new and potentially revolutionary way of education and active enculturation with unfamiliar musical styles". What about dynamic notation applications? Can they play a significant role in a new way of improving sight-reading skills?

Music sight-reading has long been a subject of research in the field of music cognition. Many authors have pointed out pattern recognition and understanding of musical structure as a few of the most important skills among good sight-readers [18, 19, 20, 14, 21]. Pianist Boris Goldovsky, interviewed by Thomas Wolf, said: "you read only a fraction of the notes and you guess at the others. A good sight-reader gets a total image of a page and extrapolates what is going on exactly" [19]. Evidently, such an ability can only come from being familiarized, through years of training, with the rules and patterns common to a certain style of music, a certain repertoire. Also, this statement is based on the assumption that the sight-reader will have the chance to at least take a glance at the whole music page before beginning to play. But most dynamic score sight-readers do not have that luxury. Hence, they have to develop other skills in order to become successful at that task. Could generative algorithms, such as the one implemented in Comprovisador's Practice Tool, be of aid to the development of those skills?

While searching for applications or systems that use dynamic notation and aim for sight-reading improvement we did not find anything relevant. There are great amounts of smartphone applications intended for music notation learning and some do use dynamic score technology. Yet, the majority uses previously written (coded) music excerpts and it is rare to find one that joins dynamic notation technology with the power of generative algorithms.

In July 2017, during a talk at the 2nd "European Saxophone Congress", the possibility of using Comprovisador's Practice Tool as a way for saxophonists to improve sight-reading skills in a microtonal context was presented. A trial had been carried out with a small group of professional saxophone players and results were presented during the talk. Some adaptations were done to the system in order to be possible to collect user practice data for study. No other changes were made. Results pointed to potential benefits in using the application but it became clear that a progressive learning approach strategy would have to be devised.

parameter name	parameter description
range	allows control of range in concert pitch and in transposition (automatically set when choosing an instrument)
note selection	a selectable keyboard allows turning on or off certain notes or even whole registers
microtone selection	microtones can be hand-picked from a `[bach.tonnetz]` object
microtones length	enables the user more time to stabilize fingering and tunning whenever a microtone is output
tone division	selects all notes matching the set tone division
scale picker	selects all notes matching the chosen scale
polyphony	sets maximum, ranging from single notes to full polyphony (value depends on the instrument)
chord threshold	sets a threshold in milliseconds under which no chords are allowed (only single notes)
reading time window	adjusts the sight-reading window in milliseconds
maximum step	sets the maximum melodic step in half-tones
note rate or "flux"	ranging from slow to fast (proportional notation)
rhythm base	minimal units for standard notation (allowing creation of simple patterns, and to progress)
variation rate	limits the occurrence of variations of a melody (in loop mode), ranging from static to frequent
user presets	enables the user to store and recall parameter presets

Table 1. Comprovisador.client – Practice Tool's parameter list (user controlled). New parameters are marked in bold.

Such a strategy was indeed planned out aiming not only at the microtonal issue but also at a more general context. Its implementation consisted on designing a new GUI for the Practice Tool with more controllable parameters and the possibility of storing user presets (see Figure 4 and Table 1). The goal was to enable the user to match the difficulty level of the algorithmic outcome to his or her degree of proficiency.

During the implementation of this GUI, another trial was carried out – this time with music students and teachers of different instruments – in order to assess the usefulness of this tool in a generic music education context [6]. However, parameters controlling standard rhythm notation were not yet implemented when the trial took place.

5. METHOD

The trial was carried out in a music school in Portimão (south of Portugal), with 14 participants, 9 of which were students and 5 were teachers, playing the following instruments: saxophone (3 participants), violin (4), piano (3), guitar (2), double bass (1) and trombone (1). Students'

[6] It is worth noting that this tool should never be considered as a substitute for actual repertoire sight-reading, which is the best way to acquire pattern recognition skills.

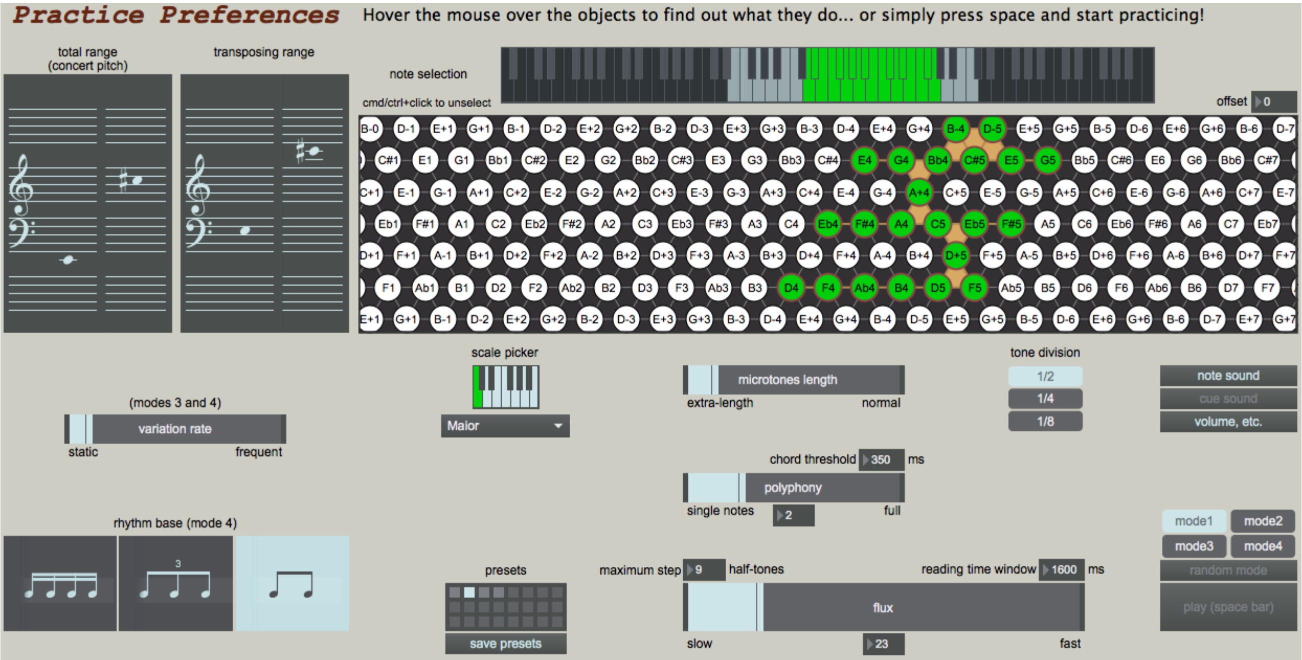


Figure 4. Comprovisador.client – practice preferences window.

ages ranged from 13 to 18. The level of experience of the participants was heterogeneous, as can be inferred by the age range and by the fact that it mixes students and teachers. Yet, none had had experience with microtonality.

Participants were individually asked to sight-read from the computer screen without any detailed explanation. As they were playing, some parameters would be manipulated in an attempt to match their proficiency level and, while doing so, we would explain what each parameter was meant to do. Towards the end of the exercise, it was explained how parameters could be stored as user presets for later recall as a way to keep track of progress. Participants were then asked to explore this feature in conjunction with the parameters previously manipulated.

With instruments that enable microtonal playing, an approach to the matter was carried out, activating only one microtonal note (in some cases, two notes) [7] and limiting the range so that the algorithm would focus on the register surrounding the chosen microtonal note. Also, longer duration time was assigned to this same note in contrast to regular notes, this way allowing stabilization.

Participants were directly observed and were videotaped while playing, for further observation. After the exercise was complete (which took around 15 minutes per participant), they filled up a form containing three sections: quantitative assessment, qualitative assessment and suggestions.

[7] Some solfege books [22, 23] address note reading through a block-building approach where, for example, lesson 1 features only notes C and D, lesson 2 introduces note E, and so on. This approach may be useful when applied to any type of exotic notation – as microtonal is for a large number of musicians, students and professionals alike.

	Notes	Micr.	Std.R	Prp.R	Dyn.S
N Valid	14	11	14	14	14
Mean	5,14	4,55	4,14	4,57	4,57
std.Dev.	,770	,873	1,351	,938	1,158
Min.	4	3	2	3	3
Max.	6	6	6	6	6

Table 2. Assessment of Comprovisador.client as a tool for sight-reading skills improvement. Categories: standard notes, microtones, standard rhythm, proportional rhythm, experience with dynamic notation systems. Rating: from 1 to 6 (1 being not useful and 6 being very much useful).

6. RESULTS AND DISCUSSION

From observation, it was possible to perceive that all participants, with the exception of two students, were able to figure out (by themselves or needing very little explanation) how to play in sync with the bouncing ball.

In all cases, with proportional notation it was possible to match the parameter settings to the proficiency level of each individual so that it always became an interesting sight-reading challenge.

The progressive microtonal approach, starting with known notes / fingerings and adding only a selected microtonal note (assigned with a longer duration), was regarded as successful (from observation, backed by answers to the form). Violinists seemed to struggle a bit more than other instrumentalists but we were not able to find a relevant cause for that contrast. Pianists obviously did not experience this part of the exercise.

As expected, it was observed that work needed to be done in the standard rhythmic notation part, in order to enable beginner students with a viable tool.

From form responses and regarding quantitative assessment of the application as a tool for sight-reading skills improvement (see table Table 2), results were encouraging in the category of standard notes. Results were also positive in the categories of proportional notation, experience with dynamic notation systems, and microtones. Although, the latter had less three responses (pianists). The lowest rated category was standard rhythm, as expected.

There was an optional category “other” where two participants (both of them wind instrument teachers) added “tunning”, rating it with the highest score. They highlighted the benefit of playing in tune with the sound produced by the computer.

Regarding qualitative assessment of functionality and appearance, the responses were the following. The bouncing ball was considered useful / effective / helpful, except for two students who deemed it confusing. The dynamics bar was considered useful / effective / legible, but nonetheless some participants reported it to be too fast / difficult to comply with / very challenging; one participant highlighted the 3D animation as a good solution. Verbal instructions had very similar responses.

Regarding the observed and reported ease to synchronize with the bouncing ball, it is in line with Richard Picking’s findings on his study where he compares three types of animated time-location tracker, in the context of reading music from computer screens (versus reading from paper). The subjects of his study reported the “jumper tracker” (which is analog to our bouncing ball) to be the preferred one [24].

My preliminary conclusion taken from observation and commentaries is that young students tend to ignore dynamics and verbal instructions – and they are fine with it. Advanced students and teachers tend to get a bit frustrated when not able to comply with everything (notes, dynamics and instructions) but also feel rewarded when they do.

There were many voluntary commentaries and suggestions. The preset management system and GUI for parameter control were regarded as having good configuration / ease of use / good control over “excess of randomness”. Pianists complained about insufficient spacing between staves. There is actually only enough space for the central C line – which is standard in many computer music applications that use GF staves – but pianists are not necessarily used to it. Some participants mentioned that the duration line should be of a lighter color because it interferes with the perception of the staff-lines. This is now fixed, as shown in Figure 1. An interesting suggestion was to implement a way to have harmonic structures as a base for the generative algorithm.

Without surprise, many comments about standard rhythm were made, for example: “Everything is changing all the time due to excess of variations”, or “It needs patterns”.

To sum up, results seem encouraging (although they have to be put in perspective regarding the small sample size) suggesting there are advantages in the use of Comprovisador.client’s Practice Tool as a way of improving certain sight-reading skills, with special focus on skills pertaining to the dynamic notation realm. Regarding the least explored field – standard rhythm – we believe there is equal potential, now that the GUI’s development is complete.

7. FUTURE WORK AND CONCLUSIONS

Much work has been done, meanwhile, in terms of correcting the reported issues, namely the color of the duration line, which is now translucent green, as well as the standard rhythm controls.

Apart from the controls, standard rhythm was enhanced at the quantizer level. Here, instead of writing two 4/4 measures, the algorithm writes eight 1/4 measures. This allows two things: 1) complex patterns are conveniently delineated by bar lines and thus easier to decipher; 2) long notes unfold into tied quarter notes, making it easier to count the beats – which is especially important when a loop is set in a way that a long note becomes truncated. In Figure 3, we can see this happening: there are three tied quarter notes that would otherwise be written as a dotted half note. The loop region is truncating the 3rd quarter note. If it was written as two 4/4 measures, the loop region would end in an ambiguous, white portion of the measure, corresponding to the duration of the dot, which would be confusing for the reader.

In the medium term, we might pursue the suggestion of implementing a way of having the generative algorithm obey a harmonic structure. This structure could be cyclic or generative.

Future developments shall include articulation signs and other features that will be made available in the upcoming Bach version. Among these features is an algorithm for respelling accidentals in a more musical way, in atonal contexts.

One goal, of course, is to do further testing, if possible with a larger sample size and during a longer period of time, so to be able to measure actual learning progressions and observe commonalities that might emerge among multiple participants.

Acknowledgments

I should like to thank my PhD supervisors Christopher Bochmann and António de Sousa Dias for their advice and continued support. I would also like to thank Filipe Valentim, for providing all the conditions necessary for the trial to take place, and all the participants. Furthermore, I want to thank Adriana Ferreira Alves, from the Institute of Social Sciences of the University of Lisbon, for the data analysis and Sara Ross for proofreading. My attendance to TENOR 2018 conference was only possible thanks to the financial support of Câmara Municipal de Lagos, Portugal. This research is supported by FCT by means of a PhD studentship (POCH / EU). I dedicate this work to the memory of composer José Luís Ferreira, avid supporter of this research.

8. REFERENCES

[1] P. Louzeiro, "Real-Time Compositional Procedures for Mediated Soloist-Ensemble Interaction: The Comprovisador," in *Mathematics and Computation in Music: Proceedings of the 6th International Conference, MCM 2017*. Mexico City, Mexico: Springer, 2017, pp. 117–131.

[2] ——, "The Comprovisador's Real-Time Notation Interface," in *Proceedings of the 13th International Symposium on Computer Music Multidisciplinary Research*, Matosinhos, Portugal, 2017, pp. 340–351.

[3] ——, "Mediating a Comprovisation Performance: the Comprovisador's Control Interface," in *Proceedings of the International Computer Music Conference (ICMC)*, Shanghai, China, 2017, pp. 362–367.

[4] M. Puckette, D. Zicarelli, R. Sussman, J. K. Clayton, J. Bernstein, B. Nevile, T. Place, D. Grosse, R. Dudas, E. Jourdan, M. Lee, and A. Schabtach, "Max 7: Documentation." [Online]. Available: https://docs.cycling74.com/max7/

[5] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015.

[6] P. Louzeiro, "Comprovisador – about," 2017. [Online]. Available: https://comprovisador.wordpress.com/

[7] T. T. Stanley, "Butch morris and the art of conduction," Ph.D. dissertation, University of Maryland, 2009.

[8] R. Dudas, "Comprovisation: The various facets of composed improvisation within interactive performance systems," *Leonardo Music Journal*, vol. 20, pp. 29–31, 2010.

[9] S. Bhagwati, "Notational perspective and comprovisation," in *Sound & Score: Essays on Sound, Score and Notation*, P. de Assis, W. Brooks, and K. Coessens, Eds. Leuven University Press, 2013, pp. 165–177.

[10] R. R. Smith, "Animated notation dot com," 2014. [Online]. Available: http://animatednotation.com

[11] J. Freeman, "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Computer Music Journal*, vol. 32, no. 3, pp. 25–41, 2008.

[12] G. Hajdu, "Disposable Music," *Computer Music Journal*, vol. 40, no. 1, pp. 25–34, 2016.

[13] A. Lehmann and R. Kopiez, "Can expert listeners hear if a piece is improvised or composed?" in *Proceedings of the 11th International Conference on Music Perception and Cognition*, Seattle, USA, 2010, pp. 577–580.

[14] J. A. Sloboda, "Visual Perception of Musical Notation: Registering Pitch Symbols in Memory," *Quarterly Journal of Experimental Psychology*, vol. 28, no. 1, pp. 1–16, 1976. [Online]. Available: https://doi.org/10.1080/14640747608400532

[15] J. Colson, *Conducting and Rehearsing the Instrumental Music Ensemble: Scenarios, Priorities, Strategies, Essentials, and Repertoire*. Scarecrow Press, 2012.

[16] R. Le Poidevin, "The experience and perception of time," *Stanford Encyclopedia of Philosophy*, 2015. [Online]. Available: https://plato.stanford.edu/entries/time-experience/

[17] C. Guedes, "Real-Time Composition, why it still matters: A Look at Recent Developments and Potentially New and Interesting Applications," in *Proceedings of the International Computer Music Conference (ICMC)*, Shanghai, China, 2017, pp. 162–167.

[18] K. L. Bean, "An experimental approach to the reading of music," *Psychological Monographs*, vol. 50, no. 6, pp. i–80, 1938.

[19] T. Wolf, "A cognitive model of musical sight-reading," *Journal of Psycholinguistic Research*, vol. 5, no. 2, pp. 143–171, 1976.

[20] J. Sloboda, "The Eye-Hand Span - An Approach to the Study of Sight Reading," *Psychology of Music*, vol. 2, no. 2, pp. 4–10, 1974. [Online]. Available: https://doi.org/10.1177/030573567422001

[21] A. J. Waters, G. Underwood, and J. M. Findlay, "Studying expertise in music reading: Use of a pattern-matching paradigm," *Perception & Psychophysics*, vol. 59, no. 4, pp. 477–488, 1997. [Online]. Available: https://doi.org/10.3758/BF03211857

[22] B. Haultier, *Nouveau Solfège*. Paris: Editions Musicales Hortensia, 1977, vol. 1.

[23] J.-C. Jollet, *Jeux de Rythmes et Jeux de Clés*. Paris: Billaudot, 2000, vol. 1.

[24] R. Picking, "Reading music from screens vs paper," *Behaviour & Information Technology*, vol. 16, no. 2, pp. 72–78, 1997.

ZSCORE: A DISTRIBUTED SYSTEM FOR INTEGRATED MIXED MUSIC COMPOSITION AND PERFORMANCE

Slavko Zagorac
Goldsmiths, University of London
slavko@zagorac.com

Patricia Alessandrini
Goldsmiths, University of London
p.alessandrini@gold.ac.uk

ABSTRACT

This paper proposes a distributed system design for mixed ensemble music composition and performance of stave-based dynamic scores. ZScore is a collection of third-party and newly-developed components which aims to implement described networked notation solutions. The solution scope includes complex notation authoring, reliable score data distribution over a network to heterogeneous clients, precise performance scheduling and dynamic rendering of interactive scores. Taking the specification of optimal system features as a starting point, this paper looks at suitable solutions from other industries where high-throughput, low-latency systems have been successfully implemented. It presents the case for SVG-based notation representation, its distribution over a reliable message-oriented middleware and the innovative alternating pane layout design for dynamic notation rendering. Finally, the paper describes the current state of ZScore development and outcomes from initial user trials. It concludes with future perspectives towards realizing the underlying ambition behind this project: to blur and thereby call into question the traditional boundaries between the roles of a composer, performer, conductor and audience through the effective utilization of cutting-edge technology.

1. INTRODUCTION

During the last two decades, a number of software solutions able to dynamically render music scores distributed over a local or wide area network have been developed. Amongst the available solutions are: InScore [1], Quintet.net [2] and NetCanvas [3] which utilize MaxScore [4] for notation, Bach [5], Decibel ScorePlayer [6] and dfscore [7]. Antescofo [8] offers integrated mixed ensemble composition and performance notation while recent Odot developments also allow for OpenMusic [9] notation integration over a network. Some composers and laptop orchestras in particular develop proprietary composition and performance software in programming environments such as MaxMSP, Processing, SuperColider and ChucK programming language. While all notation applications share common high level functional objectives, their internal data models, score rendering versatility, system dependencies, time synchronization strategy and modes of communication can vary greatly.

Open Sound Control (OSC) messaging protocol has emerged as the leading choice for data and control message encoding and distribution between music notation applications over a computer network. The majority of OSC implementations rely on User Datagram Protocol (UDP) connectionless communication. InScore supports OSC natively, while MaxMSP-based solutions use various third-party OSC implementations. Quintet.net additionally uses TCP protocol where reliable messaging is required. More recently, several solutions which utilize Websocket point-to-point connection technology have emerged, such as NetCanvas which displays notation generated in MaxScore and dfscore which relies on Node.js server event distribution.

Odot framework middleware oriented messaging [9] is a welcome step towards network services abstraction. It wraps OSC protocol and provides transcoding to JSON, SVG and S-Expressions, as well as bindings to Javascript and Lisp. Landini [10] can also be classified as a form of a middleware as it creates an additional layer between music applications communicating over OSC. Landini implements a reliable, ordered message delivery protocol which detects packet loss and attempt recovery. Furthermore, it monitors network latency and applies OSC timing corrections for more accurate event synchronization. Quintet.net also provides proprietary strategies which deal with network jitter and latency.

Most of the existing compositional tools allow for the authoring of traditional symbolic notation. Support for graphical notation, custom symbols, staves or extended performance techniques, is commonly achieved by the layering of raster graphics on top of rendered symbolic notation. The maximum number of parts allowed in a score is either restricted explicitly or by the available application memory. Delivery of larger instrumentation, such as a full-sized orchestra, remains a significant challenge in all networked notation systems. Scores are typically composed off-line in a proprietary data model. If required, they are converted to one of the common notation formats for sharing with other notation applications. Currently, there is no clear winner between competing symbolic notation formats such as GUIDO, JMSL or MusicXML. Real-time notation generation and distribution is well supported, however, communication between heterogeneous applications normally requires transcoding of the native data models to OSC on all participating nodes [3].

For time synchronization between network nodes notation applications typically either rely on the system clocks or regular heartbeats sent from the master node. Network time protocol (NTP) is used by default on most LANs for system clocks synchronization, and by design, can cause inaccuracies of up to 100 ms between computer clocks. The application scheduling resolution which defines a minimum time interval between two scheduled events is normally defined in the milliseconds range.

2. DISTRIBUTED MUSIC COMPOSITION AND PERFORMANCE MODEL

A distributed system consists of a number of components which communicate and coordinate actions by passing messages over a computer network. A collection of independent components appears as a single integrated system to its users. The key goals of a distributed system include transparency, openness, reliability, performance and scalability. ZScore is a distributed notation system which ultimately aims to provide the following:

- Reliable and scalable low latency messaging with guaranteed data and control message delivery where critical events are delivered and executed with humanly imperceptible latency (sub 10ms compound network and application latency)
- Accurate performance synchronization across all networked nodes which includes the effects of network latency and jitter
- Complex symbolic, graphical or algorithmic notation authoring for any instrumentation (e.g. full orchestra) and type (acoustic, digital, algorithmic etc.)
- Efficient score data encoding and segmentation strategies which minimize transcoding and avoid packet loss during network transport
- Dynamic and interactive networked notation views on heterogeneous clients which allow for automated notation update, animation, position tracking, conducting signals and gestures display, event triggering etc.
- Real-time capture of a conductor or a musician's gesture and integration with the score and performance flow

A distributed composition and performance model enables heterogeneous components to interact over a message-oriented middleware (Figure 1).

2.1 Message-Oriented Middleware

High-throughput and low-latency messages are typically delivered over a messaging middleware which isolates application developers from the low level networking implementation detail and provides scalable and reliable message delivery. Message-Oriented Middleware (MOM) is a software or hardware infrastructure that provides message

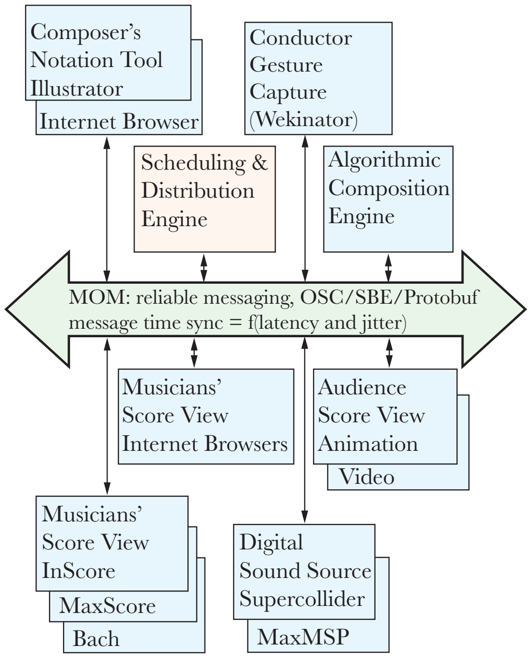


Figure 1: A distributed composition and performance system over message-oriented middleware (MOM).

delivery between distributed system components. Examples of the messaging middleware in music notation systems include Odot [9] and Landini [10]. Both are built on top of OSC protocol but offer different functionality: Odot is conceived as a framework for flexible inter-application communications, while Landini implements a reliable messaging protocol and offers improved time synchronization which takes network latency and jitter into account.

A common approach to make components with disparate data models communicate with each other is to build adapters which sit between the MOM client and each component. The role of an adapter is to translate the network data format (e.g. OSC) to the component's native data model. Language and platform neutral data serialization mechanisms such as Google Protocol Buffers (protobuf) [1] can make this process more flexible and streamlined. Humanly readable protobuf data structure definitions can be relatively easily imported and reused in other participating components. The component message processing should ideally be zero-copy and use preallocated, non-blocking data structures with minimal thread switching.

A music application message data model needs to be designed to allow for efficient data segmentation into chunks of up to 1500 bytes, which is the standard network Maximum Transmission Unit (MTU) size. This avoids data fragmentation during network transport and reduces the chances of packet loss and reordering.

2.2 Reliable UDP Multicast

Multicast is a network data routing method where a message is sent to a group of nodes from one or more sources. It can be described as one-to-many or many-to-many rout-

[1] https://developers.google.com/protocol-buffers

ing. Client nodes have to subscribe to a Multicast group to receive messages. Multicast protocols are almost always UDP and can be implemented at the application level. However, it is preferable to use network assisted delivery with Multicast enabled hardware routers where the application publishes a single message and the network router sends a copy of the message to all Multicast group members. This can significantly reduce network bandwidth usage and the sender's CPU load.

By default, UDP protocol does not guarantee message delivery. To resolve this issue, a number of reliable UDP based Multicast protocols have been proposed and implemented (PGM, TRDP, LBT-RM etc.). Unlike the TCP protocol, which sends an acknowledgment for each received packet, most of the reliable UDP Multicast protocols track ordered messages and only send a negative acknowledgment (NAK) if they detect a missing message, which is a much more efficient solution.

Several high-throughput, low-latency middleware libraries which provide reliable NAK based UDP Multicast have recently been released under the Open Source license. These libraries, such as Aeron [2] , are designed to deliver millions of messages per second (at 40 bytes per message benchmark) with microsecond latencies. Aeron operates at OSI layer 4 (Transport) and can be thought of as a TCP replacement. Internally, it uses Simple Binary Encoding (SBE) so it would need transcoding to OSC where required. As OSC operates at OSI Layer 6 (Presentation), it would naturally fit as a layer on top of Aaron.

A fully scalable and reliable distributed music notation system should ideally incorporate the concepts mentioned above and provide a middleware-like network layer abstraction for delivery of OSC (or similarly) encoded messages over a reliable NAK based UDP protocol. The choice of UDP unicast or multicast protocol should depend on the message type and routing mode (one-to-one or one-to-many). The proposed middleware implementation should also synchronize the timing of message execution on each network node based on network latency and jitter.

2.3 Precise Network Time Synchronization

Precision Time Protocol (PTP) is similar to the NTP clock synchronization protocol mentioned above. However, on LAN networks, PTP can achieve sub-microsecond accuracy which is much more acceptable for networked notation software. The drawback is that PTP is not available by default on most computers and needs to be installed and configured on all network nodes. Open Source implementations PTPd and ptpd2is are available for all Unix-like systems which includes OS X. Alternatively, there are numerous PTP enabled hardware routers and switches which can be used on LAN as master clocks.

For Internet wide performance, the most accurate master clock that can be used is GPS time signal which has a theoretical accuracy of 14 nanoseconds. An example of music system synchronization over a GPS signal is The Global Metronome project [11]. It demonstrated that the combination of GPS for the master clock and NTP for LAN synchronization can produce sub-millisecond network node clock offsets. The main issue with The Global Metronome is that it requires access to GPS signal and, therefore, a clear view of the sky. The most convenient solution to this problem is to place The Global Metronome externally, link it via Ethernet cables to a LAN within a performance venue, and use NTP to synchronize nodes on the network.

In many cases it might be more practical to synchronize notation and event execution over a network in a tempo-relative rather than absolute time. In the simplest of scenarios, the master application instance would send synchronization events at regular intervals (e.g. every 96th of a whole note) to all participating nodes in order to set their internal tempo-relative position. However, excessive synchronization events may cause network saturation, so it would be more optimal to send master synchronization events at a lower resolution (e.g. every beat or a bar) and rely on network node system clocks for more granular scheduling and synchronization. To achieve acceptable synchronization accuracy, this approach would need network latency tracking per node and event timing adjustments similar to the Landini [10] implementation.

2.4 SVG-based Score Representation

A composition data format produced by the score authoring tool needs to contain enough information to enable notation rendering tools to reliably reproduce the intended score layout and perform any time related operations. Semantic data models, such as GUIDO or MusicXML, define both spatial and temporal context for notation rendering. The problem with semantic representations is that both the score authoring tool and all participating notation rendering clients need to fully support the composer's intended notational style. Due to a vast variety of contemporary composition styles, extended playing techniques and many contemporary composers' intentional disregard of standardization, it would be very hard to create a generic yet comprehensive semantic representation solution. As a result, composers and performers are increasingly turning towards systems which allow for constraint-free, graphical notation representations.

Computers can process graphical information either in raster or vector form. Raster format defines actual pixel values that need to be displayed on the computer screen. It is therefore fast to render, however, the file size can grow significantly for higher resolution images, which is not ideal for real-time network transport. Raster format also does not scale optimally, for example, downscaling can cause visible quality loss and it cannot be easily modified. Vector graphics on the other hand define relative x and y positions, paths and attributes such as color, thickness, fill, shape and curve which need to be interpreted by the host application. Therefore, it is slower to render but much more flexible to modify and scale. Scalable Vector Graphics (SVG) is an XML-based vector image format with support for interactivity and animations. It is an open standard supported by all major Internet browsers and many other graphical applications.

[2] https://github.com/real-logic/aeron

As demonstrated by Gottfried [12], SVG format can be successfully transcoded to OSC and this work has been integrated into Odot library. Adobe recently announced its support for Node.js which could allow for the network integration of their SVG authoring tool, Adobe Illustrator. SVG can be relatively easily extended with musical context such as the score element hierarchy and temporal information. The addition of time-space mapping allows for programmable synchronization and easier integration with notation rendering software such as InScore. Similar to computer font distribution, composers can create customized SVG symbol libraries which can then be distributed to networked clients and referenced in real-time scores. In this way, the amount of data that needs to be transferred in real-time can be significantly reduced.

Recently proposed music notation markup standards MNX and related GMNX (MNX-generic) [13] incorporate linking and time-space mapping of SVG graphics. Eventual adoption of these standards will allow for rendering and synchronization of SVG notation in standard Internet browsers.

2.5 Dynamic Notation View Design

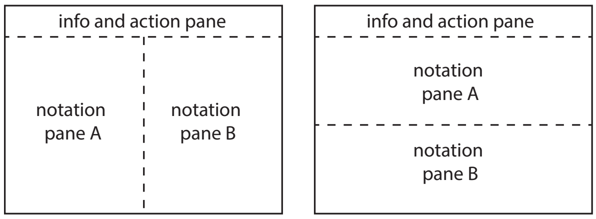
Unlike static notation, the dynamic view requires a carefully thought out refresh strategy which does not interfere with the currently played notation, providing enough time and space for musicians to prepare for the upcoming material. The refresh strategy needs to take into account network and rendering latency and ensure that notation updates do not interfere with the score continuity.

A good dynamic score front end design should fully utilize available screen real estate and provide a clear view of the notation, available actions, and any additional information musicians should be aware of during a performance. If delivery to heterogeneous platforms is required, the notation should be legible when scaled to any of the common screen aspect ratios (4:3, 16:9 and 16:10), screen sizes (10 to 17in) and resolutions (1024x768 to 2880x1800). As most of the common laptop types can only be used in the horizontal screen orientation, it is preferable to optimize notation layout for the horizontal screen viewing.

Dynamic scores with linear stave notation typically either use the full page or stave update as in Richard Hoadleys Calder's Violin where the entire view is replaced with new notation at once, or the continuous scroll as in Cat Hopes Longing and Luciano Azzigottis Spam where the notation moves continually from left to right. These strategies are suitable for particular score types. The full page refresh strategy does not provide much preparation time for musicians, especially where performance continuity is required at fast tempos. The continuous scroll strategy requires musicians to focus on a fixed point on the screen where the notation crosses a vertical synchronization line, thus reducing their capacity to look ahead and prepare for upcoming changes. It also requires continuous notation availability so it is not ideal for generative or free-timing scores.

2.6 Alternating Pane Layout

The alternating pane notation strategy aims to resolve dynamic notation update issues by providing familiar left-to-right and top-to-bottom reading direction and ample preparation time to musicians. The notation is stationary while several animated objects are transposed on top, indicating tempo, current position during performance and conducting gestures. Furthermore, it defines a clear time window for upcoming notation generation and transport. Figure 2 shows the main sections of the alternating pane layout for a full score and an instrumental part. In both cases, the notation view is divided into three main areas (panes). The top pane contains information about the score (title, part name, server status etc.), actionable buttons (for interaction with the server or other peers) and signaling information (tempo and start indicators). The main area, which takes approximately 80% of the screen real estate, is split into two notation panes, A and B. Each of the notation panes display an equivalent of a full score page or a single part stave. The notation is read left to right and top to bottom, the same as with static paper scores.



(a) Full Score (b) Part

Figure 2: Alternating pane layout

At any point of time during a performance, there is always one active and one preparatory pane. At the very start, pane A is active and pane B is preparatory. When the notation content in pane A is completed, pane B becomes active and pane A preparatory. Once the musician's focus is firmly moved to pane B, pane A is updated with the upcoming notation which is scheduled to be performed after pane B notation content is completed. The dynamic update process then continues in a similar fashion, following an ABAB... sequence.

2.6.1 *Time restrictions and allowances*

There are several time restrictions that should be taken into account when working with alternating notation panes. The notation to be played after the active pane notation is completed needs to be generated, transferred and rendered in the preparation pane by the time the active pane notation is around half way through its duration (T/2). This is to allow for performance continuity and preparation time for musicians. Furthermore, the preparation pane notation should only be refreshed once the active pane notation is played for an appropriate time duration (T1, e.g. longer than one beat) to allow for the musicians' focus switch. This means that the minimum time window for notation preparation (generation, network transport, rendering etc.) is from the active focus switch time (T1) to the active pane half duration time (T/2). If, for example, the composition tempo is 120 bpm and the active notation pane contains 5 bars with 4/4 time signature, then the minimum notation preparation time window is 4.5 seconds (T1 = 0.5s, T/2 = 5s). In most cases this would be sufficient time for the network transfer of graphical stave files or generation of algorithmic notation. This also creates clear timeline rules for the real-time notation generation and display when using alternating pane layout.

3. ZSCORE CURRENT STATE

ZScore is a distributed networked notation system which aims to satisfy requirements and implement the solutions outlined in previous sections. Currently, it is a collection of third-party and custom-made software. Composition authoring is done in Adobe Illustrator extended with the new set of JavaScript plugins. A proprietary Java engine was developed for score distribution and synchronization over a network, while InScore stand-alone clients are used for dynamic score rendering. The video "Composition for Networked Ensembles" [3] explains ZScore's main features and user trials with Moscow Contemporary Music Ensemble in March 2017.

3.1 Time-space mapping and synchronization

ZScore utilizes tempo-relative time synchronization described in section 2.3 which takes advantage of InScore's time-space mapping functionality. In this mode, the master server application sends regular synchronization messages carrying the global tempo-relative position to all clients. Each InScore client runs its internal clock and can synchronize independently if given a tempo and time-space mapping configuration. The master synchronization events effectively override internal client clocks with the global tempo-relative position and therefore ensure common time-space positions across the clients. The synchronization message frequency can be selected per composition and its choice depends on tempos and rhythmical structures used in the score.

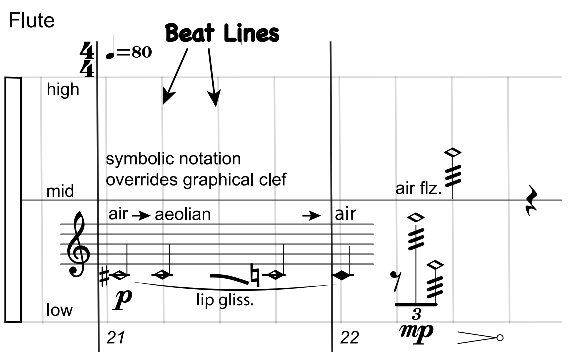


Figure 3: An excerpt from "Ukodus" flute part with visible Beat Lines used for time-space mapping.

The concepts of a Beat Line (BL, Figure 3) and Beat Division Unit (BDU) were introduced for easier time-space mapping and event scheduling workflows. The BDU value can be set to any fraction of a whole note (e.g. 1/8 which is equivalent to a quaver duration) and represents the lowest time resolution available for event scheduling and synchronization. The current minimum BDU value is 1/96. Beat Lines coincide with the bar beat onsets and contain information about their spatial and time position. The BL spatial position is set in terms of x and y coordinates on the score page while their time position is expressed in a number of BDU units from the composition start. Beat Lines are a form of proportional notation, however, there are no restrictions regarding consecutive spatial Beat Line positioning so they can be set individually to suite the score notation density. The time interval between Beat Lines is measured in BDU units. For example, if the BDU value is set to 1/8 then in a 4/4 bar each time interval between Beat Lines is 2 BDU units and in 5/8 bar with the beat division of (3 + 2)/8, the first beat consists of 3 and the second beat of 2 BDU units. Beat Line spatial and time position is exported with the score data and is used in InScore client for space-time synchronization.

[3] https://youtu.be/ioqNP4qg6JQ

3.2 Score Authoring

A vector graphics editor, Adobe Illustrator, is used for composition authoring at present. It allows for the unconstrained positioning of any notation type; export of SVG and multiple raster formats; import and creation of user defined symbol libraries; and is scriptable, which opens a range of opportunities for functional extensions and potential real-time network integration. Illustrator does not provide any musical context by default, therefore, a number of improvements have been implemented for more efficient music composition flows and integration with the networked software.

3.2.1 *Hierarchical Layer Structure*

Inspired by Gottfried [12], a hierarchical layer structure was created to provide a musical context in Illustrator and enable the automation of score creation and export. The hierarchical layer elements can contain one or more child layers (Figure 4). Currently, a Part layer has a one-to-one

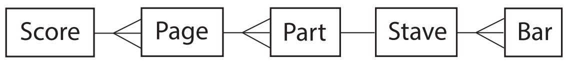


Figure 4: Hierarchical score layer entity relationships.

relationship to a Stave layer which contains all the graphical data required to display an instrument staff. The Bar layer contains all the graphical data required to render the notation and logical data required for synchronization such as tempo, time signature and beat line positions. The notation layer contains all the symbolic or graphical data required to display bar notation and can contain arbitrary notation types. The Illustrator layer structure is displayed in the screen capture in Figure 5.

3.2.2 *SVG Symbol library*

To accelerate symbolic notation generation, a set of custom symbols based on open-source LilyPond notation font were imported into Illustrator. Due to the flexibility of a

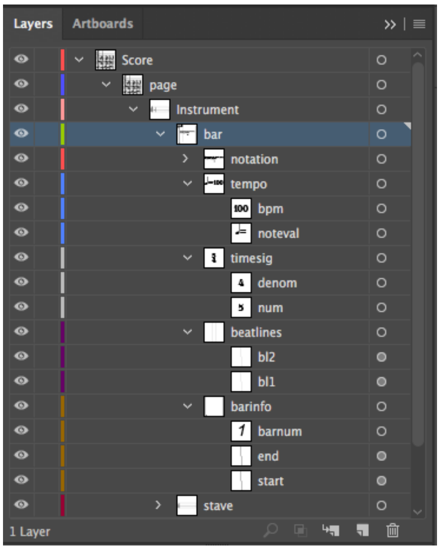


Figure 5: Adobe Illustrator ZScore layer structure.

vector graphic editor, it was straight forward to extend the library with custom symbols, such as different note head types and sizes and instrument fingering charts etc. (Figure 6). For variable length continuous lines, such as crescendo and decrescendo markings, a set of brushes were created and imported into Illustrator. An example of mixed symbolic and graphical notation created in Adobe Illustrator with the help of ZScore tools plugin and music symbol libraries is displayed in Figure 8.

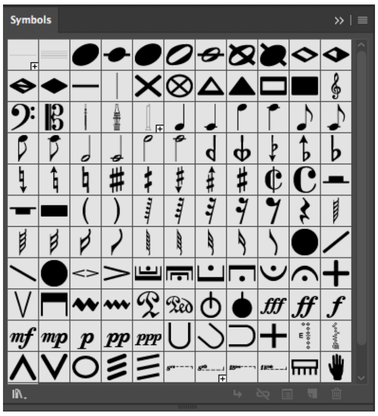


Figure 6: Notation symbol library imported and extended in Adobe Illustrator.

3.2.3 *ZScore Tools JavaScript plugin*

The set of JavaScript plugins were developed to speed up composition workflows and automate score export. Currently ZScore Tools includes: Layer, Page, Bar and Export plugins (Figure 7). The Layer plugin allows for Illustrator layer structure definition, editing, XML import/export and copying between one or more scores. Similarly, Page and Bar plugins help create required pages and bars at specified locations including any meter or tempo changes and Beat Line positioning. Export plugin allows for the export of the full score and parts in SVG or PNG graphical formats. In

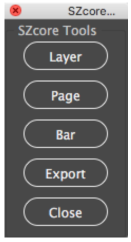


Figure 7: ZScore Tools plugins for Adobe Illustrator .

order to provide accurate and automated space-time mapping, the export process also creates necessary data for InScore in the required format:

$$([X_{start}, X_{end}[[Y_{start}, Y_{end}[)([BDU_{start}, BDU_{end}[)$$

where X and Y are space coordinates and BDU is the number of units since the beginning of the piece. The exported values define two-dimensional rectangles between two Beat Lines and the corresponding start / end tempo-relative time. Additionally, the export process automatically collates score meta data required by the score scheduling engine. This information about time signatures, tempo changes, other score events, BL positions and related BDU values is currently stored in a csv (comma separated values) formatted file. An example of a score page authored in Adobe Illustrator is displayed in Figure 8.

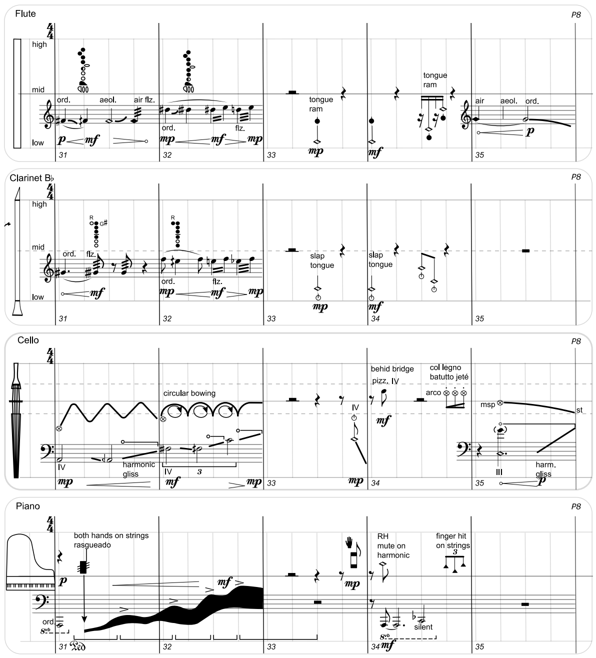


Figure 8: An example of the score page created in Adobe Illustrator demonstrating mixed clef and notation styles.

3.3 Distribution and scheduling engine

The central hub in charge of scheduling and distributing the score data over a network in real-time is the ZScore server and management client written in Java programming language. ZScore network management client (Figure 9) can import and parse score definition data exported from the ZScore Tools Illustrator plugin and submit to the server, creating an internal representation of the score metadata. The internal score metadata model mimics layer hierarchy shown in Figure 4.

The server listens to notation client connections and sends information about available parts to all connected clients. When musicians select individual parts on their notation client, the server associates the selected score part with the client's host address. From then on, all messages related to a particular part will be routed to the associated client host.

The server internal scheduling resolution is 1 millisecond with a maximum measured deviation of 0.8 milliseconds. It translates local absolute time to a tempo-relative value expressed in BDU units and evaluates any scheduled score events accordingly. The server supports multiple transports with different meters and tempos which allow for compositions and performances of polyrhythmic and polymetric scores. Events can be preloaded in score data or dynamically added during a performance according to the timing rules discussed in chapter 2.6.1. ZScore's real-time functionality is bounded by the timing rules, so any notation generated during the performance needs to be available in the time window defined by T1 and T/2 boundaries. The current server implementation utilizes LMAX Disruptor [4] which allows for high throughput lock-free data processing with microsecond latencies. At present time, OSC messages are delivered over UDP unicast.

The network management client (Figure 9) can load and start the score from any position defined in terms of the Page, Bar and Beat number on all participating networked clients. It also allows for tempo multiplication in the range from 0.1 to 2.0 for rehearsal purposes. Tempo can also be dynamically modified during the performance.

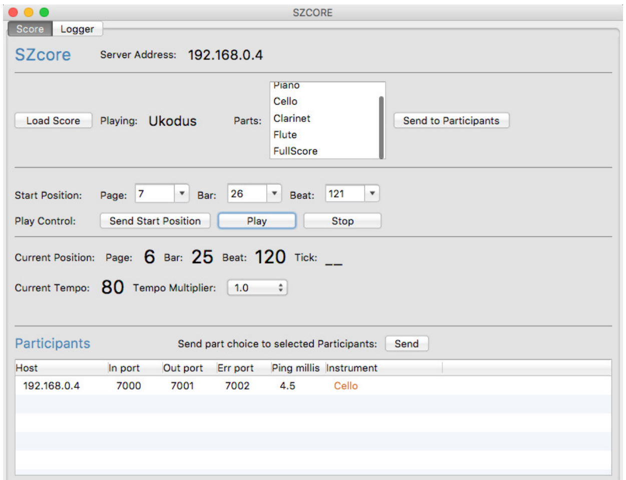


Figure 9: ZScore distribution and scheduling engine management client screen shot.

3.4 Dynamic notation rendering

InScore is currently used for dynamic notation rendering due to its networking capabilities, native OSC support, time-space mapping, built-in interactivity and scripting engine.

It also supports multiple graphical file formats, although its SVG support depends on the underlying Qt library so some features such as symbol referencing via the xref attribute were not available at the time of the writing. To work around this, all score pages were exported and distributed in PNG raster format.

Once the ZScore startup file is opened, all communication with the server can be done directly from InScore client. The start-up file contains configuration for the alternating pane layout and the set of JavaScript functions which handle all interactive tasks. An example of the dynamic score with alternating pane layout is presented in Figure 10. The active notation pane is highlighted with red borders while the preparatory pane is slightly dimmed. Several features aiming to replace some of the conducting gestures have been added. The signalling traffic light at the top left corner aims to draw the attention of the musicians and provide an indication of the starting tempo. The animated position line, visible as the light green line on the upbeat leading to bar 28 (Figure 10), indicates the current real-time position within the score. It also has the attached tempo indicator ball which is visible as the red circle on top of the stave in Figure 10. For ease of orientation, the actual starting position is marked with a light purple line (on the first beat of bar 28 in Figure 10). The starting position can be set to any Beat Line from the network management client (Figure 9). When the score performance is started in the network management client (Figure 9), the traffic light signal flashes in the starting tempo frequency and the position line starts moving from left to right, indicating the current position on the screen. The attached tempo indicator ball starts simulating conductor signals with vertical movements calculated from the simplified pendulum motion formula where the ictus plane is at the top of the stave. The current position line always starts from the upbeat before the selected starting position in order to mimic the familiar conductor gesture on start. When the position line reaches the penultimate beat of the active pane stave, the preparatory pane current position line will start from the upbeat at the same time to provide notation view continuity.

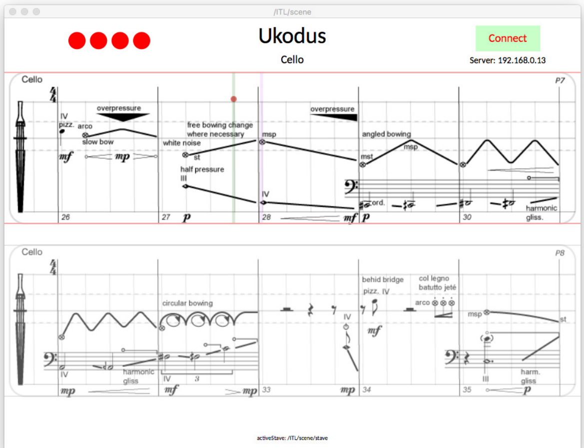


Figure 10: InScore view of the cello part dynamic notation in alternating pane layout.

[4] https://lmax-exchange.github.io/disruptor/

3.5 User trials

In March 2017, a trial session was held with the Moscow Contemporary Music Ensemble at Goldsmiths, University of London to test ZScore's technical functionality and user experience in a workshop situation. [5] The piece Ukodus for flute, clarinet, cello and piano quartet was composed by Slavko Zagorac for this occasion. A combination of laptops and tablets running both OS X and Windows operating systems were used for the musicians' front ends while the Java scheduling engine and the management client were hosted on OS X laptop. The musicians' devices were networked over the dedicated tri-band wireless hardware router while the scheduling engine laptop was connected directly to the router over the ethernet cable. The maximum round trip latency recorded during the workshop was 12 ms. Apart from some intermittent instability of the notation clients on older Windows OS versions, there were no significant technical issues during the performance.

The system setup was relatively time consuming as all musicians' devices needed software installation and WiFi network configuration as well as the initial functional testing. Due to the familiarity of the notation layout, these highly skilled musicians were able to quickly grasp the technical aspects of the system and perform the entire piece without stopping on their first sight-reading attempt. The mixed clef staves and spatial layout of the notation did not require any additional explanations, nor did they present any particular problems during the performance. The musicians feedback on the alternating pane layout usability and overall system performance was positive and encouraging.

4. CONCLUSIONS AND FUTURE WORK

This successful user trial of the current ZScore implementation has reinforced the case for SVG-based complex notation representation and the alternating pane layout dynamic notation view design. The downsides of the proposed approach are SVG authoring complexity and the considerable development effort required to enable distributed system interoperability. The ZScore Tools Adobe Illustrator JavaScript plugins have significantly accelerated the SVG authoring process while further planned automation should make the composition process even more streamlined. The distributed system complexity can be encapsulated on the server side and within the client API implementation, thereby simplifying the end user experience. Future ZScore development plans include the reliable UDP multicast middleware integration and SVG-based notation authoring and rendering in standard Internet browsers, thus enabling the utilization of any mobile device without any additional software installation. The planned publication of ZScore OSC API for score data scheduling and distribution is expected to encourage collaboration and open the system to third-party software integration. The machine learning software Wekinator [6] integration would enable conductor gesture capture and mapping to score events, thereby allowing for humanized real-time tempo and dynamic changes. The existing beat tracker could be enhanced to provide a much richer representation of the conductor gestures on musicians' screens through animation of its velocity, shape and color. Similarly, direct audience participation and integration with the score decision logic during the performance may be achieved through proprietary audience score views on mobile devices. This would allow not only composers, but also conductors, performers and audience members to significantly impact the composition flow and even create new compositional material through algorithmic functions. The expectation is that these technical innovations may lead to new creative possibilities in music composition and performance.

[5] https://youtu.be/ioqNP4qg6JQ

[6] http://www.wekinator.org/

5. REFERENCES

[1] D. Fober, Y. Orlarey, S. Letz, and F. Bevilacqua, "Programming Interactive Music Scores with INScore," in *Proceedings of the Sound and Music Computing conference (SMC'13)*, Stockholm, Sweden, 2013, pp. 185–190.

[2] G. Hajdu, "Quintet.net: An Environment for Composing and Performing Music on the Internet," *Leonardo Music Journal*, vol. 38, pp. 23–30, 2005.

[3] S. James, C. Hope, L. Vickery, A. Wyatt, B. Carey, X. Fu, and G. Hajdu, "Establishing connectivity between the existing networked music notation packages Quintet.net, Decibel ScorePlayer and MaxScore," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017, pp. 171–181.

[4] G. Hajdu and N. Didkovsky, "MaxScore – Current State of the Art," in *Proceedings of the International Computer Music Conference (ICMC'12)*, Ljubljana, Slovenia, 2012, pp. 156–162.

[5] A. Agostini and D. Ghisi, "Bach: An environment for computer-aided composition in Max," in *Proceedings of the International Computer Music Conference (ICMC'12)*, Ljubljana, Slovenia, 2012, pp. 373–378.

[6] C. Hope, L. Vickery, A. Wyatt, and S. James, "The Decibel ScorePlayer – a digital tool for reading graphic notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015, pp. 59–69.

[7] R. Constanzo, "Dfscore: networked notation software." [Online]. Available: http://www.rodrigoconstanzo.com/dfscore/

[8] G. Burloiu, A. Cont, and C. Poncelet, "A Visual Framework For Dynamic Mixed Music Notation," *Journal of New Music Research*, vol. 46, no. 1, pp. 54–73, 2017.

[9] J. MacCallum, R. Gottfried, I. Rostovtsev, J. Bresson, and A. Freed, "Dynamic Message-Oriented Middleware with Open Sound Control and Odot," in *Proceed-*

ings of the International Computer Music Conference (ICMC'15), Denton, TX, USA, 2015, pp. 58–65.

[10] J. Narveson and D. Trueman, "LANdini: a networking utility for wireless LAN-based laptop ensembles," in *Proceedings of the Sound and Music Computing conference (SMC'13)*, Stockholm, Sweden, 2013, pp. 309–316.

[11] R. Oda and R. Fiebrink, "The Global Metronome: Absolute Tempo Sync For Networked Musical Performance," in *International Conference on New Interfaces for Musical Expression (NIME'16)*, Brisbane, Australia, 2016, pp. 26–31.

[12] R. Gottfried, "SVG to OSC Transcoding: Towards A Platform for Notational Praxis and Electronic Performance," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015, pp. 155–162.

[13] W3C Music Notation Community Group, "MNX Draft Specification." [Online]. Available: https://w3c.github.io/mnx/

TOWARDS A NOTATION FOR TRUMPET VALVE ROTATION

Dan O'Connor
Edith Cowan University
dan@danoconnormusic.com

Lindsay Vickery
Edith Cowan University
l.vickery@ecu.edu.au

ABSTRACT

Many contemporary performers and composers seek new sounds through extension of traditional instrument techniques. For the trumpet one such extended technique is valve rotation, the rotation of a trumpet piston valve within its casing affecting the timbral complexity of airstream effects. This paper describes the development of a system for notating valve rotation using a prescriptive graphical language and an animated interface for entering continuous rotation and airstream data.

1. INTRODUCTION

Valve rotation is an extended technique unique to the trumpet and other valved brass instruments. The technique employs the rotation of the valves within the valve casings, directing the air that is moving through the trumpet in unconventional ways and altering the timbre and complexity of airstream effects.

Although the possibilities of the technique have been apparent since the invention of the valve, the historical origins of valve rotation in performance are not clear. Extended techniques such as flutter tonguing, growling, half-valving and lip bends, have a long, documented history in recording and, more recently, notated composition. However, valve rotation appears to have remained relatively unexplored perhaps because its relatively quiet, detailed and granular texture is more suited to amplified performance, recording, personal listening and the microsound, lower-case aesthetics of the Post-Cage/post-Walkman era.

Trumpeters Craig Pedersen [1], Nate Wooley [2] and Axel Dörner [3] all currently include valve rotation amongst their extended techniques, suggesting that its use in performance originated in non-idiomatic and free improvisational contexts. While notated solo works for trumpet by Stockhausen [4], Gruber [5] and Turnage [6] employ a wide range of extended techniques, including airstream effects, multiphonics and slide removal, the only notated composition involving valve rotation uncovered by this study was Rama Gottfried's *speckle* [7] (Figure 1). Gottfried provides a basis for the depiction of the trumpet valve block in composition and the rotation of valves.

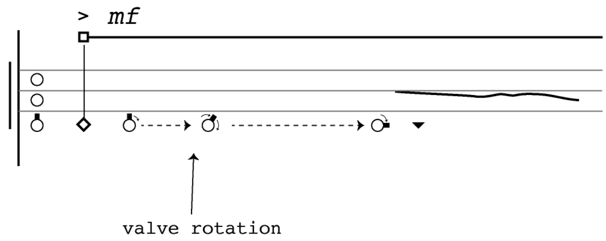


Figure 1. Excerpt of trumpet valve notation from *speckle* [8] notates a 90-degree valve rotation, clockwise.

Currently valve rotation is a difficult technique for performers and composers to communicate due to limited:

- documentation referring to the sounds created by valve rotation;
- investigation into the application of valve rotation in improvisation and composition; and
- methods for the effective notation of the technique.

This study is a starting point in the exploration into trumpet valve rotation and in particular the communication of this technique through the development of an animated score creator/player software application: Valverotator. This paper outlines the concurrent development of a valve rotation notation through a practice-based approach involving improvised experiments.

2. GRAPHIC NOTATION

Due to the difficulty of describing the nuances of sound made with the valve rotation technique, notating the actions required to generate the sound is more practical. Kojs [8] defines this approach, found in works such as Cage's *Variations III* [9], Kagel's *Pas de Cinq* [10], Berio's *Sequenza V* [11] and Lachenmann's *Pression* [12], as action-based. In developing this action-based notation the intuitive nature of the symbols was of utmost importance. Vickery, et al., note the importance of "semantic soundness—the degree to which the graphical representation makes intuitive sense to the reader—rather than necessitating learning and memorisation of new symbols" [13].

2.1 Evolving through practice

The initial impetus to notate valve rotation was born of the need to document interesting sounds that occurred during improvisation with the technique. This lead to the preparation of the valve buttons to include marks as a visual representation of rotational position — achieved by marking the valves with a permanent marker pen. When an interesting sound was discovered by O'Connor the orientation of these marks was transcribed onto paper — *Set and Forget* (Figure 2, left) is an example of this.

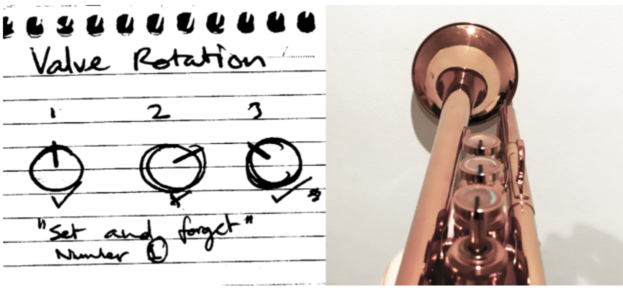


Figure 2. Score for O'Connor's *Set and Forget* (left) and the performers perspective of the trumpet (right). The authors note the difference in orientation of the valves and valve notation.

It became apparent that the orientation of the symbols in Figure 2 was not particularly intuitive, as the horizontal placement of the valve graphics did not reflect the perspective of the trumpet player (Figure 2, right), in which the first valve is at the bottom of the frame and the third at the top. Due to the angle from which the valve buttons are viewed, it is also easier to read the valve marks when they are directed back towards the player, which led to selecting this direction as the neutral, unrotated position.

The notation in *Progression Sketch #1* (Figure 3) takes the perspective of the trumpet player into account, orienting the valve symbols vertically and the rotation marks toward the player when unrotated. The 'ticks and crosses' in *Set and Forget* indicating the removal of valve slides, have become squares — a filled square meaning slide in place (filled), and slide removed (unfilled). The ticks and crosses possessed established meanings that were unhelpful, or even confusing when used in this way.

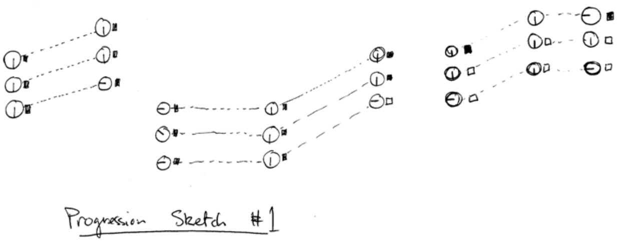


Figure 3. Excerpt from O'Connor's *Progression Sketch #1,* read left to right with dotted lines instructing when to blow air.

Progression Sketch #1 attempts to choreograph valve rotation and air velocity, with the aim of creating a piece solely involving valve rotation. In *Progression Sketch #1* the dotted lines indicated that air is to be blown through the instrument, and the height of the valve cluster graphic on the page, the velocity of that airstream. The sketch is played left to right and duration is proportional to the spacing of the valve cluster graphics. Experiments with *Progression Sketches #1,* and (not illustrated) *#2* and *#3* identified the more challenging parameters to represent in this notation — duration of events, direction of rotation, and air velocity.

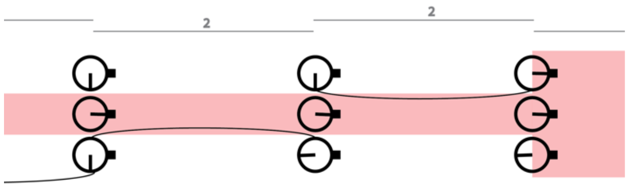


Figure 4. Excerpt 1 from *Valverotator Test Score* 2 by O'Connor notates 1st valve rotation, 90 degrees clockwise, then 3rd valve, 90 degrees anticlockwise.

Valverotator Test Score 2 (Figure 4) was designed in Adobe Illustrator and presents solutions for notating the rotation and air velocity parameters. In this score, airstream velocity is indicated by the depth of the red block of colour. When the air velocity block encompasses the whole valve cluster graphic, the air velocity is at its maximum, if it is a very thin red line a very slow air velocity is required, and if no red block is present then the performer does not blow through the instrument. The score also indicates the direction of valve rotation via a line with an upward arch (clockwise rotation) or downward arch (anticlockwise rotation) attached to the top of a valve diagram. Above the valve cluster graphics are the durational indicators, both geometrically proportional and with a duration in seconds inscribed above.

In order to choreograph the removal of valve slides the small black squares symbolising the slides are detached from their corresponding valve. The valve slide choreography is via a dotted line pre-empting the slide removal (Figure 5), warning the trumpet player that by the next frame the slide should be removed.

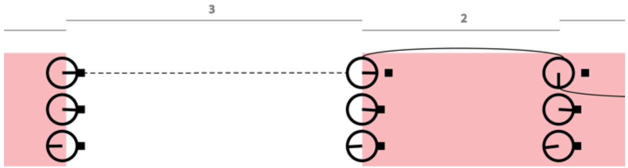


Figure 5. Excerpt 2 from *Valverotator test score 2* by O'Connor shows 3rd valve slide detached notation.

In Figure 5 the third slide graphic appears detached in the second valve cluster, instructing the performer to remove the third slide from the instrument. Slide removal can be slightly clumsy whilst moving air through the instrument. In *Valverotator Test Score 2* all the slide movements are undertaken without the flow of air through the instrument, to avoid unintended timbral variation.

Having developed a set of symbols that allow for the transcription of valve rotational position, therefore facilitating composition with the valve rotation technique, the question then arises; how does one create and present a composition? Two methods, the static score and the animated score, were considered.

3. THE SCORE

3.1 Static score

In the history of notated music the creation of a static score, often on paper, is the predominant form of presenting a composition; "The paper-based technology of CPN [common practice notation] has remained almost unchanged for 400 years" [13]. The advantages of the static score are:

- Accessibility due to the lack of necessity for technical equipment required to perform the work.
- The value of the aesthetic nature of an arrangement of symbols on a page.
- The ease of discussion and education — everything is potentially visible at all times and thus can be referred to efficiently.

Valverotator Test Score 2 is an example of a static score for valve rotation. In O'Connor's practice he found these scores both playable and aesthetically pleasing. There are some deficiencies in the static score; the time-consuming nature of graphically composing the score and potential issues of precise ensemble synchronisation.

In order to find a more efficient means of composition for valve rotation within a medium in which multiple scores can be precisely synchronised, a software application (app) for composition and performance of valve rotation scores was developed.

3.2 Animated notation

Animated notation offered the possibility of more precisely specifying the degree and rate of valve rotation and the potential to bundle other specifications such as airflow and detachment of slides from their corresponding valve. When seeking models for an animated notation for trumpet valve rotation Ryan Ross Smith's *Study No. 8* [14] provided a starting point. The use of animated rotating dial-like objects to indicate percussive actions for performers in *Study No. 8* (Figure 6 left) provided a constructive analog to valve rotation that was similar to the graphic representation of the trumpet valves O'Connor had devised for static scoring (Figure 6 right).

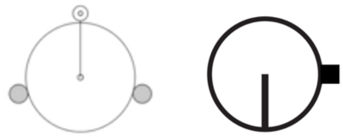


Figure 6. Smith's [14] percussion notation (left) provided a starting point for O'Connor' valve button symbol (right).

Smith's 2015 paper, "An Atomic Approach To Animated Music Notation" [15] also provided a useful touchstone for the conceptual development of the app. The terminology proposed by Smith is used to describe the graphical language that was adopted:

- Primitive – an irreducible static or dynamic symbol;
- Compound primitive — Two or more primitives seamlessly combined in such a way that a secondary primitive enhances or embellishes the primary;
- Structure – two or more primitives in some interrelated relationship;
- Aggregate – a collection of primitives, structures, and their respective dynamisms that corresponds to a single player; and
- Intersection – a dynamic attack cursor intersecting a static node or playhead.

3.3 Software considerations

A number of platforms for the software development were considered. Decibel ScorePlayer was first considered due to the recent development of 'Canvas' mode [16] which can accommodate scrolling and stationary objects simultaneously. Unfortunately, the ability to rotate score segments was not achievable in Canvas mode during this study. A recent score, *Southern Currents* by Meg Travers,[1] does employ a rotating playhead in the ScorePlayer environment and may allow further consideration of ScorePlayer in the future.

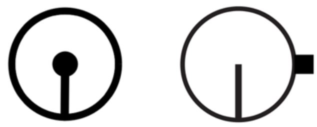


Figure 7. Max 6 dial object's (left) similarity to O'Connor's valve button symbol (right) lead to development of a Max 6 app.

The standard Dial in Max 6 (Figure 7, left) bears a resemblance to the valve graphic representation O'Connor developed for static scoring and quickly became the

[1] *Southern Currents*, by Meg Travers: concert performance at Perth Institute of Contemporary Arts, Perth, Australia, 24th October 2017.

platform for the development of the animated notation for valve rotation.

3.4 Development in Max 6

The initial versions of Valverotator employed Max 6 function objects that the composer could graphically program to input the degree of rotation at desired points in the composition. When pressing play this information would then be fed into the Dial objects, rotating them continuously. Each Dial had a corresponding function object and the velocity of air was directed by a fourth function object that controlled the opacity of the background colour of the Dials.

In practical trials the immediacy of rotational movement was noted as problematic in that there was no forewarning to an upcoming rotational gesture. In discussing 'contact' in animated music notation Smith [15] notes the 'setup' before a point of contact and the ability of the setup to convey performance instructions. To build on Smith's example, the conductor's baton falls (the 'setup'), stopping at an invisible boundary (the 'contact') to denote the instant of the downbeat. If the falling of the baton did not precede contact with the invisible boundary the performer would not have the necessary information to decipher this as the instant of the downbeat. To translate this to the dynamic valve rotation notation, there needed to be an analogous 'setup' before the motion of a dial in order to convey performance instructions to the performer, giving them forewarning of the rotational gesture. The concept of the setup is perhaps even more pertinent to the exhalation of air through the instrument, which of course requires a preparatory inhalation.

Similar issues had been resolved by scrolling notation from right to left across the screen and actualising them at a playhead — a point of contact at which the instructed gesture is to be actualised [17]. It is the influence of this scrolling score, playhead relationship that manifested the animation of the function objects within the Valverotator app.

The function objects contain graphically visible 'x, y' data points connected by a line, and from version three onwards of the Valverotator app the function objects themselves scroll from right to left into a playhead. Due to the close proximity of the valve and slide compound primitive to the left side of the playhead the scrolling objects are terminated at the playhead, so as to avoid cluttering of the other information presented.

The scrolling function objects bundle three pieces of information for the performer:

- Direction of rotation via the direction of slope of the line;
- Relative speed of rotation via the gradient of the line; and
- Instant of the actualisation via the contact point of function line and playhead.

This seems to be all the information the performer would need to execute the gesture, however the scrolling function objects lack 'semantic soundness' in this scenario, decoding of rotational information from scrolling line graphs being unintuitive. Valverotator combines a playhead, valve symbol, and scrolling function object for each valve, a more intuitive and fully descriptive structure (Figure 8). In rehearsal O'Connor found keeping focus on the valve diagrams and the scrolling information in his periphery was most effective.

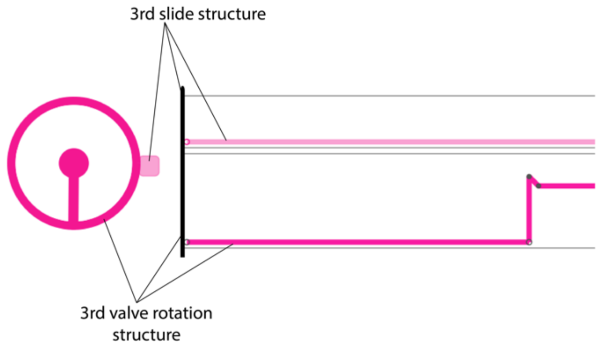


Figure 8. Screenshot of the Valverotator 3rd valve aggregate, a combination of the 3rd slide and 3rd valve rotation structures.

Smith's term 'intersection' [15] lends itself to discussion of the animated notation of the trumpet valve slide. Valverotator 3 employs a Max 6 multislider bar oriented vertically to communicate air velocity (Figure 9). In practice this is intuitive because the minimum and maximum air velocities are clear at all times, the full multislider bar and empty multislider bar respectively.

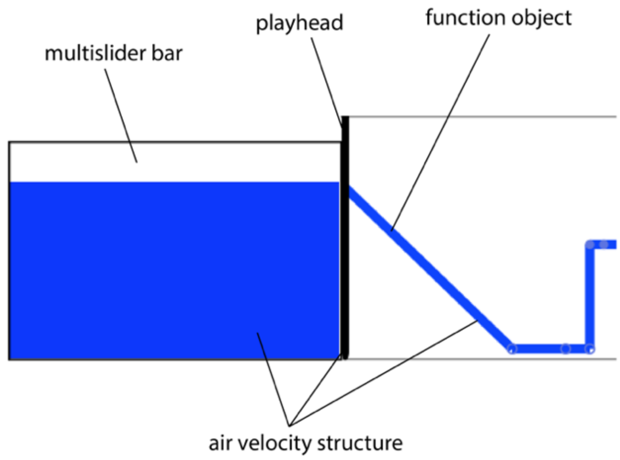


Figure 9. Screenshot of the air velocity aggregate made from multislider, playhead and function object.

Figure 10 shows the valve slide geometric primitive, simply a square. It is not the slide primitive's shape that is important in transferring performative instruction but its intersection with the valve button primitive. As the scrolling slide function object contacts the playhead the slide primitive detaches from the valve primitive (Figure 10 right). It is this state of attachment or detachment that is intuitively decoded by the performer.

Colour is used to visually separate aggregates. Each aggregate is formed from identical primitives to control the same parameters for each valve and air velocity, thus the use of colour differentiates the streams of information, preventing ambiguity in the decoding process.

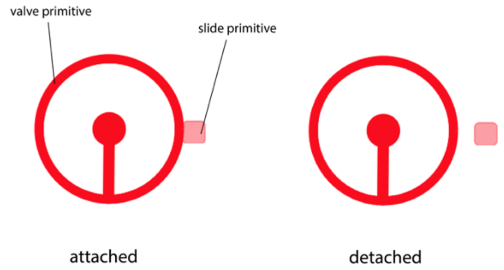


Figure 10. Slide primitive in attached (left) and detached (right) positions.

The specific colours were intuitively selected to contrast with one another and the white background. The valve slide compound primitive is subtly different in opacity to its relative valve rotation compound primitive (Figure 10).

3.5 Composing in Valverotator

Figure 11 shows the screen when the Valverotator app first opens. The composer must first input a total duration for the piece. Next the composer simply draws onto the function objects, via a sequence of x, y coordinates, a line representing temporal changes in each parameter under the trumpet players control — valve rotation, slide position and air velocity. Figure 12 displays a complete score. The piece can then be played by pressing ‘space bar’, paused by pressing ‘enter’ and reset to the start by pressing ‘esc’.

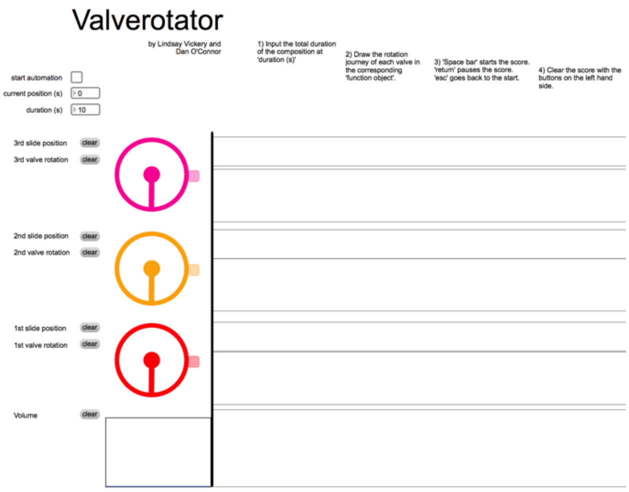


Figure 11. Valverotator app opening screen, before composition input.

As mentioned earlier the Valverotator app bypasses the use of image creation software to notate composition and facilitates fast turnaround from idea to score. Furthermore, the ability to make small adjustments or additions with minimal disruption to the entirety of the score is an advantage. Over the course of the research three scores were created with Valverotator app, two being translations of static scores and the third composed entirely within the Valverotator app. It was noted by O’Connor during the composition process that “the magic of Valverotator is the immediacy with which one can compose”.

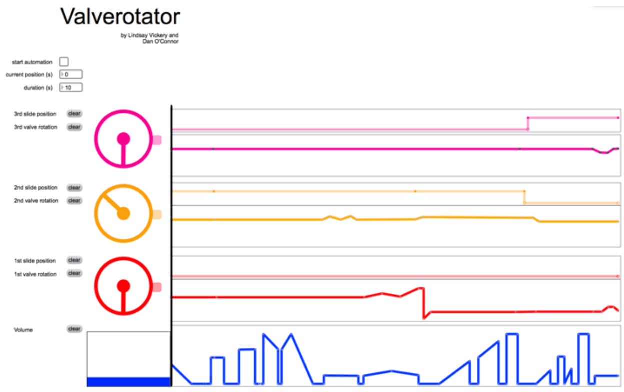


Figure 12. Screenshot of completed composition in Valverotator.

3.6 Distribution and performance

Valverotator, the composition app, is a Max patch that can be distributed to be run within Max 6 installed on any computer. There is also a standalone OS X app that requires no additional software to run. It has been successfully tested in OS X El Capitan 10.11.6.

When considering the completion, distribution and performance of scores, the robust and universal nature of the delivery format is critical. Thus, in the course of this research Quicktime’s ‘screen record’ function has also been used to capture, as a video, the animated score for distribution and performance of Valverotator scores. The plethora of devices available to performers at this time mean that the playback of video is easily within the grasp of most.

4. CONCLUSION

At present Valverotator works effectively as a fast and efficient way to craft a score, though continued development and refinement are necessary. The following challenges are yet to be addressed:

- Addition of numerical readout in degrees of rotation when placing a point anywhere on the function object in order to increase accuracy when composing.
- Relocation of the air velocity multislider to place it at the centre of the performers focus.
- Increased codification of the slide compound primitive. Can degrees of extension be informed by degrees of slide compound primitive movement? — rather than just the binary, attached or detached, movement currently employed.
- Currently manipulation of total composition duration affects individual event duration. Separation of these

parameters would allow greater compositional freedom.

- Inclusion of directions for tongue position, posture and valve depression — techniques that O'Connor found complementary to valve rotation during improvised performance.
- Compatibility with current versions of Max. Creation of a jsui object is in progress for Max 7 (and potentially 8) versions of the software as a replacement for the Max 6 dial.
- Consideration of graphical human interface (GUI) with regard to the alignment of function objects and corresponding multislider object.

Further research and development will extend Valverotator to involve these techniques. In order to do this the transition or incorporation of the Decibel ScorePlayer or similar system may be necessary.

Another avenue for future development is the extension of the valve rotation technique to ensembles with brass instruments capable of valve rotation. Currently the performance of multiple scores in video format could be synchronised using software such as Multivid. The Decibel ScorePlayer also has networking capability for performance in this way.

It is hoped that this paper is a starting point for the discovery and use of valve rotation by composers and performers alike. In developing a unique notation for trumpet valve rotation performers and composer now have a communication tool with which they can discuss valve rotation and create new work. The notated form is by no means universally codified and the continued assessment and development by third parties is welcomed by the authors — hopefully creative people will take this notation, develop and refine it, and create interesting music.

5. REFERENCES

[1] C. Pedersen, "wind noise with space like eight," on *Cut and Gouge* [CD]. Canada: Mystery and Wonder, 2015. Track 2.

[2] K.J. Reilly, *Charmaine Lee, Nate Wooley @ Gallery 456 11-20-16 1/3*, 2016. [Online]. Available: https://www.youtube.com/watch?v=H0OHEAthpNQ

[3] T. Sossi, *Kevin Drumm and Axel Dorner live in Brugherio (Milan)*, 2000. [Online]. Available: https://www.youtube.com/watch?v=icXg13ZKaGI, 2017.

[4] K. Stockhausen, *Oberlippentanz fur trompete solo*. Hamburg: Acanta, p1985.

[5] H.K. Gruber, "Exposed Throat for solo trumpet" "[Recorded by Håkan Hardenberger], on *Exposed throat* [CD], BIS, 2006. Track 1.

[6] M-A. Turnage, From the wreckage, on *Speranza & From the wreckage* [CD]. London Symphony Orchestra, 2013. Track 1.

[7] R. Gottfried, *speckle*, 2016. [Online]. Available: http://www.ramagottfried.com/scores/speckle_score_transposed.pdf

[8] J. Kojs, "Notating action-based music," *Leonardo Music Journal*, vol. 21, pp. 65–72, 2011.

[9] J. Cage, *Variations III*. New York: Henmar Press, 1962.

[10] M. Kagel, *Pas de Cinq*. London: Universal Edition, 1965.

[11] L. Berio, *Sequenza V*. London: Universal Edition, 1966.

[12] H. Lachenmann, *Pression*. Wiesbaden: Breitkopf & Härtel, 1968.

[13] L. Vickery, L. Devenish, C. Hope, and S. James, "Expanded percussion notation in recent works by Cat Hope, Stuart James and Lindsay Vickery," *Contemporary Music Review*, vol. 36, no. 1-2, pp. 15–35, 2017.

[14] R. R. Smith, *Study no. 8 [15 percussionists]*, 2012. [On-line] Available: http://animatednotation.com/RyanRossSmith.html

[15] R. R. Smith, "An atomic approach to animated music notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015.

[16] S. James, C. Hope, L. Vickery, A. Wyatt, B. Carey, X. Fu, and G. Hajdu, "Establishing connectivity between the existing networked music notation packages Quintet.net, Decibel ScorePlayer and MaxScore," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017.

[17] L. Vickery, "The Evolution of Notational Innovations from the Mobile Score to the Screen Score", *Organised Sound*, vol. 17, no. 4, pp. 128–36, 2012.

GIOQOSO, AN ON-LINE QUALITY ASSESSMENT TOOL FOR MUSIC NOTATION

Francesco Foscarin
CNAM, Paris
francesco.foscarin@lecnam.net

David Fiala
CESR, Tours
david.fiala@univ-tours.fr

Florent Jacquemard
INRIA, Paris
florent.jacquemard@inria.fr

Philippe Rigaux
CNAM, Paris
philippe.rigaux@lecnam.net

Virginie Thion
IRISA, Rennes
virginie.thion@irisa.fr

ABSTRACT

Quality is a daily concern to everyone involved in the production of digitized scores. We propose an on-line interface devoted to music notation, freely accessible to the community, intended to help users to assess the quality of a score thanks to a combination of automatic and interactive tools. This interface analyzes a score supplied in MusicXML or MEI, and reports quality problems evaluated with respect to a taxonomy of quality rules. We expose the motivation, describe the interface, and present the methodology.

1. INTRODUCTION

It is a common and shared experience that achieving high quality standards for digitized scores is quite difficult and currently requires a lot of time and efforts devoted to inspect the score rendering and detect mistakes. The difficulty of this task is due to the complex semiology of music notation. The issue is particularly sensible in the context of collaborative editing, since each contributor is free to use her own engraving software, and to communicate with others via some XML format, typically MusicXML [1], sometimes MEI [2] [1], and probably in a near future the W3C Music Notation format [2].

Unfortunately, these XML-based encodings are extremely permissive, and allow for all kinds of problems regarding correctness, consistency and completeness. This can be understood if we consider that they have to adapt to the wide flexibility and variability of music notation throughout ages. This is also probably unavoidable, given the complexity of rules that can hardly be expressed as constraints in the document's schema. As a result, music score encoding a quite error-prone, and currently requires a careful revision by human experts as part of the publishing process.

[1] http://music-encoding.org
[2] https://www.w3.org/community/music-notation/

1.1 Evaluating a score

Quality evaluation is commonly done by a combination of audio and visual inspections. Given the high semiologic complexity of music notation, this evaluation cannot be fully reliable, and would highly benefit from the assistance of automatic tools. What makes things even worse is that even if a score has been checked visually by several people who spent hours to inspect every detail, this does not guarantee that the underlying encoding is correct. Let us take two simple and concrete examples:

1. *Lyrics encoding*. The association of text and music obeys some complex rules. Lyrics are decomposed in syllables, and, at the graphical level, syllables from a same word are linked by dashes, melismas are indicated by underscores, *etc.* People engraving music have to be aware that a correct encoding has to distinguish the syllables from the metadata that describes how they are interrelated and linked to the music. We already found many examples where both aspects are glued, because the engraver directly encodes continuation symbols in the text itself. As a consequence, although not directly visible, the score encoding becomes faulty: the text cannot be cleanly extracted or searched, and some notes in melismas are not properly attached to syllables.

2. *Layers encoding*: many music pieces are organized as a combination of layers, and in order to make sense of these pieces, it is important that the engraver identifies the layers content and carefully reflects them in the encoding. Unfortunately, many engravers loosely use the layer concepts in engraving softwares for tricking the visual rendering, losing the internal music structures.

We can cite many other examples where an apparently correct score, at least when printed or rendered on a screen, turns out to be wrongly encoded internally: slurs instead of ties, title or composer entered as raw text, and not as metadata, *etc.* This results in unexpected distortions when another renderer is used, and makes the music representation unsuitable for other usages: analysis, audio/score alignments, or production of alternative representations (Braille for instance).

1.2 Defining and measuring quality

Defining and measuring the quality of a score encoding is not easy. First, there is no universal definition of what a "correct" score encoding is: it highly depends on the music itself on one hand, and on the score usage on the other hand.

Second, many abstraction levels can be considered, and many granularities. Some aspects are purely syntactic (do all slurs have a start/end point? Are all measures exactly filled?), other pertain to metadata, which may or may not be mandatory (title, composer, date, copyright). Some aspects are specific to the score layout (symbol overlapping, appropriate position of clefs and staves). And, of course, the music content itself has to be correct and should faithfully reflect the source and editors choices. The latter is probably the most difficult part to assess with an automatic evaluation, although we can imagine to check if the material is consistent with the style and expected idiomatic features.

All these points have to be simultaneously taken into account by a proof-reader. As explained above, visual inspection is both unreliable and insufficient, in particular if we are keen to ensure an accurate representation of the score content, apt at being exploited in other contexts that the mere printing of the music sheet. Controlling manually the encoding itself is not really an option, even assisted by advanced editors – A single inspection of a large XML file should be enough to be convinced that nothing can reliably done at this level. What we need is a holistic approach that combines visual and audio evaluation with an automatic inspection of the encoding to report potential quality issues.

1.3 Our approach

We propose a tool that attempts to provide in a single interface all the components that participate to a score evaluation, and makes this evaluation automatic as much as possible. This tool is publicly available online [3] and can be used by anyone to evaluate an XML-encoded score (MusicXML or MEI) as soon as the document can be retrieved from some URL. The implementation is, and will continue to be, in progress, because the list of quality rules that can be envisaged is potentially endless. However, we believe that the foundations of our method are now established, and that the main functionalities of the user interface are operational. We therefore submit to the TENOR community the current status of our work. The main contributions are:

1. A taxonomy of quality rules that relies in particular on a distinction between the concepts of *score content* and *score engraving*. This distinction was proposed in one of our earlier works [3] as a necessary step to make sense of the heterogeneous information gathered in digitized scores. It is used as the foundation of a hierarchical presentation of quality aspects which, in our opinion, helps the end user to organize her evaluation.

[3] http://neuma.huma-num.fr/quality

2. An implementation of representative quality rules for each of the main categories of our taxonomy. We describe a sample of indicators to illustrate their specific features.

3. Last but not least, an integration of the methodology in the GIOQOSO public Web interface.

For the sake of concreteness, we start with a description of the user interface in Section 2. Section 3 explains the foundations of our digitized scores quality model. We examine our taxonomy and some representative examples in Section 4 and conclude the paper in Section 5.

2. THE GIOQOSO ONLINE INTERFACE

Figure 1 shows the current status of the GIOQOSO tool[3]. GIOQOSO is integrated in the NEUMA Digital Score Library [4], but is an independent component that can be used to analyze any XML score accessible at a public URL.

2.1 Importing and displaying the score

Figure 1 illustrates how we import a score coming from the *Lost Voices* CESR project [4]. The score has to be encoded either in MusicXML or in MEI. However, when the input is in MusicXML, an internal conversion is operated first to obtain an MEI encoding that enjoys two major advantages in our context.

1. Each element of the score (notes, rests, slurs, measures, staves, *etc.*) has a unique id. This is is essential to *annotate* this element with some semantic label, in our case, a quality indicator. For instance a note can be annotated with a *missing lyrics* indicator, or a measure with a *incomplete duration* indicator.

   MusicXML, unfortunately, does not offer this ability to refer to score elements. This is one of the main new features that will be incorporated in the forthcoming W3C recommendation. [5]

2. A second advantage of the MEI encoding is that it comes with several analysis and interactive tools. We use in particular the Verovio toolkit [6] [5] to display and interact with the score. Verovio relies on a conversion from MEI to SVG that preserves the id of elements. As a result, an annotation (*i.e.*, some meaning attached to a note or a measure) can be graphically displayed as a decoration of the corresponding SVG element.

The ability to play a MIDI rendering of a score, possibly starting from any note, is also a Verovio feature. This functionality corresponds to the standard "Play" option proposed by all score engravers, and is the quite useful tools when it comes to check the content of a score.

[4] http://www.digitalduchemin.org
[5] https://www.w3.org/community/music-notation/
[6] http://verovio.org

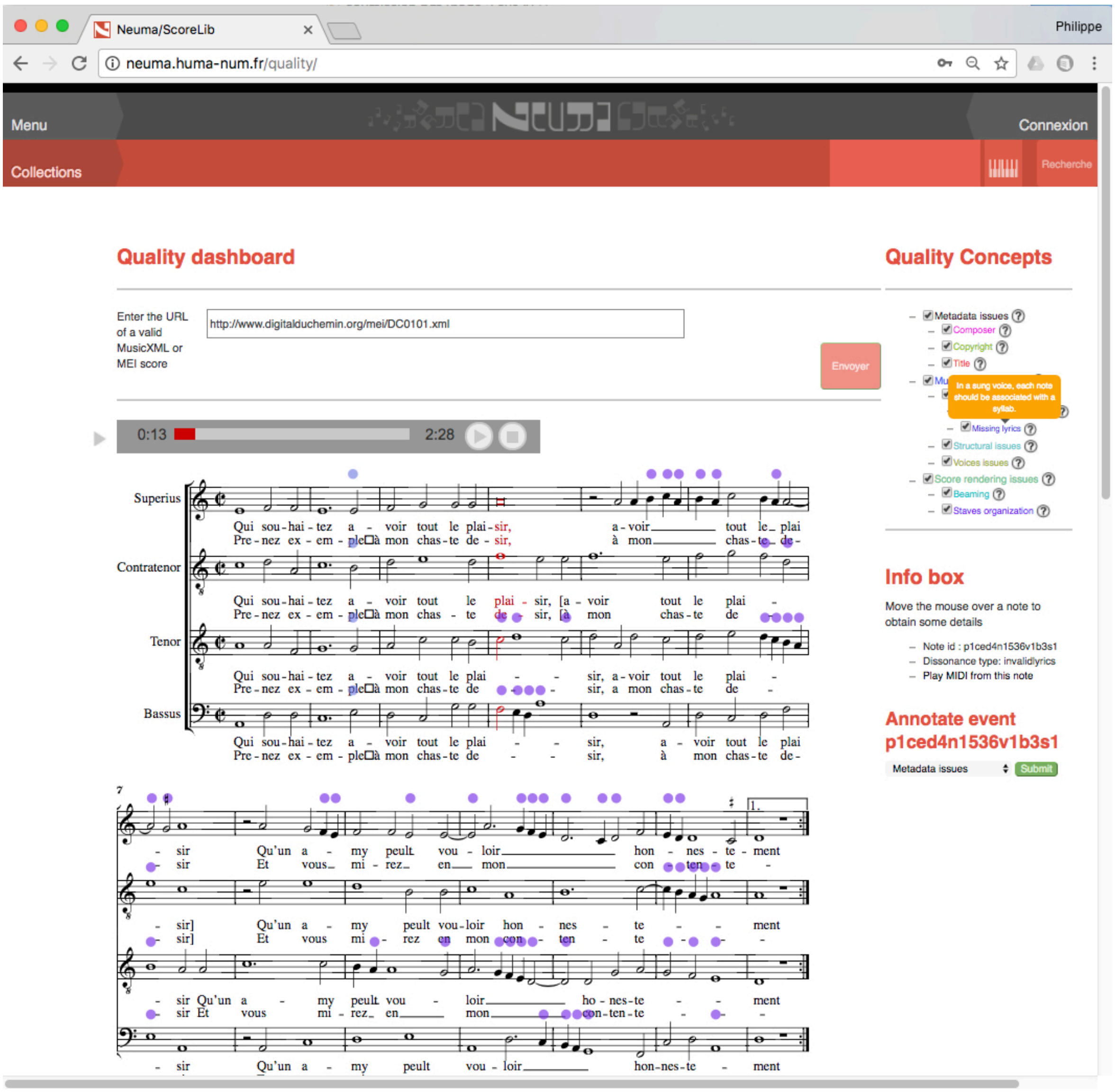


Figure 1. The GIOQOSO User Interface

2.2 Showing/hiding quality annotations

The document is analyzed on-the-fly in order to complete it with quality annotations. Each annotation is an instance of a quality indicator, and the indicators themselves are organized as a forest, displayed in the top-right part of the user interface (see also Section 4).

The taxonomy of the quality model is extensible. We add new rules regularly, based on input from our scientific experts (the CESR and IReMus musicology labs), on best notational practice found in reference sources on score rendering/engraving, *e.g.* [6], and on mere exploration of various online score libraries that reveal many encoding and rendering issues.

In the interface, each indicator comes with a description that can be highlighted by dragging the mouse over its name (the orange rectangle in Figure 1, column 'quality concepts'). Every annotation is displayed as a small colored circle above the elements or groups of elements that constitute the annotated fragment. Its color characterizes a specific quality indicator. The user can hide/show a set of annotations by clicking on any level of the model tree. This makes convenient to focus on a particular aspect, or to ignore altogether some indicators if they are deemed irrelevant.

2.3 Interactions

Finally, actions can be undertaken by the user. Each annotation can be inspected in detail by clicking on it. The *Info box* part of the interface then displays details on the related score elements, and on their annotations (there might be many). A form is also proposed to report an annotation error, or to complete existing annotations. Such inputs might become quite useful in the future in order to include user feedback in the context of a large collaborative system.

Note that, since the score is loaded from its remote location, the user can directly correct the identified issue on her local version. It suffices then to reload GIOQOSO to trigger a new evaluation of the quality rules that will hopefully show that some formerly identified quality issues have been fixed. GIOQOSO can therefore be seen as a complementary tool closely and easily integrated to the user's score production environment. The only requirement is that the score under production is accessible at a fixed URL.

3. MODELING DIGITIZED SCORES QUALITY

Our model of rules for notational quality follows a conceptual view of score of score production that distinguishes three steps (Figure 2).

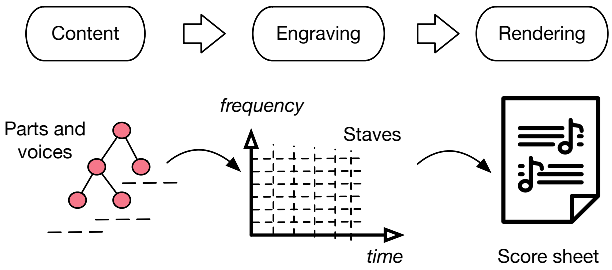


Figure 2. The workflow of (digitized) score production

1. *Score content modelling*. This part covers all aspects related to what we call the *score content*, independently from any encoding or rendering concern. Essentially, it captures the *structural* organization of a score in parts and *streams* [7], and the description of streams as time-dependent elements.
2. *Score engraving*. Score engraving denotes the mapping of the score content into a set of staves. We model a staff as a grid covering a restricted range in the space of frequencies, and the mapping associates a content with a 2D (frequency, time) space.
3. *Score rendering*. The final steps take a score content, score engraving specifications, and produces a layout of score based on the properties of a specific media (paper, screen, *etc*).

We believe that this distinction is extremely useful to identify and characterize the specific quality issues that can occur at each step, and to determine how we can evaluate and possibly fix these issues.

First, clearly, the last step (score rendering) depends on the rendering software and on the properties of the displaying media. Therefore, we consider this part as out of scope for the score quality evaluation process: a high quality score can be displayed very badly with a poor renderer or on a tiny screen.

This leaves us with the distinction between *score content* and *score engraving*. We think that it makes sense for exactly the same reasons that led to separate the content of web pages (structured in HTML) from their display features (defined with CSS rules) [7]. Defining the *content* of a score, and evaluating its quality, is a data modelling and representation task. It requires the definition of the structure of a score, and the specification of constraints on instances of this structure. On the other hand, *engraving* is a process that applies to a score content, and defines the relationships between this content and a 2-dimensional space organized with respect to a temporal dimension (abcissa) and a frequency dimension (ordinate). Evaluating the engraving quality implies to take into account both the content and the mapping.

[7] The metaphor also holds for the *rendering step*, carried out in the case of HTML by a Web browser that adjust the textual content and CSS rules to the displaying window.

3.1 The score content model

The "score content" focuses on the aspects of a digital score representation that describe the intended production of sounds, and is independent from any visualization concern. If we assume an ideal music performer, the content is the part of the score that contains the sufficient and necessary information to produce the intended music. In order to decide whether a piece of data belongs or not to the content, we just have to wonder whether it is likely to influence this music production. A MIDI player is a possible candidate, but we actually require a more sophisticated performer model, apt at taking account for instance of the meter to infer strong and weak beats.

In an earlier work, we proposed a *notation ontology*, called MUSICNOTE [8], to model this content [8]. Essentially, a score is modeled as a *hierarchical structure*, where leaves consist of *streams*, and inner nodes of *parts*. A stream is a sequence of *events*, which can belong to several subclasses. Let us explain the structural aspect first by taking as an illustration the sketch of a piano concerto score (Figure 3).

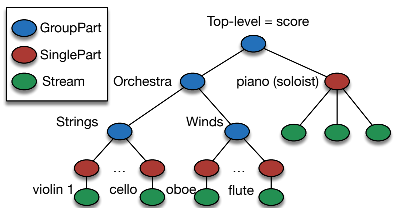


Figure 3. Structure of a score

The score is made of *parts*, where the concept of part is is refined into two sub-concepts. A *group* (of parts) consists of a set of subparts, and mostly serves the organizational aspect of the score. For instance, the orchestral material of a concerto score typically defines a group for wind instruments, another one for string instruments, *etc*. A *single part* encapsulates the music events assigned to an individual performer (instrument or vocal). Figure 3 shows for instance a single part for the soloist (piano), another one for the violins, cellos, *etc*. The informations related to measures (in particular time signatures) are represented at this level. A single part contains one or several *streams*.

Streams are objets where music content, as time-dependent production of sounds, is actually described, as illustrated by Figure 4. A *stream* is essentially a time series of *events*, where an event denotes the production of a sound artifact

[8] http://cedric.cnam.fr/isid/ontologies/files/MusicNote.html

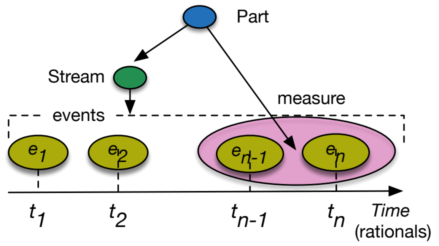


Figure 4. Stream as a time series of events

at a specific timestamp (the "onset"). Particular cases of events are notes and chords (with pitch and duration information), textual contents, or information on dynamics and articulation.

The quality issues that related to the score content concept are therefore organized with respect to the above ontology [9] .

3.2 The score engraving model

A score is a graphical artifact that represents some music content according to two dimensions:

1. *Time*. This dimension is represented by the horizontal axis, and is discretized in measures, beats, and finite subdivisions of beats.

2. *Frequencies*. Sound frequencies are represented on a vertical axis, and discretized in octaves, and subdivision of octaves in (usually) 12 semi-tones.

This yields a 2-dimensional discretized space, that could be represented as a grid. In principle, a score could be fully displayed in this grid, each note being a segment whose height corresponds to its frequency, and length to the note duration. The score engraving is close to this general model, but makes some choices, motivated by practical reasons, that lead to the usual layout. First, each part (or instrument) gets its own space visualization in order to avoid the confusion that would result from the merge of several parts with similar ranges in the same layout. Second, since the range of a single instrument is usually restricted, the frequency grid allocated to this instrument is reduced to a few lines that cover this range, of *staff*. The common representation chooses to use 5 lines, and to encode the range with a clef that gives the frequency of one of those lines (*e.g.*, the second line for treble clef).

This perspective on score engraving is summarized by Figure 5. The engraving rules take a score content, determine the number of staves, allocate parts to staves, and develop the stream representation on each staff.

Our quality model relies on this perspective, and focuses on the organization of staves, their relationships, and on the inner quality of stream representation for each staff. The general question that we try to address in this context is: to which extent the content/staves mapping defined by the

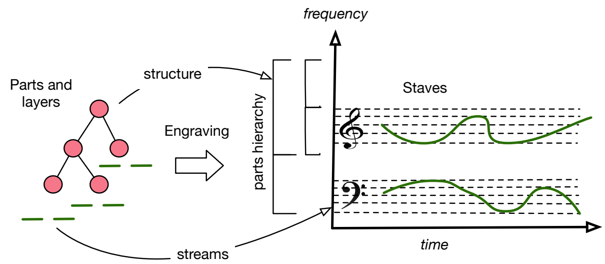


Figure 5. Engraving = mapping the content to (time, frequency) space

engraving ensures a consistent and correct layout of score? If the engraving quality is high, then we can expect that a good renderer will be able to produce a readable score display at visualization time.

3.3 Metadata

Finally, we consider a third, optional part of score encoding: metadata. Metadata is data about data, *i.e.*, in our case, any content that annotates either the score content or the score engraving. The title, subtitle, composer are metadata that annotates a score as whole. Instrument names annotates parts. There are at least two reasons to incorporate metadata issues in quality evaluation. First, metadata supplies in some cases some knowledge which is useful to measure a quality indicator. Knowing the instrument for a part allows for instance to check that the range of the music content is compatible with this instrument, or that the clef is appropriate. Second, metadata is typically a factor of inconsistencies when we consider quality concerns at a collection level. Music collection editors are eager to ensure that the level, accuracy and encoding of metadata are similar for all the scores. Although the present paper focuses on single scores, this motivates the inclusion of metadata as part of our quality model.

4. THE TAXONOMY

Based on the models introduced in the previous section, we created a taxonomy of quality indicators. The taxonomy is a forest where each tree corresponds to a "facet" of quality evaluation, and contains the related set of indicators. Currently, our taxonomy contains three such trees. It is fully accessible at http://neuma.huma-num.fr/quality/model, and partially described below. Quality indicators in boldface are detailed in the following as representative examples of the salient categories.

1. Score content issues
   (a) Structural issues
      i. Unbalanced parts
   (b) Stream issues
      i. Pitch
         A. Out of range
      ii. Rhythm
         A. **Incomplete measures**

[9] In some cases, these issues can even be formalized as rules expressed over the ontology with SWRL, the Semantic Web Rule Language [9]. We refer to [8] for a discussion on the pros and cons of a declarative approach to specify annotations semantics.

B. Overflowing measures

iii. Lyrics

A. **Missing lyrics**

B. Invalid lyrics encoding

2. Engraving issues
   (a) Staves organization
      i. Invalid staff order
      ii. Too many parts per staff
   (b) Staff parameters
      i. Invalid key signature
      ii. Invalid clef
   (c) Staff layout
      i. **Erroneous Duration**
      ii. **Unappropriate beaming**
3. Metadata issues
   (a) Missing title
   (b) Missing composer
   (c) Invalid instrument name.
   (d) ...

Note that we chose to organize our taxonomy with respect to *functional* concepts that are highly specific of the data at hand. Another possible organization would consist in considering generic quality problems [10] such as completeness, accuracy, and consistency. We believe that, in essence, quality is a multi-dimensional problem. The choice to favor the functional dimension is motivated by the need to help the user focusing on a "semantic" perspective during her inspection of the score. As such, the hierarchical organization mostly serves to navigate in the rules trees to hide/show some of them.

4.1 Score content issues

4.1.1 Structural issues

As an example of structural quality indicator, we check that all single parts have the same length. This is done by computing the sum of the durations of all the events in streams and comparing.

For this purpose, we rely on a routine of MUSIC21 [11] for extracting the duration of every event, expressed in fraction of the duration of a quarter note (quarter length). The correspondence between this duration value and the notated duration value (in term of note figures) is checked separately in GIOQOSO, see Section 4.2.2.

4.1.2 Music notation issues

At the stream level, an important property is that all the measures are correctly filled, *i.e.* that for each measure, the total duration of the events contained corresponds to the expected measure length, according to the time signature (specified in the embedding part). This is done using the same Music 21 duration event information as above.

Some issues related to lyrics quality have already be mentioned in introduction.

4.2 Score engraving issues

4.2.1 Staff parameter issues

This part of the taxonomy covers quality problems related to an incorrect or inconsistent assignment of parts to the staves system and on the parameters that dictates how the music content is rendered on a staff. The following is a list of examples that related this "functional" approach to some common quality dimensions [10].

1. *Consistency*. We check that all key signatures are consistent, including a correct transposition for transposing instruments. This is simply done by checking the key signatures encoding of all the parts in the XML document.

2. *Correctness*. The clef should be chosen to ensure that the majority of notes lies inside the staff's range (i.e., do not show a bass part on a treble claf staff).

3. *Completeness*. We check that all parts of the score are assigned to a staff, with a maximum of two parts per staff.

4.2.2 Staff layout issues

In music theory, there are precise rules for deducing actual durations from note values and meter (TS) and common practice / recommendations for writing rhythms (using in particular beams for defining nested groups), in order to improve score readability and emphasize the meter.

Digital scores *e.g.* in MusicXML usually contain rhythmic elements of different nature: features related to score content, like time signature and actual note durations, and features related to engraving content, like note symbols and beams. Despite their strong relationship, these elements can be presented independently in documents. This redundancy can be source of inconsistency in rhythm notation.

Let us give below some details about the procedures proposed in GIOQOSO for assessing the quality indicators related to the rhythmic notation in scores. That concerns in particular the consistency of the different elements represented durations and the satisfaction of some beaming conventions.

Our approach works by extracting tree structures from a score XML document and then performing verification on these trees. We consider a hierarchical model of rhythm notation inspired by the Rhythm Trees of Patchwork and OpenMusic [12, 13, 14, pages 976-978]. However, our model differs from RT in several aspects: In addition to the representation of proportional durations of notes, it also includes engraving elements related to rhythm notation (note figures, beams, *etc*). Let us describe more precisely this representation on the example in Figure 6.

Every leaf represents a note (or rest, chords...), with a label describing the note figure: `n` for simple note head, `n..` for double-dotted note head, `-n` for a note tied to the previous one... Every node is associated a duration in quarter length: The root node is associated the duration of a whole measure and every edge in the tree is labeled by the ratio between the duration of the parent node and the child

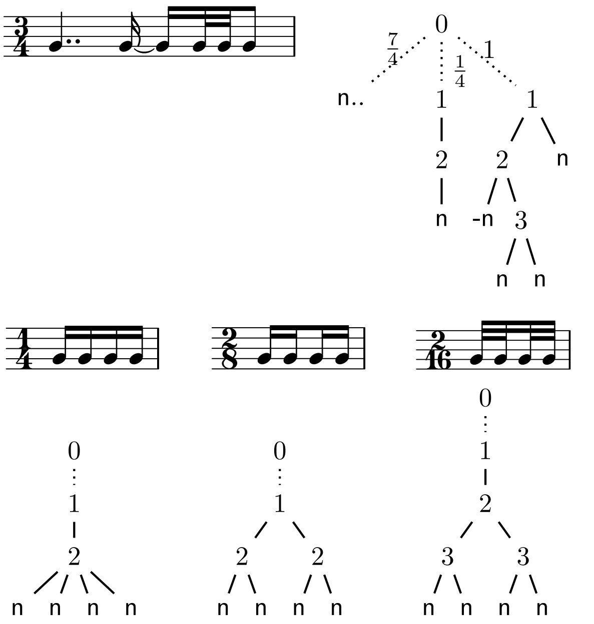


Figure 6. Beaming-tree representations of 4 measures.

node. For readability, we omit the edge labels when all sibling have same ratio. Finally, the representation of beams follows these two principles:

1. every leaf represents a note (or rest) whose number of beams is the depth minus 1 (in Figure 5, inner nodes are labeled by their depth).

2. the number of beams between two successive notes is the depth of their least common ancestor node (which is not necessarily their direct parent).

The first property holds both for isolated notes, like the two first notes in the 3/4 measure in Figure 5, and notes in groups. And, according to the property 2, two notes not connected by beams are child of the root note (see the same example 3/4 as before).

In GIOQOSO, we extract a tree as above for each measure in a digital score. The structure of trees is inferred from the beaming information and note types (as specified *e.g.* by the MusicXML element `type`) in the digital score file (MusicXML element `beam`), and the edge labels are computed from the durations given in the file (MusicXML element `duration`).

Then, several properties are checked on the trees. For instance, we check the consistency between note durations (quarter length) and the note types (a note type depends on a leaf label, a number of beams computed as above and the arity of inner nodes representing tuplets). A detected inconsistency can be seen as a critical issue in a score file, that may result in many errors when processing the score.

We also check some beaming conditions, less critical but that help the readability of the score. For instance, some position corresponding to strong beats in measures (like the third beat in a 4/4 measure) should not be crossed by beams, see [6] page 155. This can be checked using the property (2) above (in that case the depth of the least common ancestor of last note before position and first note after must be 0).

For other readability motivations, big groups of short notes are easier to read when subdivised in subgroups whose duration depends on the meter, providing that: *the number of beams separating the groups is equal to the duration of the groups they separate*, see [6]. This is illustrated in Figure 7 and can also be checked using the property (2) above. Failure when checking such properties can be signaled as recommendations in GIOQOSO.

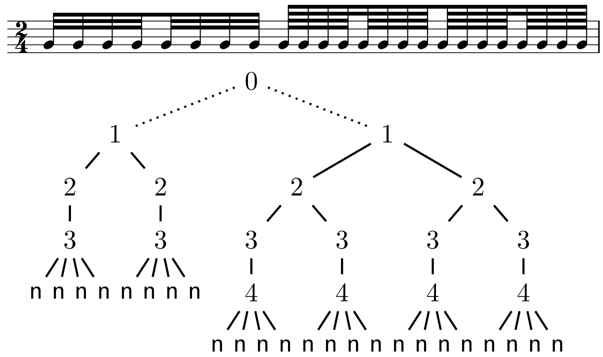


Figure 7. Groups of inner beams of various durations [6].

5. CONCLUSION

In this paper, we proposed a methodology for assessing data quality of digitized scores. We believe that the topic is important, because scores can no longer be considered as mere graphic artifacts, but as digital pieces of information. Assessing the quality of this information is essential, not only for a proper rendering on various media, but also for preserving, exchanging, and analyzing score content in all kinds of future applications.

We hope that our approach provides a ground for proof-checking score beyond graphical concerns. It requires of course several extensions in the future to fully achieve its goals. First, the list of quality indicators currently evaluated is by no means complete, and we can bet that it will never be. This is essentially harmless, this the design of our methodology makes it easily extendible. Second, we currently focus on single score inspection. In the context of collections and digital score libraries, the consistency of the encoding choices for all the scores of the collection is essential. This is particularly sensible for metadata that should be uniformly handled.

Finally, a part a score proof-reading which is basically left apart for the moment is the correctness of the content itself with respect to the source. There is no easy solution to the problem, and it appears that we will remain dependent on the user's expertise for this matter.

Acknowledgments

This work is partially supported by the GIOQOSO project [10], and the ANR MuNIR project [11].

[10] http://gioqoso.irisa.fr/
[11] http://cedric.cnam.fr/index.php/labo/projet/view?id=41

6. REFERENCES

[1] M. Good, "Musicxml for notation and analysis," in *The Virtual Score: Representation, Retrieval, Restoration*, W. B. Hewlett and E. Selfridge-Field, Eds. MIT Press, 2001, pp. 113–124.

[2] P. Rolland, "The Music Encoding Initiative (MEI)," in *Proceedings of the First International Conference on Musical Applications Using XML*, Milan, Italy, 2002, pp. 55–59.

[3] S. Cherfi, F. Hamdi, P. Rigaux, V. Thion, and N. Travers, "Formalizing Quality Rules on Music Notation – an Ontology-Based Approach," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017.

[4] P. Rigaux, L. Abrouk, H. Audéon, N. Cullot, C. Davy-Rigaux, Z. Faget, E. Gavignet, D. Gross-Amblard, A. Tacaille, and V. Thion-Goasdoué, "The design and implementation of Neuma, a collaborative Digital Scores Library - Requirements, architecture, and models," *International Journal on Digital Libraries*, vol. 12, no. 2-3, pp. 73–88, 2012.

[5] L. Pugin, R. Zitellini, and P. Roland, "Verovio: A library for Engraving MEI Music Notation into SVG," in *Proccedings: conference of the International Society for Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014, pp. 107–112.

[6] E. Gould, *Behind Bars*. Faber Music, 2011.

[7] E. Cambouropoulos, "Voice and stream: Perceptual and computational modeling of voice separation," *Music Perception: An Interdisciplinary Journal*, vol. 26, no. 1, pp. 75–94, 2008.

[8] S. Cherfi, C. Guillotel, F. Hamdi, P. Rigaux, and N. Travers, "Ontology-Based Annotation of Music Scores," in *International Conference on Knowledge Capture (K-CAP'17)*, Austin, TX, USA, 2017.

[9] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosofand, and M. Dean, "SWRL: A semantic web rule language combining OWL and RuleML," W3C Member Submission, W3C, 2004. [Online]. Available: http://www.w3.org/Submission/SWRL/

[10] L. Berti-Equille, I. Comyn-Wattiau, M. Cosquer, Z. Kedad, S. Nugier, V. Peralta, S. S. Cherfi, and V. Thion-Goasdoué, "Assessment and analysis of information quality: a multidimensional model and case studies," *International Journal of Information Quality*, vol. 2, no. 4, pp. 300–323, 2011.

[11] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *International Society for Music Information Retrieval Conference (ISMIR 2010)*, Utrecht, Netherlands, 2010.

[12] M. Laurson, "Patchwork: A visual programming language and some musical applications," Sibelius Academy, Helsinki, Tech. Rep., 1996.

[13] C. Agon, K. Haddad, and G. Assayag, "Representation and rendering of rhythm structures," in *Proceedings Second International Conference on WEB Delivering of Music*, Darmstadt, Germany, 2002.

[14] G. Mazzola, *The Topos of Music: Geometric Logic of Concepts, Theory, and Performance*. Birkhäuser, 2012.

MazurkaBL: SCORE-ALIGNED LOUDNESS, BEAT, AND EXPRESSIVE MARKINGS DATA FOR 2000 CHOPIN MAZURKA RECORDINGS

Katerina Kosta
Centre for Digital Music,
Queen Mary University of London,
London, UK
katkost@gmail.com

Oscar F. Bandtlow
School of Mathematical Sciences,
Queen Mary University of London,
London, UK
o.bandtlow@qmul.ac.uk

Elaine Chew
Centre for Digital Music,
Queen Mary University of London,
London, UK
elaine.chew@qmul.ac.uk

ABSTRACT

Large-scale analysis of expressive performance—with focus on how a performer responds to score markings—has been limited by a lack of big datasets of recordings with accurate beat and loudness information with score markings. To bridge this gap, we created the MazurkaBL dataset, a collection of score-beat positions and loudness values, with corresponding score dynamic and tempo markings for 2000 recordings of forty-four Chopin Mazurkas. MazurkaBL forms the largest annotated expressive performance dataset to date. This paper describes how the dataset was created, and variations found in the dataset. For each Mazurka, the recordings were first aligned to the score and one to another to facilitate the transfer of meticulously created manual beat annotations from one reference to all other recordings. We propose a multi-recording alignment heuristic that optimises the reference audio choice for best average alignment results. Loudness values in sones are extracted and analysed; we also provide the score position of dynamic and tempo markings. The result is a rich repository of score-aligned loudness, beat, and expressive marking data for studying expressive variations. We further discuss recent and future applications of MazurkaBL and future directions for database development.

1. INTRODUCTION

The musical score provides an incomplete representation of a composer's intended expressions for the rendering of a piece. How a performer responds to these instructions can vary widely, and has increasingly become an important area of study in recent years. However, systematic analyses of score-informed performance data has been beset by a lack of large datasets with appropriate information, such as synchronisation between performance and score information, and between performances, essential for comparing audio features and prosodic decisions along with score representation. Synchronisation is often done through beat alignment. This is particularly problematic for music with large tempo and timing deviations as current automatic beat tracking methods perform poorly for such music. Alignment between highly expressive music audio and symbolic score information is also fraught with error, requiring manual intervention. The problems are typically circumvented by manual annotation, which does not scale well to large datasets.

As a result, only a limited number of datasets exist for highly expressive music that is score-aligned and synchronised with expressive features; of these, few have large numbers of recordings of the same pieces or do so with only a handful of pieces. Table 1 shows a representative sample of such datasets, together with the expressive information layers they provide. As can be seen, there is a lack of a systematic collection of annotations for a large number of recordings that represented a range of interpretations of the same music pieces.

To bridge this gap, we created the MazurkaBL dataset, which augments 2000 recordings from the CHARM [1] Chopin Mazurka Project database with expressive information layers containing score-beat positions, loudness values, and locations and labels of score-based dynamic and tempo markings. The Mazurka Project database has been the subject and object of a few previous studies. For example, Sapp [6] created hierarchical scape plots for visual analysis of tempo and loudness similarity at multiple timescales. The dataset also provided material for testing beat tracking algorithms (eg. [7] and [8]) and for creating robust tempo-based novelty detection functions by harnessing simultaneous analyses of multiple recordings of the same piece [9].

The rationale for focusing on Chopin's Mazurkas is not only because the Mazurka dataset exists. For the majority of pianists, and indeed other instrumentalists as well, the Romantic repertoire presents a wealth of expressive possibilities [10]. The reason for indexing the recordings by score beat information and expressive markings is because the score encapsulates the composer's intentions while the recording reflects the performer's interpretation of the notated score. Each symbol—be it a note, dynamic marking, indication of articulation, or phrase grouping—can have a variety of possible interpretations. In performance studies, the original score is considered to be refracted through the performer [11, p.59], who can choose to render the symbols in unique ways. In order to understand expressivity, it is important to be able to have recordings, and hence au-

[1] http://www.charm.rhul.ac.uk/index.html

Dataset	*Description*	*No.*	*Expression annotations*
MazurkaBL	Forty-four Chopin Mazurkas (audio)	2000	beat, loudness, expr marks
Mazurka-CS [1]	Five Chopin Mazurkas (audio)	239	beat, loudness, phrase, expr marks
Magaloff Project [2]	Complete works of Chopin (Bösendorfer)	155	Midi-score alignment
Saarland Music Dataset [3]	Selected piano pieces (Disklavier, audio)	50	Midi-audio alignment, pedal
EEP Dataset [4]	String quartet movements w mocap	23	bowings
QUARTET Dataset [5]	Intonation, dynamics, phrasing exercises (audio, mocap, video)	95	bowings

Table 1: Datasets annotated with expression markings and parameters.

dio features, aligned to the score so that comparisons can be drawn between the performer's choices and the composer's notations.

The choice of annotating beats, from which one can infer tempo and timing information, and providing loudness values synchronised with notated expressive markings follows that of [1]. [2] In order to obtain reliable measurements and scalable analyses, we rely on computational audio analysis tools, which despite their imperfections are becoming standard tools for empirical musicologists [12, p. 225–233]. The large scale in which we were able to deploy the beat annotation and calculation of tempo and loudness values was made possible by a state-of-the-art audio-to-audio alignment technique [13]. Only one recording was painstakingly annotated with beat information, and that annotation transferred to all other recordings. A multi-alignment heuristic, described later in this paper, optimised the choice of reference recording for the alignment procedure.

A large dataset facilitates systematic and empirical studies aimed at understanding the range of expressive possibilities proffered by a score. It also enables the design of robust statistical models that can capture the range of possible expressive variations. A big dataset will allow scholars to discern what constitutes a typical style for the performance of a piece. Knowledge of this performance style can, in turn, constrain parameters in models of expressive performance. It also allows researchers to identify what constitutes an outlier in performance style.

The paper is organised as follows: Section 2 gives an introduction to the development of expressive notation and its use in Chopin's works, Section 3 describes the Mazurka-BL dataset, Section 4 presents the method created for obtaining score beat positions from audio recordings, Section 5 presents the method used to extract loudness information from the audio, Section 6 describes studies that have used the MazurkaBL dataset, and Section 7 offers some future directions.

2. DEVELOPMENT OF EXPRESSIVE NOTATION AND CHOPIN'S WORKS

This section gives a brief introduction to the concept of music notation, the symbolic representation of music in written form, so that it can be reproduced, as it developed in European classical music, and the development of expressive notation in Chopin's works.

The neumatic notation—from the Greek word "neuma" meaning "gesture" or "sigh"—was the system of musical notation used from the 7th to 14th century. It evolved from grave and accute accents to a system of precise indications of pitch for singing [14]. Referring to a study by Sam Barrett [15], which posits that neumatic notation is more than a memory aid, being a "reflexive tool for disciplined knowing", Cook [16, pg.11] concludes that music is "conceived platonically, as an abstract and enduring entity that is reflected in notation".

Developments to music notation as we know it today mainly involved changes on the representation of the duration and pitch of the notes that are sounded. Innovations included the development of notational symbols for different playing techniques and performance actions. Giovanni Gabrieli (1554-1612) was the first composer to specify dynamics in a score, in the *Sonata pian e forte* from the Sacrae symphoniae (1597) [17, pg. 28–29]. Annotation of dynamics, such as ***p*** for *piano*, "has remained relatively constant, although contemporary composers have explored its extremes." [14]

Next we consider the use of expressive notation in Chopin's works. Chopin's compositions can be best understood through his core inspirations, the prime one being traditional Polish music. Even in solo piano works, the dance impulse can be found in his Mazurka or Polonaise pieces [18, p.150]. [19] suggests that Chopin was influenced by late baroque and pre-classical composers; however, J. S. Bach's imprint can be found in his later works.

Searching for the characteristics that make a performance 'musical', Shaffer in [20] analyses recordings of Chopin's Prelude Op. 28 No. 8 in F# minor, examining the structural tension and the variations in tempo and dynamics to decide whether a performance "conveys an insight into the musical meaning" [20, p.184]. The combination of melodic, harmonic and rhythmic processes identify structure, while operating on different levels, interacting within and perhaps across the levels. The results of the study show the use of a phrasing gesture where there is an acceleration and increase in dynamics into a musical unit (such as a phrase) and the respective deceleration and decrease towards its boundary. Focusing on the expressive intentions that go beyond simply conveying phrase grouping, we see that related features include chord progressions, melody alterations among the phrases, and even a repeat of the same harmonisation in positions where ***ff*** and ***p*** markings appear, which helps emphasise the dynamic contrast.

[2] See also `http://mazurka.org.uk/info/excel/beat/` and `http://mazurka.org.uk/info/excel/dyn/gbdyn/` for beat and dynamic information on the Mazurka project.

In terms of dynamics, Thomas [18] refers to accents and dynamic contrasts in Mazurka pieces as emphasising the "foot-stamp or heel-clicking leap": if they are located on the first beat they may emphasise a long-breathed four-bar phrase or a short-breathed two-bar phrase. If they are located on the second bar they are usually combined with expressive harmonic or melodic stresses or with the case of having accompaniment rests on the first beat. Finally if they are located on the third bar they may either give a quiet understatement of the third movement—an example being the accompaniment rests on the first and second beat followed by a chord on the third one in Mazurka Op. 63 No.1—or emphasising the opening of a new section.

With regard to Chopin's own performed dynamics, Chopin himself preferred pianos capable for depicting refined nuances rather than ones constructed based on providing acoustic sharpness and high intensity sounds [21]. Although markings such as ***ff*** and ***fff*** appear in his works, "all his contemporaries agree in reporting that his dynamics did not exceed the degree of *forte*, without however losing a single bit of shading" [22, p.215].

Other aspects of articulation have to do with pedaling and timing. A feature found in many of the Mazurka pieces is the use of one pedal-point joining usually four-bar chord progressions which produced a "dominant fanfare" [18]. In the case of features related to timing, a characteristic of Chopin's music is that it draws inspiration from singing, which translates to a bel canto style of piano playing [23, p.216]. This style offers a strong sense of rubato by keeping a more steady rhythm with the left hand while freeing the other to push forward or hold back. Carl Mikuli, one of his pupils, "complimented Chopin's rubato for its naturalness and its 'unshakeable emotional logic'" ([24, p.91]).

3. SYNOPSIS OF THE DATASET

The MazurkaBL dataset [3] was created from 2000 selected recordings from the CHARM Mazurka dataset. The audio recordings cover a total of forty-four different Chopin Mazurkas. Table 3 shows the Chopin Mazurkas and the number of recordings of each Mazurka included in the dataset. MazurkaBL contains a table for each Mazurka in `.csv` (comma separated value) text format that includes the score beat positions (details in Section 4) in seconds per recording. Also, it contains a separate table for each Mazurka that includes the loudness information (details in Section 5) per score beat per recording. In both table formats, the rows represent the number of score beats and the columns represent the index of the recordings of the particular Mazurka. The recordings have been labeled using the same pianist-ID as in the Mazurka dataset. For each Mazurka another table has been created that includes the name of an expressive marking annotation found in the score and the number of score beat position where it is located. The score markings extracted are listed in Table 2.

We have included recordings in which the performer followed the repetitions designated in the score, and excluded ones that do not. We also excluded noisy recordings. By noisy recordings, we mean recordings with distortion artifacts (some old recordings) or live recordings with audience sounds that could not be removed. Following this cleanup process, the remaining Mazurkas and recordings were not included if the total number of recordings did not exceed twenty.

Dynamics
Markings: p, pp, mf, f, ff, sf, fz, accent (>), crescendo, decrescendo Text: sotto voce, dolce, dolcissimo, con anima, con forza, calando, espressivo, risoluto, leggiero, perdendosi, maestoso, gajo, smorzando
Tempo
Marking: fermata Text: ritenuto, a tempo, Tempo I., lento, vivo, Allegro ma non troppo, Allegro, legato, legato assai, legatissimo, moderato, animato, rubato, scherzando, stretto, agitato, rallentando, tenuto

Table 2: Score markings having to do with dynamics and tempo or timing.

The recordings date from 1902 to the early 2000s. There is no information available on the score edition used by each performer. Tracing the actual score used in the preparation of each performance is an impossible task. Multiple editions of Chopin's Mazurkas exist; as noted in [25, p.56], "since most of [Chopin's] works were published in simultaneous 'first' editions in France, Germany and England, and since he also made alterations in the scores of various pupils, there are inevitably many discrepancies." Even the (arguably) most widely used editions of Peters, Schirmer, and Augener bear the marks of later edits.

For the purposes of obtaining score-based tempo and dynamic markings, we used the Paderewski, Bronarski and Turczynski edition as it is one of the most popular and readily available editions. A comparison of dynamic markings across different score editions reveals a few differences. The most common reason for a difference between editions arises from a slight displacement in marking position of usually only one or two beats. Less commonly, if a location typically does not have any dynamic marking, an outlying edition may have one there, presented directly or inside parentheses. On a rare occasion, a marking that appears in most editions may be replaced by a completely different one in a maverick edition.

We encode each Chopin Mazurka score in XML format using Musescore [4] and we extract the location of each tempo and dynamic marking using the Music21 software package [26], the result of which was verified manually. A long '>' appears in the score edition mentioned above, which serves as an indication of an "agogic" accent: "an emphasis created by a slight *lengthening* rather than dynamic emphasis on a note or chord" [25, p.53]. However this marking could not be included in our XML edition as it is not supported by the Music21 software.

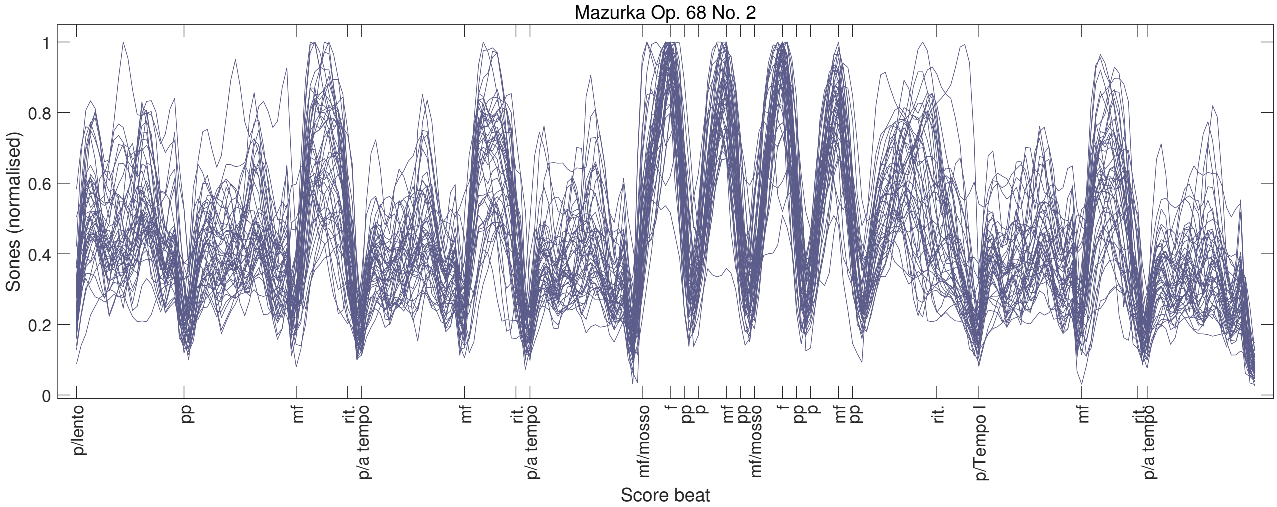
Figure 1 graphs the score-aligned loudness and inter-beat-interval (IBI) values for all 48 recordings of Mazurka Op. 68

[3] The dataset is publicly available and it can be found at: `https://github.com/katkost/MazurkaBL`. For copyright reasons, it does not include the audio files.

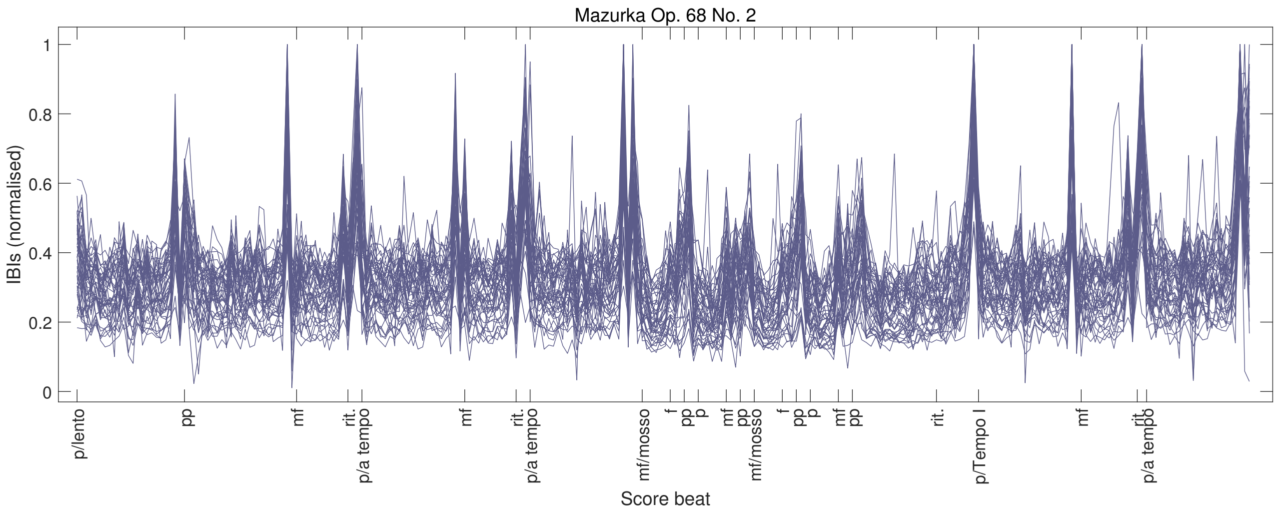
[4] `http://www.musescore.org`

Mazurka index	M06-1	M06-2	M06-3	M07-1	M07-2	M07-3	M17-1	M17-2	M17-3	M17-4	M24-1
# recordings	34	42	42	41	35	58	45	50	36	67	46
Mazurka index	M24-2	M24-3	M24-4	M30-1	M30-2	M30-3	M30-4	M33-1	M33-2	M33-3	M33-4
# recordings	56	39	54	45	50	54	55	48	50	23	63
Mazurka index	M41-1	M41-2	M41-3	M41-4	M50-1	M50-2	M50-3	M56-1	M56-2	M56-3	M59-1
# recordings	35	42	39	33	45	40	67	34	48	51	41
Mazurka index	M59-2	M59-3	M63-1	M63-3	M67-1	M67-2	M67-3	M67-4	M68-1	M68-2	M68-3
# recordings	56	56	42	62	35	31	40	42	38	48	42

Table 3: Chopin Mazurkas used in this study and the number of recordings for each one. Mazurkas are indexed as "M<*opus*>-<*number*>."



(a)



(b)

Figure 1: Raw time-series representation of the MazurkaBL dataset for Mazurka Op. 68 No. 2. (a) shows a plot of the dynamic values in sones and (b) the Inter-Beat-Interval (IBI) per score beat for all 48 recordings, each presented as a separate curve. Expressive markings show on the x-axis at their corresponding locations in the score.

No. 2 from the MazurkaBL dataset. Each recording's loudness and IBI values were re-scaled to the range $[0, 1]$. Each recording is represented as an individual time-series curve of either the sone values for dynamics (a) or the IBI values for timing (b). By inspection, regions of agreement and parts where greater variation occurs are immediately apparent, as are the regions where certain outliers can be found. Similar interactive plots for all Mazurkas are available online [5] where it is possible to include or exclude particular curves separately, provide details of the exact values as well as the name of the pianist per curve, and zoom in to regions of interest.

In the next section we explain how we dealt with the problem of linking the beat positions in the score to their corresponding positions in each recording.

4. SCORE BEAT INFORMATION

The position of score markings can be specified using the musical time axis of beats and measures. To study how a specific pianist realises a given marking in a performance, we need to locate its corresponding position in the recording in seconds. A common way to do this is to manually annotate the position of each musical beat in each available recording by tapping while listening to the music [1] and using specialised tools such as Sonic Visualiser [6] to check and correct the results. While manual annotations are typically quite reliable and accurate, creating them is highly time consuming and labour intensive. For example, for this research, the manual annotation and correction by inspecting the spectrogram of a single recording of Mazurka Op. 6 No. 2, which is approximately three minutes long, took 35 minutes on average.

To automate much of this annotation process, one can employ computational music alignment methods. Given a beat position in one rendition of a music piece, such synchronisation methods automatically locate the corresponding position in another version. In this way, for each piece, we only need to annotate a single recording, as we can use the automatically computed alignments to find, for each beat position in the annotated recording, the corresponding position in another recording. We call this annotated recording the *reference audio*. Its beat positions are transferred automatically to all the remaining recordings using a multiple recording alignment heuristic described in the next sections.

The approach to use a *reference* recording in an alignment procedure is not new—see, for example, [7] and [8]—and it has been shown to provide a significant stabilising effect on alignment accuracy. In this study, the multiple alignment heuristic calls the pairwise alignment algorithm by Ewert et al. [13], which applies Dynamic Time Warping (DTW) to chroma features. This pairwise alignment technique extends previous synchronization methods by incorporating features that indicate onset positions for each chroma. The authors report a significant increase in alignment accuracy resulting from the use of these chroma-onset features and an average onset error of 44 ms for piano recordings.

While alignment errors and corresponding inaccuracies in the derived annotations cannot be completely avoided, the synchronization enables the re-use of manually created annotations for a relatively small number of recordings to efficiently mass-annotate large databases. The choice of reference audio directly impacts the accuracy of the alignment. Intuitively, if an audio is an outlier, highly different from all the others in the set, it is a poor choice as a *reference* audio for accurate alignment to all other recordings. In order to determine the best choice of a *reference audio*, we created a ground truth dataset, which consisted of all forty-two recordings of Mazurka Op. 6 No. 2, each manually annotated with score beat positions. We computed the optimal *reference audio*, then determined its properties and designed a heuristic to automatically select this *reference audio* for other Mazurkas.

The goal of the multiple recording alignment heuristic is to optimise the choice of a *reference audio* with which we can obtain better alignment accuracies than with another audio file. In order to understand the characteristics of such an audio, in Section 4.1 we present an analysis of the *reference audio* properties, and in Section 4.2 we present a heuristic to detect the optimal *reference audio*.

4.1 Optimal *reference audio* choice

For this section, we use as ground truth our manual annotations of score-beat positions in all forty-two recordings of Mazurka Op. 6 No. 2. As a rule, in our manual annotations, we have chosen to follow the melody line so as to capture the lyricism of the rubato in the piano playing. Here, our goal is to determine the audio file (*reference audio*) that, when aligned and its beat annotations transferred to other audio recordings in the set, predicts most accurately the score-beat positions of the other recordings.

For this experiment, we removed silences in the beginnings and ends of all recordings by discarding any audio at the beginning and end in which the loudness value was $<$ 0.002 sones (more information about the extraction of the sone values is given in Section 5). There are a total of 288 beats; no notes were struck on 11 of these beats. The alignment procedure calls the algorithm described in [13] for audio-to-audio alignment and the annotations (beat positions) from each candidate reference audio recording were transferred to all other recordings in a pairwise fashion.

Let n be the number of recordings. We thus obtain a total of $n \times (n-1)$ new sets of annotations generated from all the candidate reference audio files. To determine the audio that performed best in providing the alignments with the lowest beat prediction error, we compared the predicted beat positions to the annotated beat positions, the ground truth. The Jarque-Bera test showed that not all sets of prediction errors followed the normal distribution, hence every alignment result is described by the median error for each alignment pair. For each recording, we thus arrive at $n-1$ median error values. For the sets of median values, we implemented the non-parametric Friedman test, where the small p-value (p $= 3.1546 \times 10^{-31}$) indicates that at least one column's sample median is significantly different from the others. The multiple comparison test shows the audio with the lowest median error value, which we interpret to be the best *reference audio*, to be Sztompka (1959), highlighted in bold in Figure 2, followed closely by the median error value of Kiepura's (1999) recording. Note that the y-axis is oriented so that the lowest values are at the top.

[5] `https://goo.gl/xC5LcY`
[6] `http://sonicvisualiser.org/`

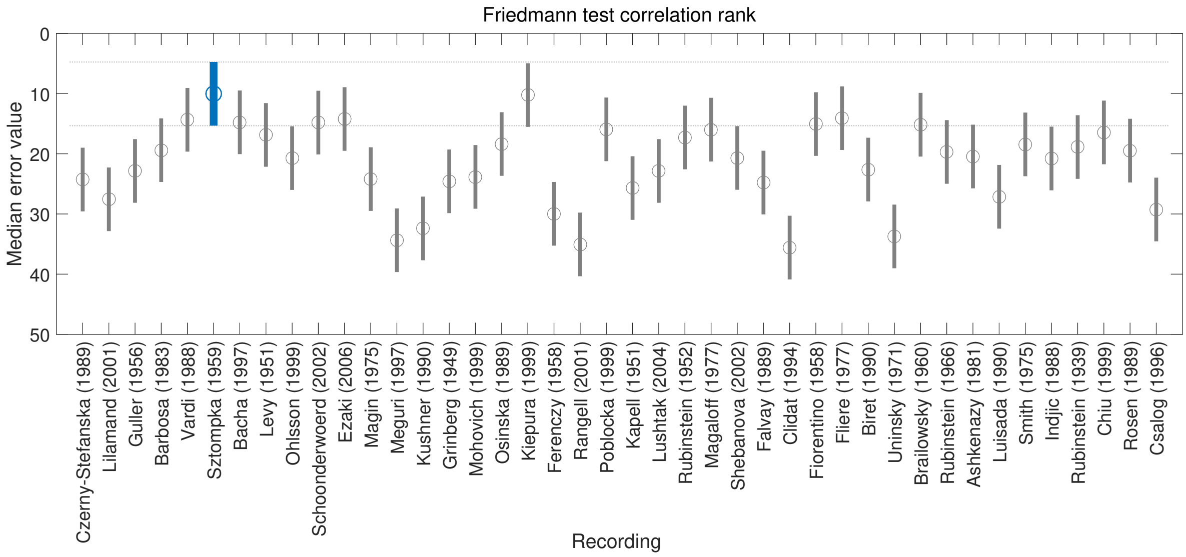


Figure 2: Error bars for the median beat prediction error when each of the 42 recordings of Mazurka Op. 6 No. 2 served, in turn, as the *reference audio*. x-axis shows pianist and recording year. The Friedmann correlation rank test showed Sztompka (1959) to be the recording with the lowest correlation rank and it is identified as the optimal *reference audio,* followed closely by Klepura (1999). Dotted horizontal lines mark the error bar limits of Sztompka (1959).

4.2 *Reference audio* selection heuristic

We first set up a fitness measure for a *reference audio* choice. The pairwise alignment algorithm [13] produces a match between two audio files, say i and j, using dynamic time warping. The alignment result is presented in the form of two column vectors $\mathbf{p}_i$ and $\mathbf{q}_j$, each with m entries, where m depends on the two recordings chosen, i and j. Each vector presents a nonlinear warping of the chroma features for the corresponding audio file, and represents the timing difference between the two recordings. A pair of entries from the two vectors gives the indices of the matching time frames from the two audio files. We compute the Euclidean distance between each pair of the dynamic time warped audio files as follows:

$$d_{i,j} = \sqrt{\sum_{k=1}^{m} (q_{j,k} - p_{i,k})^2}, \qquad \forall i \neq j, \tag{1}$$

where $m \in N$ is the size of the vectors. In this way, each audio has a profile corresponding to its alignment to all other audio recordings, $\mathbf{d}_i = [d_{i,j}]$. The average value of all the alignment accuracies for the i^{th} recording in relation to the remaining ones is $\overline{\mathbf{d}}_i$.

We consider the best reference file to be one with the minimum average distance to other audio files, which, at the same time, does not exhibit extreme differences to more than two other audio recordings as measured by the norm distance. In this way, after exploring alternative values of outliers, a test on Mazurka Op. 6 No. 2 identified the same *reference audio* as that found using the exact method of Section 4.1. Mathematically, the problem of finding the *reference audio* can be expressed as one of solving the following problem:

$$\begin{aligned} &\min_i \overline{\mathbf{d}}_i \\ \text{s.t.} \quad &\#\{j : |d_{i,j}| > q_3(\mathbf{d}_i) + 1.5[q_3(\mathbf{d}_i) - q_1(\mathbf{d}_i)]\} \leq 2, \end{aligned}$$

where $q_\ell(\mathbf{d}_i)$ is the ℓ-th quantile of $\mathbf{d}_i$, and the left hand side of the inequality uses an interquartile-based representation of an outlier. The *reference audio* is then given by $\arg\min_i \overline{\mathbf{d}}_i$.

We evaluate the method using the ground truth created using Mazurka Op. 6 No. 2. For each candidate *reference audio*, we compared the *reference audio*-derived beat positions with the manually annotated beat positions for the remaining forty-one recordings of the Mazurka. The average error was found to be 30.7 ms.

4.3 Evaluation of score beat positions

Several approaches for evaluating alignment procedures exist—see, for example, [27] and references therein. For alignment procedures that do not follow a *reference* recording, such as in [28], the number of beats that are created may not be the same as the number of beats in the ground truth; thus, evaluation metrics different from that in this study may be employed.

For this study, in order to evaluate the beat positions of the MazurkaBL dataset, we compare them with the manual annotations provided by the Mazurka project. The Mazurka project provides publicly available manual annotations for 63 recordings of Mazurka Op. 17 No. 4, 64 recordings of Mazurka Op. 24 No. 2, 34 recordings of Mazurka Op. 30 No. 2, 95 recordings of Mazurka Op. 63 No. 3, and 50 recordings of Mazurka Op. 68 No. 3. The intersection of these with the recordings in MazurkaBL provides pairs of aligned positions for 48, 54, 30, 62, and 42 recordings of the respective Mazurkas mentioned for comparison. The results of the comparison in terms of mean and standard deviation of the beat difference (in milliseconds) are presented in Table 4.

The average beat difference between our manual beat annotations in the *reference audio* and the manual beat annotations of the corresponding recording from the Mazurka

Piece (# beats)	Diff mean (ms)	Diff std (ms)
M17-4 (395)	85	150
M24-2 (360)	69	119
M30-2 (193)	66	41
M63-3 (229)	71	61
M68-3 (180)	80	69

Table 4: Summary statistics for the difference between MazurkaBL (alignment-based beat transfer from manually-annotated reference audio) and Mazurka project (all manual) beat annotations.

Piece (# beats)	Diff mean (ms)	Diff std (ms)
M17-4 (395)	65.7	86.6
M24-2 (360)	64.2	21.9
M63-3 (229)	63.8	21.7
M68-3 (180)	57.5	33.7

Table 5: Summary statistics for the difference between the manual beat annotations of the MazurkaBL *reference audio* and the manual annotations of the corresponding recording from the Mazurka project.

Project is given in Table 5. Beat annotations of the Mazurka recording of Op. 30 No. 2 corresponding to the *reference audio* for that Mazurka in MazurkaBL was not available.

Table 4 shows that the beat annotations of the *reference audio* and of the annotations transferred from the *reference audio* for Mazurka Op. 17 No. 4 differ most from the corresponding manual annotations of the Mazurka project. The information provided in Table 5 shows how much manual annotations may differ from one annotator to the next; this may reflect a difference in the chosen criteria for marking beats.

5. LOUDNESS INFORMATION

In the MazurkaBL dataset, the loudness time series is extracted from each recording using the *ma_sone* function in Pampalk's Music Analysis toolbox [7]. The loudness time series is expressed in sones. There are two reasons we choose the sone values as a measure of dynamics. The sone scale is psycho-acoustically linear, so we can more readily and accurately normalise the values across different recorded environments. Furthermore, without having to apply any audio compression or modification, the sone calculations automatically pre-processes the audio intensity values based on the psychoacoustic concept of equal loudness curves.

The specific loudness sensation in sones per critical band is calculated by following the process explained in [29]. Using this procedure, we calculate the power spectrum of the audio signal using a Fast Fourier Transform. We then use a window size of 256 samples, a hopsize of 128, and a Hanning window with 50% overlap. The frequencies are bundled into 20 critical bands and these frequency bands "reflect characteristics of the human auditory system, in particular of the cochlea in the inner ear." [29] We also calculate the spectral masking effects, based on the research presented in [30]. Then we calculate the loudness in dB-SPL units, and from these values we calculate the equal loudness levels in phons via stored curves of equal loudness level. Next, from the phon values, we detect the values in sones, following the calculation described in [31], according to which the loudness level S in sones can be calculated from the loudness levels L in phons using the formula:

$$S = \begin{cases} 2^{(L-40)/10}, & L \geq 40 \\ (L/40)^{2.642}, & L < 40, \end{cases} \tag{2}$$

the rationale being that "in this way the threshold of hearing and the nonlinear and frequency-dependent response of the ear to intensity differences are taken into account." [31]

The sone values are smoothed by local regression using a weighted linear least squares and a 2nd degree polynomial model (the "loess" method of MATLAB's *smooth* function [8]). The loudness time series for each recording is normalised to $[0, 1]$ by dividing the values of a recording by the maximum loudness value of that particular recording.

6. RECENT APPLICATIONS OF MazurkaBL

This section presents some studies that have used the MazurkaBL dataset and briefly describes their findings.

The set of markings $\{\boldsymbol{pp}, \boldsymbol{p}, \boldsymbol{mf}, \boldsymbol{f}, \boldsymbol{ff}\}$ were studied in [32], which explored the absolute meanings of the dynamic markings change as a function of the intended (score defined) and projected (recorded) dynamic levels, and that of the surrounding musical context. The analysis revealed a (sometimes) wide range of realisations of the same dynamic markings throughout a recording of a piece. Reasons for this counter-intuitive phenomenon include the score location of the markings, such as the beginning of a piece, and the marking's location in relation to that of previous ones. The analysis showed that, transitions from a louder to a softer marking, between markings of high intensity, and between markings of high contrast, tend to be more consistent. For markings that appear in the score more than once, most often than not, there was significant variation in the ways the markings were interpreted.

Offering a different perspective, [33] addressed the question of whether changes in dynamics, as automatically identified by statistical change-point algorithms, corresponded to dynamic markings. The assumption was that a dynamic marking indicated a point of change, and thus served as ground truth on which to evaluate the change-point algorithms. The results show that significant dynamic score markings do indeed correspond to change points, and evidence suggests that change points in score positions without dynamic markings serve to bring prominence to structurally salient events or to events the introduce a change in tempo.

A subset of the MazurkaBL dataset was used in [34] to investigate the bi-directional mapping between dynamic

[7] www.pampalk.at/ma/documentation.html

[8] http://uk.mathworks.com/help/curvefit/smooth.html?refresh=true

markings in the score and performed loudness. The study applied machine-learning techniques to the prediction of loudness levels corresponding to dynamic markings, and to the classification of dynamic markings given loudness values. The results show that loudness values and markings can be predicted relatively well when trained on different recordings of the same piece, but fail dismally when trained on the pianist's recordings of other pieces, demonstrating that score features may trump individual style when modeling loudness choices. The evidence suggested that all the features chosen for the prediction and classification tasks—current/previous/next dynamic markings, distance between markings, and proximity of dynamic-related and non-dynamic markings —were relevant. Furthermore, analysis of the results reveal the forms (such as the return of the theme) and structures (such as dynamic marking repetitions) influence the predictability of loudness levels and dynamic markings.

Finally, [35] describes another study that applied machine learning techniques to a subset of the MazurkaBL dataset. The goal of this study was to examine tempo-loudness interactions at specific score markings over a set of recordings, and to investigate how including information about one parameter impacted prediction of the other. The authors considered score markings indicating loudness or tempo change, and the model included score, tempo, and loudness-related features. When considering recordings of the same Mazurka, experiments showed that considering loudness-related features did not improve prediction of tempo change. However, adding tempo-related features did result in marginal improvement in predicting loudness change. As before, the predictions failed when the model was trained on loudness or tempo change information from recordings of multiple Mazurkas performed by the same pianist.

7. FUTURE DIRECTIONS

We have presented MazurkaBL, a new dataset for expressive music performance studies, comprising of 2000 beat-aligned recordings of forty-four Chopin Mazurkas overlaid with loudness information and score markings pertaining to tempo/timing and dynamics.

We provide material to quantitatively investigate what the score notation represents from a performer's perspective. Future tools providing different ways of visualising the dataset can bring insights that lead to a new notation system that represents changes in expression. Once important changes have been identified, symbols can be chosen to signify these changes and the representation can serve as a tool for comparing and analysing performances.

Much research has focused on proposing and establishing the relationship between dynamics and timing variations (see, for example [36] and references therein.) These studies range from establishing simple rules such as louder passages tend to be faster [37] to audio-synchronised animations of expressive parameters in tempo-loudness space [38]. Musical timing and amplitude has also been linked to subjective ratings of emotionality, for example in [39]. Timing and loudness variations in a music performance form critical cues for the identification of core music features such as phrase boundaries—see, for example, [40], [41], and [42]). The MazurkaBL dataset opens up many more avenues for explorations of this kind, and on a much larger scale.

Some future directions include expanding the list of score markings such as pedaling, and including audio features such as timbre. Further analytical studies could investigate gradual changes such as the analysis of *crescendo* or *diminuendo*. Also the same approach of large-scale annotation of score-beat information can be applied to other audio recordings of music by other composers, for other instruments, and of other genres.

Acknowledgments

This research was funded in part by a Queen Mary University of London Principal's PhD studentship. The authors are also grateful to Sebastian Ewert for providing the code for the audio-to-audio alignment algorithm.

8. REFERENCES

[1] C. Sapp, "Comparative analysis of multiple musical performances," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 497–500.

[2] S. Flossmann, W. Goebl, M. Grachten, B. Niedermayer, and G. Widmer, "The Magaloff project: An interim report," *Journal of New Music Research*, vol. 39, no. 4, pp. 363–377, 2010.

[3] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland Music Data (SMD)," in *Late-Breaking and Demo Session of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, 2011.

[4] M. Marchini, R. Ramirez, P. Papiotis, and E. Maestre, "The Sense of Ensemble: A Machine Learning Approach to Expressive Performance Modelling in String Quartets," *Journal of New Music Research*, vol. 43, no. 3, pp. 303–317, 2014.

[5] P. Papiotis, "A computational approach to studying interdependence in string quartet performance," Ph.D. thesis, Univesitat Pompeu Fabra, 2016.

[6] C. Sapp, "Hybrid numeric/rank similarity metrics for musical performance analysis." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Philadelphia, USA, 2008, pp. 501–506.

[7] A. Arzt and G. Widmer, "Real-time music tracking using multiple performances as a reference," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015, pp. 357–363.

[8] S. Wang, S. Ewert, and S. Dixon, "Robust and efficient joint alignment of multiple musical performances," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2132–2145, 2016.

[9] M. Müller, T. Prätzlich, and J. Driedger, "A cross-version approach for stabilizing tempo-based novelty detection." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012, pp. 427–432.

[10] A. Benetti Jr., "Expressivity and musical performance: Practice strategies for pianists," in *Performance Studies Network International Conference*, Cambridge, UK, 2013. [Online]. Available: http://www.cmpcp.ac.uk/wp-content/uploads/2015/11/PSN2013_Benetti.pdf

[11] J. Rink, *Musical Performance: A Guide to Understanding*. Cambridge University Press, 2002.

[12] D. Fabian, R. Timmers, and E. Schubert, *Expressiveness in music performance: Empirical approaches across styles and cultures*. Oxford University Press, 2014.

[13] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[14] T. Rutherford-Johnson, M. Kennedy, and J. Kennedy, *The Oxford Dictionary of Music*. OUP Oxford, 2012.

[15] S. Barrett, "Reflections on music writing: Coming to terms with gain and loss in early medieval latin song," in *Vom Preis des Fortschritts: Gewinn Und Verlust in der Musikgeschichte*, A. Haug and A. Dorschel, Eds. Universal Edition, 2008.

[16] N. Cook, E. Clarke, L.-W. Daniel, and J. Rink, *The Cambridge Companion to Recorded Music*. Cambridge University Press, 2009.

[17] J. Sadie, *Companion to Baroque Music*. University of California Press, 1998.

[18] A. Thomas, "Beyond the dance," in *The Cambridge Companion to Chopin*, J. Samson, Ed. Cambridge University Press, 1994.

[19] J.-J. Eigeldinger, "Placing chopin: Reflections on a compositional aesthetic," in *Chopin Studies 2*, J. Rink and J. Samson, Eds. Cambridge University Press, 1994, vol. 2.

[20] L. H. Shaffer, "Performing the F# minor prelude op. 28 no. 8," in *Chopin Studies 2*, J. Rink and J. Samson, Eds. Cambridge University Press, 1994, vol. 2.

[21] J. Methuen-Campbell, "Chopin in performance," in *The Cambridge Companion to Chopin*, J. Samson, Ed. Cambridge University Press, 1994.

[22] A. Einstein, *Music in the Romantic Era: A History of Musical Thought in the 19th Century*. New York : W.W. Norton, 1975.

[23] J. Rink, "Authentic chopin: History, analysis and intuition in performance," in *Chopin Studies 2*, J. Rink and J. Samson, Eds. Cambridge University Press, 1994, vol. 2.

[24] H. C. Khoo, "Playing with dynamics in the music of chopin," Ph.D. dissertation, Royal Holloway, University of London, 2007.

[25] E. Bailie, *Chopin: A Graded Practical Guide*, ser. Pianist's Repertoire. Kahn & Averill, 1998.

[26] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, Netherlands, 2010, pp. 637–642.

[27] T. Prätzlich and M. Müller, "Triple-based analysis of music alignments without the need of ground-truth annotations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 266–270.

[28] P. Grosche, M. Müller, and C. S. Sapp, "What makes beat tracking difficult? a case study on chopin mazurkas." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, Netherlands, 2010, pp. 649–654.

[29] E. Pampalk, A. Rauber, and D. Merkl, "Content-based organization and visualization of music archives," in *Proceedings of the ACM Multimedia*, Juan les Pins, France, 2002, pp. 570–579.

[30] M. R. Schroeder, B. S. Atal, and J. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *The Journal of the Acoustical Society of America*, vol. 66, no. 6, pp. 1647–1652, 1979.

[31] R. A. W. Bladon and B. Lindblom, "Modeling the judgment of vowel quality differences," *The Journal of the Acoustical Society of America*, vol. 69, no. 5, pp. 1414–1422, 1981.

[32] K. Kosta, O. F. Bandtlow, and E. Chew, "Dynamics and relativity: Practical implications of dynamic markings in the score," *Journal of New Music Research*, 2018 (to appear).

[33] ——, "A change-point approach towards representing musical dynamics," in *Mathematics and Computation in Music*. London, UK: Springer Lecture Notes in Artificial Intelligence 9110, 2015, pp. 179–184.

[34] K. Kosta, R. Ramirez, O. F. Bandtlow, and E. Chew, "Mapping between dynamic markings and performed loudness: A machine learning approach," *Journal of Mathematics and Music*, vol. 10, no. 2, pp. 149–172, 2016.

[35] C. Vaquero, I. Titov, and H. Honing, "What score markings can say of the synergy between expressive timing and loudness," in *Abstracts from European Society for Cognitive Sciences Of Music Conference*, Ghent, Belgium, 2017.

[36] G. Widmer and W. Goebl, "Computational Models of Expressive Music Performance: The State of the Art," *Journal of New Music Research*, vol. 33, no. 3, pp. 203–216, 2004.

[37] B. H. Repp, "The dynamics of expressive piano performance: Schumann's Träumerei revisited," *The Journal of the Acoustical Society of America*, vol. 100, no. 1, pp. 641–650, 1996.

[38] S. Dixon, W. Goebl, and G. Widmer, "The Performance Worm: Real Time Visualisation of Expression based on Langner's Tempo-Loudness Animation," in *Proceedings of the International computer music conference (ICMC)*, Gothenburg, Sweden, 2002.

[39] A. Bhatara, A. K. Tirovolas, L. M. Duan, B. Levy, and D. J. Levitin, "Perception of emotional expression in musical performance," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 37, no. 3, 2011.

[40] C.-H. Chuan and E. Chew, "A Dynamic Programming Approach to the Extraction of Phrase Boundaries from Tempo Variations in Expressive Performances," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 305–308.

[41] E. Cheng and E. Chew, "Quantitative Analysis of Phrasing Strategies in Expressive Performance: Computational Methods and Analysis of Performances of Unaccompanied Bach for Solo Violin," *Journal of New Music Research*, vol. 37, no. 4, pp. 325–338, 2008.

[42] D. Stowell and E. Chew, "Maximum a Posteriori Estimation of Piecewise Arcs in Tempo Time-Series," in *From Sounds to Music and Emotions – International Symposium on Computer Music Modeling and Retrieval (CMMR)*. London, UK: Springer LNCS 7900, 2012, pp. 387–399.

TRADITIONAL MODAL MONODIES GENERATIVE GRAMMAR ENCODING IN THE MUSIC ENCODING INITIATIVE

Mark Asmar
Ticket Lab, Faculty of Engineering
Antonine University
mark.asmar.ma@gmail.com

Talar Atechian
Ticket Lab., Faculty of Engineering
Antonine University
talar.atechian@ua.edu.lb

Nidaa Abou Mrad
Research Centre for Musical Traditions
Faculty of Music and Musicology
Antonine University
nidaa.aboumrad@ua.edu.lb

Sylvaine Leblond Martin
Chaire Unesco ITEN
Maison des Sciences de l'Homme Paris Nord
Université Paris 8
sylvaine.leblond-martin@mshparisnord.fr

ABSTRACT

Encoding written music with a textual format is a technology developed for exchanging and analyzing digital music scores. In the literature, many standards exist for this purpose, such as, MusicXML and the Music Encoding Initiative (MEI). The Music Encoding Initiative is a standard developed for encoding music scores in XML. It supports the encoding of different types of notations, such as, Common Music Notation, Neumes Notation, etc. It encodes many features and elements related to musical components such as the pitch name, the octave and the duration of notes. However, for the researchers in musicology, additional information are necessary to enrich the MEI and in order to provide more specific music analysis. In this paper, we target the modal monodies analysis. Thus, we propose to enrich the MEI by appending to its initial schema additional information extracted from the generative grammar of modal monodies. The proposed solution consists of adding a custom module to the MEI containing new elements and attributes. In addition, a new semi-automated analysis component is proposed for the analysis of traditional Modal Monodies of the Middle East and the Mediterranean cultures.

1. INTRODUCTION

An important part of analytical musicology (and analytical ethnomusicology) is dedicated to study the melodic and rhythmic structure of compositions and improvisations from different traditions in the world, and to propose models to explain how it works. Thus, generative grammars were developed and proposed to enrich the analysis of music utterances. A musical generative grammar proposed in the Modal Semiotics theory [6], serves for analyzing traditional modal monodies of the Middle East and the Mediterranean cultures (including medieval European monodic music, as well as Mashriq and Maghreb traditions).

The Music Encoding Initiative encodes music in XML, a textual, yet structured format. An MEI document is an XML document associated to a schema that defines its structure [2]. An XML document would have a schema written in one of the following three languages: the W3C Schema (XSD), RelaxNG (RNG) or Document Type Definition (DTD) [2]. The MEI schema defines and describes the numerous elements existing for music encoding. A group of elements used for the same purpose constitutes a module. Unlike MusicXML, the MEI encodes notations other than the Common Music Notation (CMN) as well as metadata [1]. The Music Encoding Initiative suits our project of encoding the generative grammar of modal monodies according to the Modal Semiotics theory expressed in [6].

The project proposed in [11] and discussed in this paper aims at encoding the generative grammar proposed in [6] of the traditional modal monodies in MEI format. The proposed solution consists of adding a custom module to the MEI. Thus, an algorithm is developed and implemented to extract the grammar out of MEI-encoded music before encoding them back again with its grammar.

The paper structure is as follows: In the next section, a state of the art is presented introducing the added modules to the MEI as well as presenting existing projects concerning music analysis. The third section discusses the proposed solution. The fourth section discusses the experiments conducted to evaluate the proposed solution. Finally, we conclude the paper in the last section.

2. STATE OF THE ART

Several theories-based solutions are presented in the literature for analyzing music scores and partitions. In this section, we discuss solutions developed for analyzing music. In addition, we present contextualized modules added to MEI to enrich the MEI standard schema. The author in [7] developed a user interface for Schenkerian Analysis, aiming to analyze musical scores based on the Schenkerian theory proposed by Heinrich Schenker. In an attempt to overcome the difficulties of its computer implementation, the authors in [8] developed a solution for the Lerdahl and Jackendoff's Generative Theory of Tonal Music. They developed, implemented and tested four different analyzers.

An extension of the Text Encoding Initiative (TEI), a standard developed for encoding texts, made it possible to support the encoding of music within texts. The TEI used to encode texts and music occurring within texts, considering musical pieces or notes as images [4]. By adding the <notatedMusic> element to the TEI, the latter now supports the inclusion of music expressed in MEI, a graphical representation of the music or any other format representing the music [4]. All MEI elements within the <notatedMusic> element are prefixed with "*mei*:", for example <mei:music> [4].

The project described in [9] uses both MEI and TEI, creating a data model and using both the MEI and TEI for encoding of holdings of the Detmold Court Theatre (1825-1875), providing a catalog, which can also be used as a searching tool for specific data [9, 10].

The Solesmes module proposed in [2] captures Solesmes-specific music notation about Gregorian chant. According to [2], the MEI supports the encoding of the neume music notation; however, it lacks some specific features for the Solesmes-neume notation. A new module is proposed, adding new elements and attributes to the MEI schema in order to capture more accurately features related to the Solesmes-neume notation.

A new module proposed in [3], adds layout-related components in MEI, since the latter does not encode information concerning the layout. Using a separate sub-tree, the layout module allows the encoding of information concerning multiple visual representations of the music, while keeping the musical content intact.

However, in order to encode the generative grammar of modal monodies, which is the main goal of this paper, it is necessary to propose a contextualized custom module as an extension of the MEI schema. In addition, a semi-automated algorithm is proposed to implement the analysis process.

3. MODAL SEMIOTICS

The Modal Semiotics theory in [6], describes a generative grammar for modal monodies, related to musical traditions of the Mashriq. This grammar aims at rewriting these modal monodies based on a set of rules, describing mainly some particular modal monodies features such as the *rhythmic* parameter (morphological rewriting) and the *melodic* parameter (rhythmic melodic morphophonological rewriting and modal syntactic rewriting) of the music.

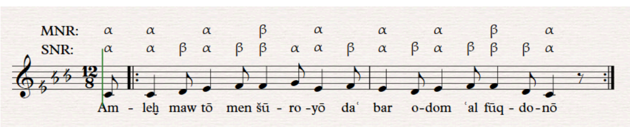



Figure 1. Nuclear Reduction of "Suġītō Qūm fawlōs", a Syriac Maronite Hymn.

According to the phonological component of this theory [6], the final note of a music utterance where the piece ends, helps deciding which notes belong to the alpha or primary modal nucleus and which belong to the beta or secondary modal nucleus (see Figure 1). Compared to the final note (considered as the first degree), even notes are alphas and odd notes are betas. Each "focal note" in the piece placed at the beginning of a syllable is assigned the appropriate symbol, either α or β. This is the "*Syllabic Nuclear Reduction*" [6].

The "*Metasyllabic Nuclear Reduction*" in rhythmic melodic morphophonological rewriting chooses out of the symbols in the previous phases, the ones that are more important. Having the following matrices for the previous musical score:

$$\begin{pmatrix} (♩, ♪) \\ (♩, ♪) \\ (♩, ♪) \\ (♩, ♪) \end{pmatrix} \leftrightarrow \begin{pmatrix} ♩. \\ ♩. \\ ♩. \\ ♩. \end{pmatrix}$$

The first matrix defines the rhythmical structure of the measure, meaning at each quarter note ♩, and each eighth note ♪ have one of the either two symbols, α or β, assigned to it. The second matrix shows that each quarter and eighth note are equal to a single quarter dotted note in the "*Metasyllabic Nuclear Reduction*", taking the symbol of the highest note's duration into consideration, in this case the quarter note.

The next step consists of rewriting the entire music score into rhythmic melodic morphophonological matrices and modal syntactic vectors. The process described is as follows:

$$N\big(E(\mu_1)\big) = N\big(\Lambda(\mu_1)R(\mu_1)\big) = N\big(\Lambda(\mu_1)\big)R(\mu_1) =$$

$$(\alpha, \alpha, \beta, \alpha) \begin{pmatrix} ♩. \\ ♩. \\ ♩. \\ ♩. \end{pmatrix} \leftrightarrow (\alpha, \alpha, \beta, \alpha) \begin{pmatrix} (|\text{leḫ}|♩, -|\text{maw}|♪) \\ (|\text{tō}|♩, -|\text{men}|♪) \\ (|\text{šū}|♩, -|\text{ro}|♪) \\ (|\text{yō}|♩, -|\text{da}'|♪) \end{pmatrix}$$

The matrix containing α and β is derived from the previous "*Metasyllabic Nuclear Reduction"* phase. However, adding the negative sign before notes depends on the multiplier of that note and its associated symbol [6]. For example, |maw|♪ symbol is β while having α as its multiplier, so a minus sign precedes it ($\alpha = -\beta$).

Next, the multiplication of the two matrices takes place to get the following result [6]:

$$\begin{pmatrix} \alpha.\,(|\text{leḫ}|♩, -|\text{maw}|♪), \alpha.\,(|\text{tō}|♩, -|\text{men}|♪), \\ \beta.\,(|\text{šū}|♩, -|\text{ro}|♪), \alpha.\,(|\text{yō}|♩, -|\text{da}'|♪) \end{pmatrix}$$

$$\leftrightarrow \begin{pmatrix} (\alpha.\,|\text{leḫ}|♩, \beta.\,|\text{maw}|♪), (\alpha.\,|\text{tō}|♩, \beta.\,|\text{men}|♪), \\ (\beta.\,|\text{šū}|♩, \alpha.\,|\text{ro}|♪), (\alpha.\,|\text{yō}|♩, \beta.\,|\text{da}'|♪) \end{pmatrix}$$

Later, notes' pitch names replace alphas and betas to obtain the "*Phonological Realization*":

$$\begin{pmatrix} (C.\,|\text{leḫ}|♩, D.\,|\text{maw}|♪), (E.\,|\text{tō}|♩, F.\,|\text{men}|♪), \\ (F.\,|\text{šū}|♩, G.\,|\text{ro}|♪), \big(E.\,|\text{yō}|♩, F.\,|\text{da}'|♪\big) \end{pmatrix}$$

The "*Vector Transcoding*" comes next, which consists of transforming the piece into vectors (primordial prolongative vector $\vec{p}$, suspensive vector $\vec{s}$, questioning vector $\vec{q}$, responsive vector $\vec{r}$) under the following rules, as described in [6]:

$$\overrightarrow{\alpha\alpha} = \vec{p} \qquad \overrightarrow{\beta\beta} = \vec{s} \qquad \overrightarrow{\alpha\beta} = \vec{q} \qquad \overrightarrow{\beta\alpha} = \vec{r}$$

The result for the first measure and the entire piece are as follows respectively:

$$(\vec{p},\vec{q},\vec{r}) \qquad (\vec{p},\vec{q},\vec{r},(\vec{p}),\vec{p},\vec{q},\vec{r})$$

The so called "*Syntactic Elaboration*" clotures the analysis, by rewriting entirely the music utterance using vectoral decomposition equations, starting with the Fundamental/Original Structure or Primordial Dichotomy equation, as described in [6]:

$$\{\vec{p}\} \rightarrow \{[\vec{q}] + [\vec{r}]\}$$

4. PROPOSED SOLUTION

In order to attend our goal, the solution consists of adding a new custom module, named "grammar" to the MEI schema. In addition to the semi-automated algorithm for music scores analysis.

4.1 The Schema Extension

Similar to TEI, MEI schema is extensible. It allows enriching the encoding process by contextualized custom modules. Thus, developers can generate their own custom schemas out of the initial MEI schema [2]. First, an XML document describes the expected custom output schema, then using the TEI stylesheets; the latter use the XML file and the MEI schema to generate a schema file describing the expected MEI schema. Referring to this technical approach, the new custom module so called "*grammar*" is generated, for encoding generative grammar of Modal Monodies within MEI. The new elements and attributes added to the schema are present below in Figure 2.

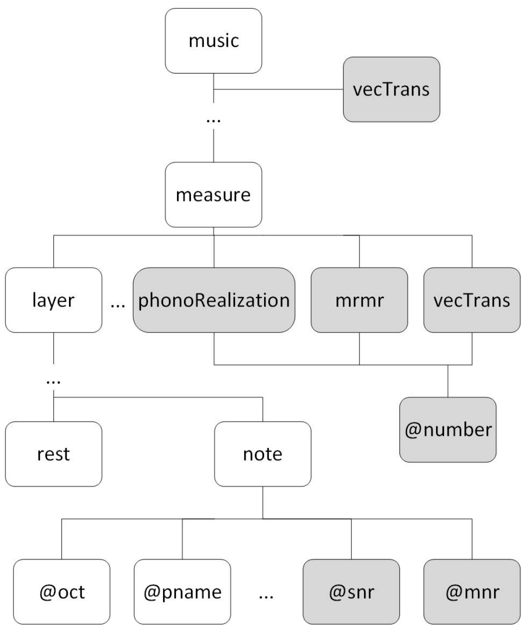


Figure 2. Newly added elements and attributes to the MEI schema.

Table 1 describes the elements and attributes of the "*grammar*" module.

Attributes	
Attribute	**Description**
snr	Attribute for the note element. Used for the "*Syllabic Nuclear Reduction*", it may contain only two values α and β.
mnr	Attribute for the note element. It may contain "yes" or "no" values, describing whether the note is taken into consideration in the "*Metasyllabic Nuclear Reduction*" or not.
number	Serves as an identifier for the <phonoRealization>, <mrmr> and <vecTrans> elements in the case of considering repetitions, reflecting the index of the measure in the analysis before and after the repetition.
Elements	
Element	**Description**
mrmr	Child of the <measure> element. Destined for the encoding of the "*Morphophonological Rhythmic and Melodic Rewriting*"; it contains matrices and mathematical equations.
phonoRealization	Child of the <measure> element. Used to encode the "*Phonological Realization*" phase of the analysis, just like *mrmr* it contains matrices and equations as well.
vecTrans	Child of both the <measure> and <music> elements. It serves as the element for the encoding of the "*Vector Transcoding*" phase, and it contains vectors.

Table 1. Description of the components of the "grammar" module.

All elements and attributes present in Table 1 contain mathematical expressions, symbols or expressions, except for the *mnr* and *number* attributes, so having a textual format for representing equations within XML is mandatory. The *snr* attribute has only two values, "\alpha" or "\beta", while the three elements contain equations including matrices and vectors, making the TeX language suitable, similar to the TEI encoding of mathematical expressions [5]. The TeX package, "lilyglyps" a package for displaying "Lilypond" music symbols, used for the representation of notes and rests in the matrices during the encoding process.[1]

[1] https://ctan.org/pkg/lilyglyphs

4.2 The Semi-Automated Music Analysis Algorithm

The semi-automated music analysis algorithm is presented in this section. It consists of implementing the music analysis proposed in [6], and to encode the result in the "*grammar*" module. However, the phases stating from the "*Syllabic Nuclear Reduction*" until the "*Vector Transcoding*" were done, leaving the last phase, "*Syntactic Elaboration*", unimplemented.

The algorithm takes as an input a score of music encoded in MEI, and returns an updated MEI document containing the "*grammar*" elements and the attributes as results of the analysis. The analysis process proceeds in a sequential manner, for each measure in the analyzed music score, all analysis phases are applied; when completed, the next measure is analyzed and so on.

The same process executes when the analysis includes repetitions except that prior to the analysis repeated measures is mandatory. In the case, of multiple verses in each processing of the measure the appropriate verse is taken into account.

4.2.1 Syllabic Nuclear Reduction

This phase consists of assigning for each note in the music score α and β symbols. First, the lowest and the highest notes of the music score are identified.
In addition, the algorithm identifies the final note of the music score and identified as alpha note (α). According to this reference note, the alpha (α) and beta (β) notes of the music score are identified.

Descending from the final note towards the lowest note, each even note after the final note is an alpha while odd notes are betas, for example, if the fourth octave C note (C4) is the final note, and the third octave A note (A3) is the lowest, C4 is α, B3 is β and A3 is α. The process stops when the algorithm reaches the lowest note in the piece. The second part constitutes of a similar process, starting from the final note and moving towards the highest note, even notes are alpha and odd notes are betas as well, for example, if C4 is the final note, and F4 is the highest, C4 is α, D4 is β, E4 is α and F4 is β. Next, the algorithm checks the music given as input for syllables, if they exist then the notes residing at the beginning of each syllable are assigned to their corresponding symbol from the aforementioned array. If no syllables exist, then a calculation of the nuclear reduction based on a user given matrix takes place. Having the following piece of music [6]:

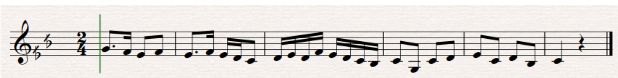



Figure 3. Dūlāb Rāst.

Moreover, the following matrix (based on the percussive *waḥda* cycle *dum*♩*, tak* ♪*, tak* ♪*)*:

$$\begin{pmatrix} ♩ \\ (♪,♪) \end{pmatrix}$$

The analysis algorithm tries to find out matches between the notes within the matrix, represented by the following values [[0.25], [0.125, 0.125]], and those within every measure in the music score. The above matrix describes each measure, meaning that every measure contains a quarter note and two eighth notes, or their equivalents; for example, two eighth notes instead of a quarter note. By processing each measure sequentially, each note's duration[2]

$$\frac{1}{n} \tag{1}$$

(a quarter note would have a duration of $\frac{1}{4}$, n retrieved from the note's *dur* attribute) is checked with the first element of the matrix. Considering that element is d, the *snr* attribute is added in case the note's duration is equal to the duration of the latter, while the result of (1) is added to a sum in the other case. The first note of the sum is the note that has the *snr* attribute added to it when the sum is equal to d, meaning that a series of notes equal to d were found, for example, finding two eighth notes while searching for a quarter note. When an element of the matrix is found, the algorithm proceeds to the next one. Dotted notes are calculated using their corresponding equivalents, for example, a ♩. is replaced by a quarter note and an eighth note ♩. = ♩ + ♪ . The equivalents' durations are useful for calculating the durations using the aforementioned formula. Figure 4 shows the result for Figure 3, as follows:

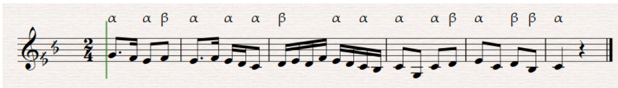


Figure 4. Dūlāb Rāst after SNR phase.

The analysis of a syllabled music score (see Figure 5), returns the result as illustrated in Figure 6.

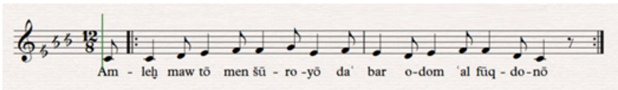


Figure 5. Suġītō Qūm fawlōs.

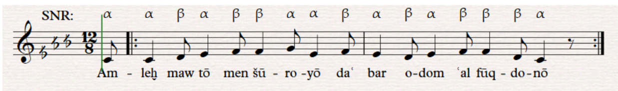


Figure 6. SNR phase result for Suġītō Qūm fawlōs.

The outcome of the "*Syllabic Nuclear Reduction*" phase introduces new elements in MEI encoding document as follows:

```
<note xml:id="m-40" dur="4" dur.ges="256p"
oct="4" pname="c" pnum="48" stem.dir="up"
snr="\alpha">
        <verse n="1">
                <syl wordpos="t">leḫ</syl>
        </verse>
</note>
```

[2] A quarter note would have a duration of four in MEI for example, however in this paper MEI durations are inversed 1.

In the element <note>, specific attributes and elements are defined as follows: the duration (*dur*) of the note, the pitch name (*pname*), the octave (*oct*), while having elements like <verse> and <syl> describing the verses and the syllables respectively.

In addition, new attribute (*snr*) is added to the schema, representing the assigned nucleus of a note in the "*Syllabic Nuclear Reduction*".

4.2.2 Metasyllabic Nuclear Reduction

The "*Metasyllabic Nuclear Reduction*" phase initiates after the completion of the previous phase. It needs as an input an initial matrix manually provided at the beginning of the analysis [6]. The following matrix is an example for the score in Figure 5.

$$SNR\ Matrix\colon \begin{pmatrix} (\text{♩}, \text{♪}) \\ (\text{♩}, \text{♪}) \\ (\text{♩}, \text{♪}) \\ (\text{♩}, \text{♪}) \end{pmatrix} \leftrightarrow MNR\ Matrix\colon \begin{pmatrix} \text{♩.} \\ \text{♩.} \\ \text{♩.} \\ \text{♩.} \end{pmatrix}$$

The analysis algorithm receives the matrices as arrays. A numerical value is assigned to each note as follows: [[0.25, 0.125], [0.25, 0.125], [0.25, 0.125], [0.25, 0.125]] and [0.375, 0.375, 0.375, 0.375] respectively. These values in the array stand for the duration of the notes in the matrix, and the arrays within the initial array represent the rows within the matrix.

For each row in the matrix, the corresponding note containing a *snr* attribute for the highest duration in that row is chosen in the "*Metasyllabic Nuclear Reduction*". The algorithm adds the *mnr* attribute to that note with a value of "yes". As illustrated in the music score of the Figure 6, the first row of its "SNR Matrix" contains a quarter and an eighth note. The quarter note being the highest, the first note in the first measure, will have the *mnr* attribute set to "yes", as for the second row in the matrix and the third note in the piece. The same process is applied for each row in the matrix.

This process is the same for every measure in the piece. The music score shown in Figure 7 contains the MNR and SNR.

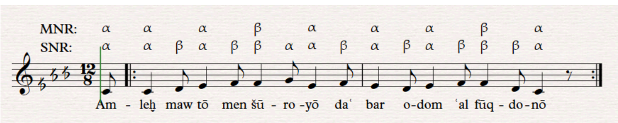



Figure 7. MNR phase result for Suġītō Qūm fawlōs.

As an example, the *mnr* attribute of the element <note> is represented as follows:

```
<note xml:id="m-40" dur="4" dur.ges="256p"
oct="4" pname="c" pnum="48" stem.dir="up"
snr="\alpha" mnr="yes">
        <verse n="1">
                <syl wordpos="t">leḫ</syl>
        </verse>
</note>
```

4.2.3 Morphophonological Rhythmic and Melodic Rewriting

After adding both the *snr* and *mnr* attributes, the process of generating matrices and the equations is necessary for the "*Morphophonological Rhythmic and Melodic Rewriting*".

We consider δ the matrix of *alphas* and *betas*. The notes of a measure containing the *snr* and *mnr* attributes having the latter set to "yes", will have their *snr* attribute values appended to δ.

$$\delta = (\alpha, \alpha, \beta, \alpha)$$

The above example represents the matrix generated for the first measure of the music score shown in Figure 6. This matrix is then, multiplied by the matrix given for the "*Metasyllabic Nuclear Reduction*", shown in Figure 6. The result is represented as follows:

$$(\alpha, \alpha, \beta, \alpha) \begin{pmatrix} \text{♩.} \\ \text{♩.} \\ \text{♩.} \\ \text{♩.} \end{pmatrix}$$

Next, the δ multiplies the "SNR Matrix" shown in the section 4.4.2. However, before the multiplication, if the matrix contains more than one element in any of its rows, then its corresponding symbol within the δ matrix is compared to its *snr* attribute value. If they are different then a negative sign precedes the note. In the case of a syllabed music score, the corresponding syllables precede the notes as well.

$$(\alpha, \alpha, \beta, \alpha) \begin{pmatrix} (|\text{leḫ}|\text{♩}, -|\text{maw}|\text{♪}) \\ (|\text{tō}|\text{♩}, -|\text{men}|\text{♪}) \\ (|\text{šū}|\text{♩}, -|\text{ro}|\text{♪}) \\ (|\text{yō}|\text{♩}, -|\text{da}'|\text{♪}) \end{pmatrix}$$

Considering $-|\text{maw}|$ ♪, the negative sign indicates that the note's equivalent within the music is β while its multiplier is α in the δ matrix.

The multiplication of the two matrices continues by multiplying each element in the δ matrix by its corresponding row in the second matrix. Resulting the following output:

$$\begin{pmatrix} \alpha.\,(|\text{leḫ}|\text{♩}, -|\text{maw}|\text{♪}), \alpha.\,(|\text{tō}|\text{♩}, -|\text{men}|\text{♪}), \\ \beta.\,(|\text{šū}|\text{♩}, -|\text{ro}|\text{♪}), \alpha.\,(|\text{yō}|\text{♩}, -|\text{da}'|\text{♪}) \end{pmatrix}$$

The final step of this entire phase[3] is the process of multiplying the symbols α and β with the elements present in each row, directly. If a minus precedes the element within the row, -its *snr* attribute is different from the multiplier- the multiplier changes, α turns into β and vice versa.

$$\begin{pmatrix} (\alpha.\,|\text{leḫ}|\text{♩}, \beta.\,|\text{maw}|\text{♪}), (\alpha.\,|\text{tō}|\text{♩}, \beta.\,|\text{men}|\text{♪}), \\ (\beta.\,|\text{šū}|\text{♩}, \alpha.\,|\text{ro}|\text{♪}), (\alpha.\,|\text{yō}|\text{♩}, \beta.\,|\text{da}'|\text{♪}) \end{pmatrix}$$

The matrices shown in this section are expressed in TeX format as added in the element named <mrmr>, child of the <measure> element, as a TeX string, as shown below:

[3] Notes of the matrix are replaced by rests if rests exist at the end of a measure

```
<mrmr>
\begin{pmatrix}
\alpha , \alpha , \beta , \alpha
\end{pmatrix}
\begin{pmatrix}
\quarterNoteDotted \\ \quarterNoteDotted \\
\quarterNoteDotted \\ \quarterNoteDotted
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha , \alpha , \beta , \alpha
\end{pmatrix}
\begin{pmatrix}
( |leẖ| \quarterNote,-|maw| \eighthNote ) \\
( |tō| \quarterNote,-|men| \eighthNote ) \\
( |šū| \quarterNote,-|ro| \eighthNote ) \\
( |yō| \quarterNote,-|da'| \eighthNote )
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha.( |leẖ| \quarterNote,-|maw| \eighthNote
),
\alpha.( |tō| \quarterNote,-|men| \eighthNote ),
\beta.( |šū| \quarterNote,-|ro| \eighthNote ),
\alpha.( |yō| \quarterNote,-|da'| \eighthNote )
\end{pmatrix}
\\
\\
\begin{pmatrix}
( \alpha.|leẖ| \quarterNote,\beta.|maw| \eighth-
Note ) ,
( \alpha.|tō| \quarterNote,\beta.|men| \eighth-
Note ) ,
( \beta.|šū| \quarterNote,\alpha.|ro| \eighth-
Note ) ,
( \alpha.|yō| \quarterNote,\beta.|da'| \eighth-
Note )
\end{pmatrix}
</mrmr>
```

If repetitions are considered in the analysis, the <mrmr> element is added twice with different values for the *number* attribute. For the first measure of the score shown in Figure 6, the result is as the following:

```
<mrmr number="1">
...
</mrmr>
<mrmr number="3">
...
</mrmr>
```

4.2.4 Phonological Realization

This phase consists of multiplying each element of the "SNR Matrix" with its corresponding note's pitch name. In the analysis algorithm, this step is part of the "*Morpho-phonological Rhythmic and Melodic Rewriting*". Repeating the final step performed in the previous phase, replacing *alphas* and *betas* by the corresponding pitch names of notes. A string in the TeX syntax expresses the entire equation, encoded as the value of the <phonoRealization> MEI element that is a child element of the <measure> element.

The "*Phonological Realization*" of the first measure of the music score shown in Figure 7 is represented as follows:

$$\begin{pmatrix} (C.|\text{leẖ}|♩, D.|\text{maw}|♪), (E.|\text{tō}|♩, F.|\text{men}|♪), \\ (F.|\text{šū}|♩, G.|\text{ro}|♪), (E.|\text{yō}|♩, F.|\text{da}'|♪) \end{pmatrix}$$

```
<phonoRealization>
\begin{pmatrix}
( C.|leẖ|\quarterNote , D.|maw|\eighthNote ) ,
( E.|tō|\quarterNote , F.|men|\eighthNote ) ,
( F.|šū|\quarterNote , G.|ro|\eighthNote ) ,
( E.|yō|\quarterNote , F.|da'|\eighthNote )
\end{pmatrix}
</phonoRealization>
```

If repetitions are considered in the analysis, the <phonoRealization> element is added twice with different values for the *number* attribute, similarly to the <mrmr> element. For the first measure of the score shown in Figure 7, the result is as the following:

```
<phonoRealization number="1">
...
</phonoRealization>
<phonoRealization number="3">
...
</phonoRealization>
```

4.2.5 Vector Transcoding

The last step of the entire analysis algorithm consists of generating vectors out of the musical analysis. Using the values of the *snr* attributes of notes taken into consideration in the "*Metasyllabic Nuclear Reduction*", each two consecutive values are equal to a vector based on the following rules [6]:

$$\overrightarrow{\alpha\alpha} = \vec{p} \qquad \overrightarrow{\beta\beta} = \vec{s} \qquad \overrightarrow{\alpha\beta} = \vec{q} \qquad \overrightarrow{\beta\alpha} = \vec{r}$$

Two consecutive notes containing the *snr* attribute set, with the *mnr* attribute equal to "yes", are replaced by their corresponding vector based on their *snr* attribute values. Vectors generation takes place on both, a per measure basis and for the entire music. In the second case, which executes when all measures are analyzed, a vector generated by two symbols that belong to different measures, an opening parenthesis and a closing parenthesis precedes and succeeds the vector respectively. At last, the <vecTrans> element is added to both <music> and <measure> elements within MEI.

The following represents the result of the *Vector Transcoding* phase for both the first measure and the entire score shown in Figure 7.

$$(\vec{p}, \vec{q}, \vec{r})$$

```
<vecTrans>
\begin{pmatrix}
\vec{p} , \vec{q} , \vec{r}
\end{pmatrix}
</vecTrans>
```

$$(\vec{p}, \vec{q}, \vec{r}, (\vec{p}), \vec{p}, \vec{q}, \vec{r})$$

```
<vecTrans>
\begin{pmatrix}
\vec{p} , \vec{q} , \vec{r} ,
(\vec{p}) ,
\vec{p} , \vec{q} , \vec{r}
\end{pmatrix}
</vecTrans>
```

Like both the <mrmr> and <phonoRealization> element occurs twice with different values for the *number* attribute. For the first measure of the score shown in Figure 7, the result is as the following:

```
<vecTrans number="1">
...
</vecTrans>
<vecTrans number="3">
...
</vecTrans>
```

4.2.6 Anacrusis

One important aspect of the analysis is dealing with the anacrusis. A "*Syllabic Nuclear Reduction*" occurs on the measure of the anacrusis if it exists, however, unlike the process described earlier; this process applies in the inverse. The process starts by finding the notes from the end of the matrix and backwards. For the example shown in Figure 5, and using the matrix shown in the "*Metasyllabic Nuclear Reduction*" section 3.3.2, an eighth note is found at the end of the matrix. This note left as it is while rests replace the other notes in the matrix for the "*Morpho-phonological Rhythmic and Melodic Rewriting*". The matrix that contains *alphas* and *betas* is retrieved from the last measure in case of the absence of repetitions, or from the measure that returns to the beginning in the other case. The same process described earlier for both the "*Morpho-phonological Rhythmic and Melodic Rewriting*" and "*Phonological Realization*" executes, while replacing absent notes with rests, and replacing pitches in the latter phase by underscores. The result of the anacrusis analysis of the piece shown in Figure 7 would be as following in both Figure 8 and Figure 9:

$$N(E(\mu_0)) = N(\Lambda(\mu_0)R(\mu_0)) = N(\Lambda(\mu_0))R(\mu_0) =$$
$$(\alpha,\alpha,\beta,\alpha)\begin{pmatrix} \text{♩.} \\ \text{♩.} \\ \text{♩.} \\ \text{♩.} \end{pmatrix}$$
$$(\alpha,\alpha,\beta,\alpha)\begin{pmatrix} (\text{𝄽},\text{𝄾}) \\ (\text{𝄽},\text{𝄾}) \\ (\text{𝄽},\text{𝄾}) \\ (\text{𝄽},|\mathrm{Am}|\text{♪}) \end{pmatrix}$$
$$(\alpha.(\text{𝄽},\text{𝄾}),\alpha.(\text{𝄽},\text{𝄾}),\beta.(\text{𝄽},\text{𝄾}),\alpha.(\text{𝄽},|\mathrm{Am}|\text{♪}))$$

Figure 8. MRMR for the anacrusis.

$$((\alpha.\text{𝄽},\beta.\text{𝄾}),(\alpha.\text{𝄽},\beta.\text{𝄾}),(\beta.\text{𝄽},\beta.\text{𝄾}),(\alpha.\text{𝄽},\alpha.|\mathrm{Am}|\text{♪}))$$

$$E(\mu_0) =$$
$$((\_.\text{𝄽},\_.\text{𝄾}),(\_.\text{𝄽},\_.\text{𝄾}),(\_.\text{𝄽},\_.\text{𝄾}),(\_.\text{𝄽},C.|\mathrm{Am}|\text{♪}))$$

Figure 9. MRMR and PR for the anacrusis.

```
<mrmr>
\begin{pmatrix}
\alpha, \alpha, \beta, \alpha
\end{pmatrix}
\begin{pmatrix}
\quarterNoteDotted \\ \quarterNoteDotted \\
\quarterNoteDotted \\ \quarterNoteDotted
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha, \alpha, \beta, \alpha
\end{pmatrix}
\begin{pmatrix}
(\crotchetRest, \quaverRest) \\
(\crotchetRest, \quaverRest) \\
(\crotchetRest, \quaverRest) \\
(\crotchetRest, |Am|\eighthNote)
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha.(\crotchetRest, \quaverRest) ,
\alpha.(\crotchetRest, \quaverRest) ,
\beta.(\crotchetRest, \quaverRest) ,
\alpha.(\crotchetRest, |Am|\eighthNote)
\end{pmatrix}
\\
\\
\begin{pmatrix}
(\alpha.\crotchetRest, \beta.\quaverRest) ,
(\alpha.\crotchetRest, \beta.\quaverRest) ,
(\beta.\crotchetRest, \beta.\quaverRest) ,
(\alpha.\crotchetRest, \alpha.|Am|\eighthNote)
\end{pmatrix}
</mrmr>
<phonoRealization>
\begin{pmatrix}
(\_.\crotchetRest, \_.\quaverRest) ,
(\_.\crotchetRest, \_.\quaverRest) ,
(\_.\crotchetRest, \_.\quaverRest) ,
(\_.\crotchetRest, C.|Am|\eighthNote)
\end{pmatrix}
</phonoRealization>
```

5. EXPERIMENTS

5.1 Analysis Algorithm Evaluation

The algorithm discussed in this paper and implemented in JavaScript using NodeJS, contains two modules, one of them processing repetitions. It provides the possibility to choose whether to consider repetitions in the analysis or not alongside providing the necessary matrices for the process. Depending on the choice, the appropriate module executes. The algorithm accepts as an input an MEI document, and outputs another MEI document containing the "*grammar*" module elements and attributes with their appropriate values alongside a PDF file containing the piece rendered using Verovio [12] and SVG processing for placing *alphas* and *betas* above notes, with the entire analysis expressed in terms of mathematical expressions.

We chose twelve music scores from [6] for testing and evaluating the correctness of the implemented analysis algorithm. The correctness verification is the most important criteria monitored and evaluated during the testing procedure. The correctness measurement is phase based, meaning an analysis is not entirely wrong if an error occurs at only one phase. However, if an error exists in one phase that may affect the next one, the latter is considered correct if its output is correct considering the error caused by a previous phase.

The algorithm analyzes and encodes all measures. However, since the algorithm cannot yet analyze correctly other than the first measure when encountering a strophic song, an exception is made, evaluating only the first measure for strophic songs. While in the other case, the evaluation considered all measures. The phases are represented by P1, P2, P3, P4 and P5, which represent the "*Syllabic Nuclear Reduction*", "*Metasyllabic Nuclear Reduction*", "*Morpho-phonological Rhythmic and Melodic Rewriting*", "*Phonological Realization*" and "*Vector Transcoding*" respectively. The results for the conducted experiments are present in Table 2.

Instrumental Music					
Music	**P1**	**P2**	**P3**	**P4**	**P5**
Repetition Included					
Dārij Ḥijāz	✓	✓	✓	✓	✓
Repetition Free					
Dūlāb Rāst	✓	✓	✓	✓	✓
Syllabled Music					
Music	**P1**	**P2**	**P3**	**P4**	**P5**
Repetition Included					
Huwwāra	✓	✓	✓	✓	✓
Jibnā l-kibbī wi-l-ḥinnā	✓	✓	✓	✓	✓
Jibnā l-‘arūs uw-jīnā	✓	×	✓	✓	✓
Daḥṭō lō neḥtē	✓	✓	✓	✓	✓
Hymn A02	✓	✓	✓	✓	✓
Suġītō Qūm fawlōs	✓	✓	✓	✓	✓
Bo’ūtō dmor afrēm	✓	✓	✓	✓	✓
Repetition Free					
Dal‘ōnā	✓	✓	✓	✓	✓
Yā šamsi ḍallik šāriqā	✓	✓	✓	✓	✓
Yā šamsi ḍallik šāriqā	✓	✓	✓	✓	✓

Table 2. Results of the performed tests.

5.2 Discussion

Evaluating only the first measure for strophic songs, eleven out of twelve pieces analyses were correct, representing a 92% correctness.

The results presented in the section 5.1, show that the algorithm is able to analyze the majority of modal monodies. In the case of strophic songs, the evaluation takes into account the first measure only; otherwise, all measures were evaluated. The analyses of eleven pieces were completely correct, while only the analysis of “Jibnā l-‘arūs uw-jīnā” was not. This is due to an unsupported case that caused an incorrect output in the second phase, the “*Metasyllabic Nuclear Reduction*” phase, choosing α instead of a β. This is due to the lack of support for some cases rarely present in some pieces. Other cases exist as well, that are not supported and yet to be done.

6. CONCLUSION AND FUTURE WORK

This paper discussed a solution proposed for encoding traditional modal monodies generative grammar in MEI. The solution consists of adding a new custom module to the initial MEI schema, alongside developing an analysis algorithm for the extraction and encoding of the generative grammar. Three new elements and three attributes were added to the MEI schema by creating the grammar module. The implemented algorithm, analyzes the musical scores as per described in [6]. All mathematical expressions and matrices were expressed in TeX, making it easier to understand and render equations, while using the “lilyglyphs” TeX package in order to represent notes like quarter notes ♩ and eighth notes ♪ . The conducted experiments tested the correctness of the implemented analysis algorithm and the results are evaluated satisfactory.

The algorithm helps performing the analysis in an automatic and time saving manner. The results provided by the output of the algorithm may also be used to generate new modal monodies.

As future work, improvements are expected to enhance the analysis algorithm in order to implement specific identified cases in the modal semiotics theory [6] not still supported in the analysis algorithm.

In addition, one of feature is to import MusicXML documents and analyze them as MEI files. Finally, it is expected to automatically identify the input matrices needed for both the “*Syllabic Nuclear Reduction*” for instrumental scores and the “*Metasyllabic Nuclear Reduction*” phase.

7. REFERENCES

[1] “An introduction to MEI” [Online]. Available: http://music-encoding.org/about/

[2] A. Hankinson, P. Roland, and I. Fujinaga, “The Music Encoding Initiative as a Document-Encoding Framework.,” in *International Society for Music Information Retrieval Conference*, Miami, USA, 2011, pp. 293–298.

[3] L. Pugin, J. Kepper, P. Roland, M. Hartwig, and A. Hankinson, “Separating Presentation and Content in MEI.,” in *International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012, pp. 505–510.

[4] TEI Music SIG, “TEI with Music Notation.” [Online]. Available: http://www.tei-c.org/SIG/Music/twm/.

[5] The Text Encoding Initiative, “14 Tables, Formulæ, Graphics and Notated Music,” *The Text Encoding Initiative Guidelines*. [Online]. Available: http://www.tei-c.org/release/doc/tei-p5-doc/en/html/FT.html.

[6] N. Abou Mrad, *Eléments de sémiotique modale : Essai d’une grammaire musicale pour les traditions monodiques*, Paris and Hadat/Baabda: Editions Geuthner and Editions de l’Université Antonine, 2016.

[7] A. Marsden, “Software for Schenkerian Analysis,” in *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[8] Hamanaka M., K. Hirata, and Tojo S., “Implementing Methods for Analysing Music Based on Lerdahl and Jackendoff’s Generative Theory of Tonal Music,” in D. Meredith D (Ed.) *Computational Music Analysis*. Springer, 2016.

[9] I. Capelle and K. Richts, “Die Welt des Detmolder Hoftheaters – erschlossen mit MEI und TEI,” *Bibliotheksdienst*, vol. 50, no. 2, pp. 199–209, 2016.

[10] “Das Modell | Detmolder Hoftheater.” [Online]. Available: http://hoftheater-detmold.de/?page_id=1095.

[11] S. Leblond Martin (Ed.), *Musiques orales, notations musicales et encodages numériques / Oral music, music notation and digital encoding*. Paris: Les Éditions de l’Immatériel, 2016.

[12] L. Pugin, R. Zitellini, and P. Roland, “Verovio: A library for Engraving MEI Music Notation into SVG.,” in *International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014, pp. 107–112.

DEZRANN, A WEB FRAMEWORK TO SHARE MUSIC ANALYSIS

Mathieu Giraud
CRIStAL, CNRS, Univ. Lille, France
mathieu@algomus.fr

Richard Groult
MIS, UPJV, Amiens, France
richard@algomus.fr

Emmanuel Leguy
CRIStAL, Univ. Lille, France
emmanuel@algomus.fr

ABSTRACT

Music analysis on traditional scores is often based on *annotated* elements, such as patterns, harmonies, or sections. Music students, teachers, players, or researchers are used to annotate music and to discuss these analyses. Music lovers, even when they do not read music, also frequently talk about music and share their reaction to specific sections. We present Dezrann, an open-source web platform for music annotation and analysis on scores in traditional notation, developed with the Web Components through the Polymer framework. Dezrann enables to view, edit, and share music analysis through sets of *labels* on a score or a waveform. Labels are linked to musical positions and possibly to voices of the score, and can have a duration or not. They can be created or edited with simple mouse or finger gestures. A public server is available to test the application on Bach fugues and chorales as well as on Mozart string quartets.

1. INTRODUCTION

1.1 Music Annotation and Web Scores

Music analysis can be seen as "casting light upon music" [1], talking about music, possibly with an historical, comparative or aesthetic perspective. Any style of music can be analyzed – oral, notated traditionally or with other means, possibly stored with some electronic material – and hearing and feeling the music may be the first action of an analyst [2]. Music analysis is however well-established on *scores* in common music notation. Analyses on such scores are often grounded on *music annotation*, labeling on the score concepts such as patterns, harmonies, or sections.

How can digital tools help *visualize and annotate traditional scores*, or, more generally, to *talk about music*? One may add comments on scores by using *score editors and engravers* – either commercial ones, or open-source projects such as Lilypond [1], which puts a strong emphasis on music typography. By the way, many music teachers or scholars use score editors to prepare annotated materials for their talks or their courses.

[1] lilypond.org

Today, scores can directly be rendered and heard from a web browser, with some projects such as Guido [3], and, more recently, VexFlow [2], MaxScore with NetCanvas [4], and, developed in the MEI initiative, Verovio [5]. Sending music notation across web notation applications was studied by [6]. Web scores are also proposed together with for-profit services, for example by NoteZilla [3] or JellyNote [4].

Music annotation elements can be encoded in platforms such as iAnalyse [7] or INScore [8, 9]. Toolkits for computational music analysis such as Humdrum [10] or music21 [11] may also output annotated scores [12]. The MEI consortium put lots of efforts into modeling not only music but annotations and metadata, and platforms built on Verovio can offer both engraving and annotation capabilities, such as with the Verovio Humdrum Viewer [13].

These software may give access to other representations than scores, such as waveforms, and some allow exporting videos with scores, or even representing interactive scores. Other projects focus on providing annotated music content, helping the discovery of music concepts for music lovers, such as *Guides d'écoute de la Philarmonie de Paris* [5].

1.2 Who Needs to Annotate Music?

People may need to talk about music, or to annotate actual music content, notated on scores but not always:

- music pedagogues, teaching either in basic music education or in more specialized classrooms such as analysis, harmony, or composition;
- pupils or students of these classes, engaging into active participation [14];
- researchers in musicology, or in music information retrieval (MIR), more precisely in computational music analysis (CMA), focusing on manual, automated, or semi-automated music analysis;
- performers studying scores and preparing their own interpretation;
- general public or music lovers, sharing their reaction to music.

[2] vexflow.com
[3] notezilla.io
[4] jellynote.com
[5] media.citedelamusique.fr/guide-ecoute

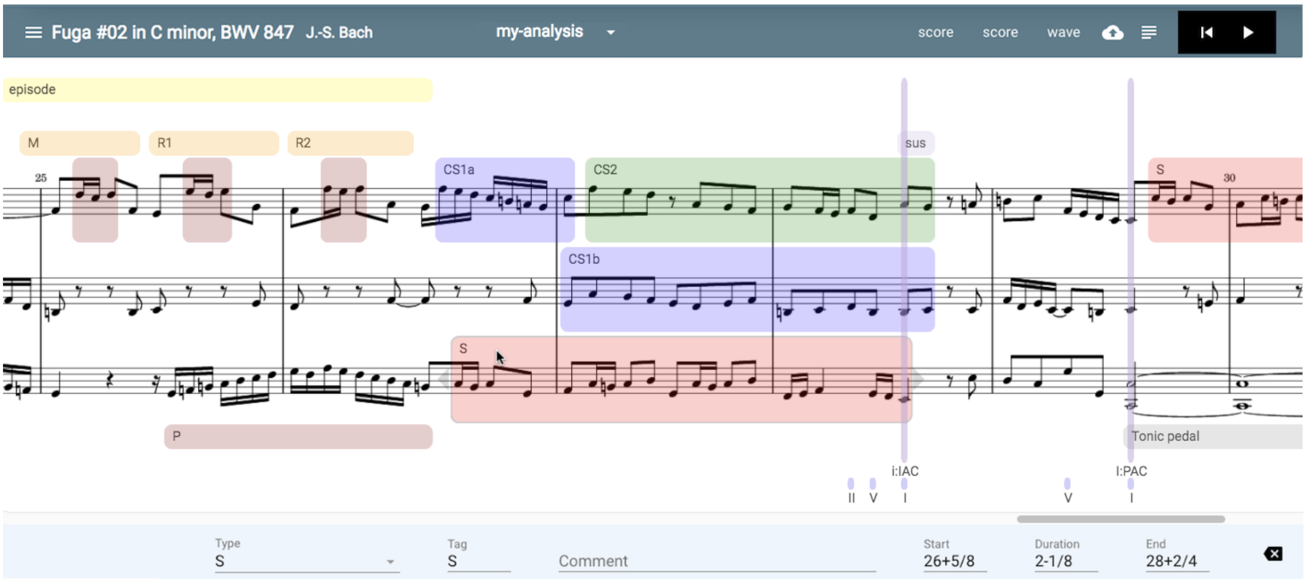


Figure 1. Current Dezrann web application (*<dezrann-app>* component, here loaded with a *<dez-music-canvas type="score">* component, see Section 3.1) showing an extract at the end of a Bach's fugue in C minor annotated with thematic patterns (subject S in blue, and counter-subjects CS in green and red) as well as cadences (IAC, PAC) and other labels. Such fugues were written for the keyboard and are usually presented on two staves. Here the three voices of the fugue are presented onto separate staves to easily label voice-related elements. The lower panel enables to edit the selected label, here a fugue subject starting at the 6th eighth (5/8) of the measure 26. The duration of the label, 2 measures minus one eighth, is computed including the first note but excluding the last note. The top panel displays piece information and enables to save or export the analysis, named here "my-analysis".

Most people are not experimented music readers. However, everyone likes to talk about music: A platform to discuss music should thus give convenient access to several representations (score, waveform) as well as audio rendering – such as the social network SoundCloud [6] that enables to annotate instants in waveforms.

1.3 Motivation and Contents

The software and platforms listed in the previous paragraphs have some strong points, and many of them bring innovation for traditional scores, and far beyond for some. Nevertheless, there has not been any open-source platform designed to *annotate scores with traditional notation* that is easily accessible on the web, enabling to visualize, to compare, to edit, and to save elements involved in music analysis. For example, researchers in MIR/CMA designing music annotation algorithms frequently need to "get back to the score" and to compare the output of their algorithms to actual music, or sometimes to existing analyses. This can be painful without adapted tools.

We present here Dezrann, an open-source web platform to view, to edit, and to share "music analyses" as annotations on scores in common musical notation, but also on waveforms. Dezrann was designed to help discuss music between people: We do not develop a music editor or engraving software – score rendering is a challenge already tackled by existing software – and focus here on the *annotation* part, allowing users to label music elements, either on a score or a waveform.

In the following, we thus present Dezrann from the point of view of users and developers (Sections 2 and 3), we detail the availability of the platform, the associated corpora, and the development roadmap (Section 4), and we conclude (Section 5).

2. USER PERSPECTIVE

Dezrann means "analysis" in Breton language, a Celtic language spoken in Brittany. Working with Dezrann is free and requires no software installation – the platform works with recent web browsers, at least with recent Safari, Firefox, and Google Chrome.

The user starts by selecting a piece inside available corpora (Figure 3) or by giving a URL. She lands on a "raw" score, or, when it is available, on an annotated score. For example, `dezrann.net/#/bwv847` leads the user to the Bach fugue in C minor BVW 847 with a pre-loaded analysis (Figure 1). The score is here split into three linear staves spanning the whole piece. The user can browse and possibly hear the score, and navigate where she wants.

To annotate music, she may click or point anywhere on the score to add or edit *labels*. Labels, as we defined in [12], are annotations on one or several staves that may have a duration, or not, such as patterns, cadences, harmonic markers, or structural elements. The creation of labels is done by *gestures*. Two gestures are now implemented:

[6] `soundcloud.com`

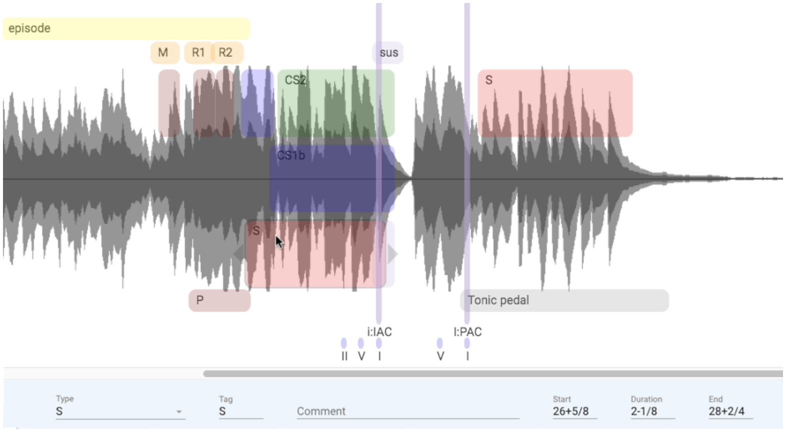


Figure 2. Annotation on a waveform through a *<dez-music-canvas type="wave">* component. The same extract than Figure 1 is displayed, with the same selected label. Labels are here synchronized both to audio time and to musical symbolic time, allowing to see or edit these values in musical time. The invisible grid for editing the labels is here on beats, although more precise values can be entered through the label editor in the bottom panel.

- A left-to-right gesture creates a label with some duration. Such labels can be created either on a staff, to highlight a musical element in one voice, or on the spaces above or below the score, for example to annotate concepts related to the structure or the overall harmony;
- A top-to-bottom gesture across the staves creates a label with no duration, that can be used for example for cadences or as a section marker.

The user can also create labels by using buttons (Figure 4) corresponding to label presets. These labels can be created on the fly while the music plays.

Labels coordinates (start, end, and duration) are shown in symbolic *musical time*, relatively to bar and beat. They can be edited:

- Either on the score or the waveform, by dragging them or by extending them to the left or to the right. These operations snap on an invisible grid built on beats, or on onsets and offsets of the notes appearing in the score;
- Or by editing text fields, entering values such as `23` (first beat of measure 23), `25-1/4` (one beat before the measure 25, works even on ternary measures), or arbitrary float values such as `26.75` or even `26.71`, allowing more flexibility in label placement.

When an audio file and its associated synchronization data are both available, the user can switch back and forth between the score view and the waveform view (Figure 2), still preserving the labels with their musical time. The *analysis*, that is the annotation made as a set of labels, can be saved for later usage and shared with other people.

The user may share links to scores through permanent URLs. Going to `dezrann.net/#/bwv847#28.5` jumps to the extract displayed on Figures 1 and 2, around the half of the measure 28 of Fugue BWV 847 in C minor – this is where there is an incomplete authentic cadence concluding the *strettos* and preceding the final cadence. Discussing this passage inside her talk or lesson, the user may thus share a link to his audience.

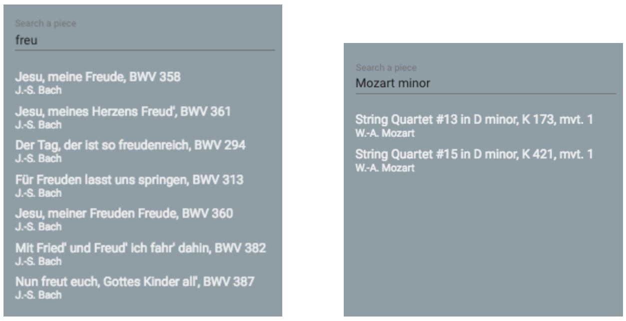


Figure 3. Corpora querying from *<dez-corpus>*.

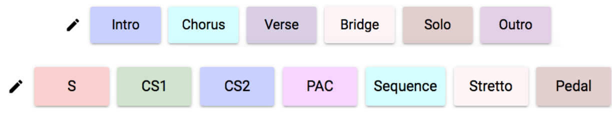


Figure 4. Label bars for annotating song structures (top) and fugues (bottom). Each button triggers a label with a preset type, tag and comment. Label presets can be edited by the user.

3. UNDERLYING TECHNOLOGY

3.1 Dezrann Components

The Dezrann web client is written in object-oriented Javascript ES6, and uses the *Web Components* model through Polymer 2.0 framework [7]. Proposed in 2011, Web Components *"are a set of features being added by the W3C to the HTML and DOM specifications that allow for the creation of reusable widgets or components in web documents and web applications. (...) The components model allows for encapsulation and interoperability of individual HTML elements"* [8]. Within this model, new "components" can be used as HTML tags such as the *<audio>* or *<video>* HTML5 tags. We implemented the following components (Figure 5):

- *<dez-analysis>* stores the set of labels. It can be imported or exported into a `.dez` json file (Figure 6).

[7] `polymer-project.org`
[8] `en.wikipedia.org/wiki/Web_Components`

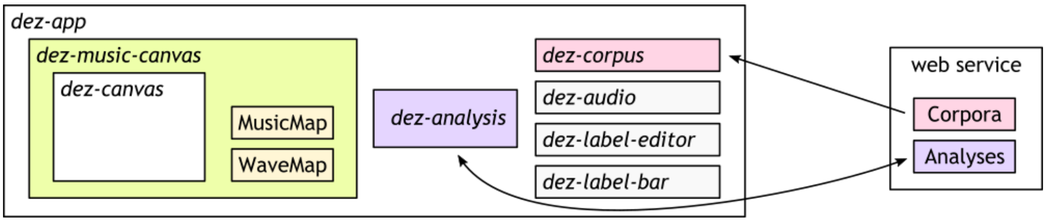


Figure 5. Dezrann architecture, see details in the text. The client side (the *<dezrann-app>* component) communicates with the web service both to get corpus files and to load and store analyses.

```
"labels": [
  { "type": "Structure", "tag": "episode",    "start": 84,    "duration": 18,    "line": "top.1" },

  { "type": "Pattern D",                      "start": 96.5,  "duration": 0.5,   "staff": 1 },
  { "type": "Pattern D",                      "start": 98.5,  "duration": 0.5,   "staff": 1 },
  { "type": "Pattern D",                      "start": 100.5, "duration": 0.5,   "staff": 1 },

  { "type": "Harmonic sequence", "tag": "M",  "start": 95.75, "duration": 1.75 },
  { "type": "Harmonic sequence", "tag": "R1", "start": 98,    "duration": 1.75 },
  { "type": "Harmonic sequence", "tag": "R2", "start": 100,   "duration": 1.75 },

  { "type": "CS1",                            "start": 104,   "duration": 6.5,   "staff": 2 },
  { "type": "S",                              "start": 102.5, "duration": 7.5,   "staff": 3 },

  { "type": "Pedal", "tag": "I",              "start": 114,   "duration": 10,    "line": "bot.1" },

  { "type": "Cadence", "tag": "I:IAC",        "start": 110,                      "line": "all" },
  { "type": "Cadence", "tag": "I:PAC",        "start": 114,                      "line": "all" },

  { "type": "Degree", "tag": "II",            "start": 109,                      "line": "bot.3" },
  { "type": "Degree", "tag": "V",             "start": 109.5,                    "line": "bot.3" },
  { "type": "Degree", "tag": "I",             "start": 110,                      "line": "bot.3" },
  ...
]
```

Figure 6. Dezrann analyses, here represented as a `.dez` json file, are collection of *labels*. These lines correspond to some labels displayed on the Bach fugue on the Figures 1 and 2. Each label has a onset, an optional duration, a type, optional tags and comments, and may have optional staff or line information. Onsets and durations are stored in musical time.

The files can also be read and written through our extension to music21 [12]. Label types are expendable by editing the `lib/LabelTypes.js` file.

- *<dez-canvas>* handles SVG graphical objects on top of the `snap.svg` library [9] (while being agnostic to the musical significance of these objects), as well as generic mouse/pointer gestures and interaction with these elements for label creation and edition. Rectangles and triangles markers are now defined. Labels are put onto one or several horizontal *lines*, each line being defined by a pair of y-positions. The x-positions of the labels can be snapped to an optional grid with the graphical positions of the notes.
- *<dez-music-canvas>* embeds a *<dez-canvas>* and handles music content, either a score or a waveform, with proper handling of x- and y- positions.
  - *<dez-music-canvas type="score">* handles a score image. The default view is now produced by Lilypond extended with scripts borrowed from ly2video [10]. These scripts output graphical x-positions of the notes, linked to their musical times, as well as y-positions of the staves.
  - *<dez-music-canvas type="wave">* handles a waveform, always with proper handling of x- and y- positions. Now the regular x-positions of the waveform are mapped, through synchronization data of *<dez-audio>* (see below and Figure 7), to the musical x-positions.

Both the *score* and *wave* views set the grid of the underlying *<dez-canvas>* component, such that label creation or edition with the mouse or the finger are snapped to musical positions.

- *<dez-label-editor>* enables to see and edit properties of one label. It is focused on the musical time, even linked to *<dez-music-canvas type="wave">*, keeping a symbolic view on audio data.
- *<dez-label-bar>* eases the creation of labels with buttons encoding a given preset (type, tag and comments).
- *<dez-audio>* handles audio output to play .mp3 or other formats through the Web API Audio. It works on its own but is better handled with a manual (or automatic) synchronization file between musical time and audio time (see Figure 7). The synchronization points can be located at each beat, at each measure, or even at only a few places in the piece. Moreover, a MIDI output (processed through `midi.js`, under

[9] `snapsvg.io`
[10] `github.com/aspiers/ly2video`

```
[
  { "onset" :   0, "time" :   0.45 },
  { "onset" :  40, "time" :  34.40 },
  { "onset" :  80, "time" :  67.91 },
  { "onset" : 117, "time" : 100.89 },
  { "onset" : 118, "time" : 101.91 },
  { "onset" : 119, "time" : 103.01 },
  { "onset" : 120, "time" : 104.18 }
]
```

Figure 7. Synchronization between musical and audio time. Synchronization points can be given at any resolution. In this piece, the tempo is somewhat regular, except for a slow-down at the last four offsets.

development) is available as soon as there is a score, and further allows hearing a selected voice.

- *<dez-corpus>* provides a searchable view on a corpus of pieces (Figure 3).
- *<dezrann-app>* is the single web page application that wraps all previous components to display one piece with one analysis. The analysis is a `.dez` json file (Figure 6) that can be either saved on the web service or exported. Analyses can be loaded and saved. Moreover, the component recognizes URLs to select a piece and optionally to link to a particular position in the piece by feeding its underlying components.

Components can be used as simple HTML elements, and created and modified with some attributes. For example, a *<dez-music-canvas not-editable>* component displays a read-only annotated score, without any edition or saving capabilities. Note that such a *<dez-music-canvas not-editable>* component fully works on a mobile device and allows scrolling with fingers. Some attributes can be set or changed even after the component has been created. Consider for example *<dez-music-canvas onset=114>*: Setting or changing the *onset* attribute by JavaScript from the outside updates the *<dez-music-canvas>* and jumps to the given offset without reloading the component.

The two synchronizations – graphical score to musical time, and audio to musical time – enable flexible combinations of the components. For example, one can change the score image, using other parameters or even another score rendering engine, and still conserve the synchronization to audio.

Note that *<dez-audio>* can even be used without any reference to any score: Labels are then referenced by their audio time. Should an audio/musical time synchronization be later available, the labels can then converted back to properly reference the musical time.

3.2 Corpus and Analysis Web Service

A simple web service, written in `node.js`, provides a set of corpora and interactions through AJAX requests (Figure 5, right):

- The *<dez-corpus>* component queries the piece list from the web service. It then feeds the required files for a piece (images, audio, positions and synchronizations) to *<dezrann-app>* and the underlying components;
- For a given piece, *<dezrann-app>* uploads and downloads analyses into the underlying *<dez-analysis>*.

4. AVAILABILITY AND ROADMAP

The web platform with developer and web service documentation [11] as well as some corpora [12] are available under open-source licenses (GPLv3+ for the code, ODBL for the data). Components are distributed with usage examples allowing individual reuse in other projects. Tests include 130 unit tests and 40 functional tests using the Mocha toolkit and Selenium [13], [14].

4.1 Platform Availability and Roadmap

Following suggestions on a first prototype of Dezrann [15], the platform evolved, most notably with a complete refactor leading to a clean separation between the *<dez-canvas>* on one side, and the *<dez-music-canvas>* on the other side. This separation helped to implement the *<dez-music-canvas type="wave">* component and to expand the set of labels.

The components and the application can now be used from the public server `dezrann.net`. The MIDI parts of *<dez-audio>* are still under development.

Beside generic improvements to the ergonomy and the features, notable perspectives include:

- *user accounts and flexible access patterns.* Authentication is now limited to a custom installation of Dezrann behind protected pages. We plan to implement a more generic authentication scheme to run scenarios needing fine-grained access to the analyses through *roles* such as teacher or student. For example, a homework might be assigned, needing first individual or group analysis, then returned and discussed or compared with other analyses such as a reference analysis provided by the teacher.
- *real-time remote collaboration.* We would like to allow simultaneous users to use Dezrann over the network through the use of `socket.io` library. This could make the platform evolve towards a "social network" to annotate music.

We welcome ideas or contributions from other groups and are ready to help the links between Dezrann components and third-party code. In particular, new links with automated or semi-automated analysis pipelines could be worthwhile.

[11] `dezrann.net/dev`
[12] `algomus.fr/datasets`
[13] `mochajs.org`
[14] `seleniumhq.org`

Bach four-voices chorales	2 staves (2 voices each)	181 pieces
Bach fugues, first book of the *Well-Tempered Clavier*	2 to 5 staves	24 annotated pieces, > 600 labels [16] including 12 with synchronized audio
Mozart string quartets, sonata form movements	4 staves	28 annotated pieces, > 500 labels [18]

Table 1. Corpus and annotations available through Dezrann.

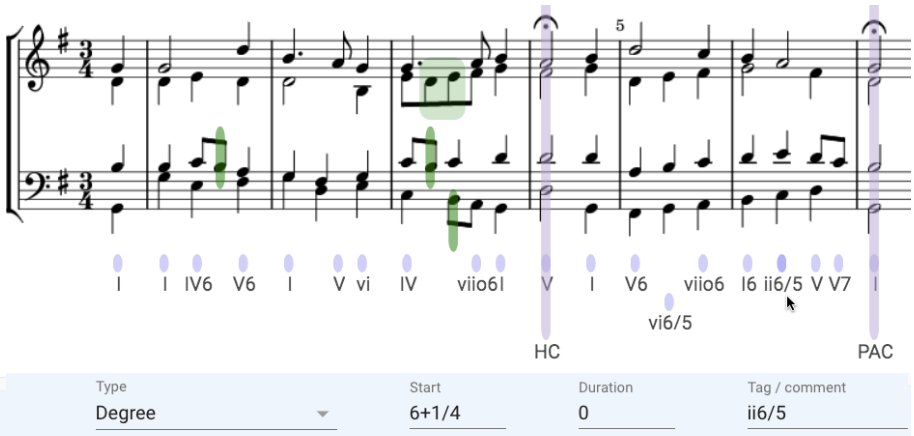


Figure 8. Annotation of the start of the four-part Bach chorale “Aus meines Herzens Grunde” (BWV 269), showing degrees coming from the `riemenschneider001` analysis from music21, imported by a script based on [12], and completed by cadences, neighbor and passing notes, annotated from within Dezrann. The resulting analysis can be saved or exported for future use.

4.2 Corpus Availability and Roadmap

Table 1 lists the available corpora on the public Dezrann server. As the current graphical capabilities of the platform are to put labels on staves, we focus on music with distinct voices – hence string quartets, or voice-separated fugues. We also included 4-voices chorales laid out on 2 staves, drawing labels on each half of the staff (Figure 8).

More than 200 scores are now available on the public server. New pieces are gradually included, and we target 1,000 scores available by the end of 2018, notably using corpora available through music21 [11]. Individual pieces or corpora can be added by the `piece.py` and `corpus.py` scripts on the server side. Perspectives include to let the user upload score files (MIDI, MusicXML, MEI, or **kern), possibly with audio files.

Dezrann is made in order that people experiment their own annotations and analyses. However, we also gather reference analyses and make them available through the platform:

- There are now analyses of the fugues of the first Book of Bach’s *Well-Tempered Clavier*. Previously published in [16], this dataset was adapted to match Dezrann format by adapting the extension [12] of the python music21 framework [11].

- Colleagues encoded within Dezrann cadences and structural markers in Mozart string quartets [17] in an ongoing study on sonata forms [18], allowing us to benefit from their feedback in developing the platform.

5. CONCLUSION

“Look at the chord progression at measure 42!” Such a comment – sometimes hard to follow – is often heard in music analysis or harmony classrooms, in music lectures, when discussing on music scores between colleagues, or when practicing into a band. Even if we love paper scores, pencil and erasers, even if we love music teaching through oral practice, we believe that new numeric tools may be pertinent to navigate easily through scores, to share people’s view, possibly different, on a same score, and to benchmark or discuss results from automated or semi-automated software. We designed Dezrann in this spirit, hoping to propose a modern and efficient way to talk about music encoded onto traditional notation.

We welcome comments or contributions from developers, users, music and MIR researchers, teachers, students, or pupils. to make the platform evolve for everyone needs, possibly with some custom components. Tests in music classrooms of a secondary school (age 10–13) are scheduled in relationship with the *inspection académique* (local education authority) supervising music teachers in Amiens district.

Acknowledgments

This project is partially funded by French CPER MAuVE (ERDF, Région Hauts-de-France) and by a grant from the French Research Agency (ANR-11-EQPX-0023 IRDIVE). We thank all users of Dezrann for their feedback and suggestions.

6. REFERENCES

[1] I. Bent and W. Drabkin, *Analysis*, 1987.

[2] F. Delalande, *Analyser la musique, pourquoi, comment ?* INA, 2013.

[3] H. H. Hoos, K. A. Hamel, K. Renz, and J. Kilian, “The GUIDO music notation format,” in *Proceedings of the International Computer Music Conference (ICMC’98)*, Ann Arbor, USA, 1998, pp. 451–454.

[4] B. Carey and G. Hajdu, “Netscore: An image server/client package for transmitting notated music t. browser and virtual reality interfaces,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR’16)*, Cambridge, UK, 2016.

[5] L. Pugin, R. Zitellini, and P. Roland, “Verovio: A library for engraving MEI music notation into svg,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 107–112.

[6] S. James, C. Hope, L. Vickery, A. W. B. Carey, X. Fu, and G. Hajdu, “Establishing connectivity between the existing networked music notation packages,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR’17)*, A Coruña, Spain, 2017.

[7] P. Couprie, “iAnalyse : un logiciel d’aide à l’analyse musicale,” in *Actes des Journées d’Informatique Musicale (JIM’08)*, Albi, France, 2008, pp. 115–121.

[8] D. Fober, C. Daudin, Y. Orlarey, and S. Letz, “Partitions musicales augmentes,” in *Actes des Journées d’Informatique Musicale (JIM’10)*, Rennes, France, 2010.

[9] G. Lepetit-Aimon, D. Fober, Y. Orlarey, and S. Letz, “INScore expressions to compose symbolic scores,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR’16)*, Cambridge, UK, 2016.

[10] D. Huron, “Music information processing using the Humdrum toolkit: Concepts, examples, and lessons,” *Computer Music Journal*, vol. 26, no. 2, pp. 11–26, 2002.

[11] M. S. Cuthbert and C. Ariza, “music21: A toolkit for computer-aided musicology and symbolic music data,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, Netherlands, 2010, pp. 637–642.

[12] G. Bagan, M. Giraud, R. Groult, and E. Leguy, “Modélisation et visualisation de schémas d’analyse musicale avec music21,” in *Actes des Journées d’Informatique Musicale (JIM’15)*, Montréal, Canada, 2015.

[13] C. S. Sapp, “Verovio Humdrum Viewer,” in *Music Encoding Conference (MEC)*, Tours, France, 2017.

[14] E. Venn, “Watch this space: On blended learning and music analysis in the classroom,” in *European Music Analysis Conference (EuroMAC)*, Strasbourg, France, 2017.

[15] E. Leguy, R. Groult, and M. Giraud, “Une application web interactive pour l’analyse musicale,” (poster) in *European Music Analysis Conference (EuroMAC)*, Strasbourg, France, 2017.

[16] M. Giraud, R. Groult, E. Leguy, and F. Levé, “Computational fugue analysis,” *Computer Music Journal*, vol. 39, no. 2, 2015.

[17] M. Flothuis, *Mozarts Streichquartette: Ein musikalischer Werkfhrer*. C. H. Beck, 1998.

[18] L. Bigo, L. Feisthauer, M. Giraud, R. Groult, E. Leguy, and F. Levé, “Learning sonata form structure on Mozart string quartets,” *in preparation*.

SYMBOLIST: AN OPEN AUTHORING ENVIRONMENT FOR USER-DEFINED SYMBOLIC NOTATION

Rama Gottfried
CNMAT, UC Berkeley, USA
IRCAM, Paris, France / ZKM, Karlsruhe, Germany
HfMT Hamburg, Germany
rama.gottfried@berkeley.edu

Jean Bresson
IRCAM – CNRS – Sorbonne Université
UMR STMS, Paris, France
jean.bresson@ircam.fr

ABSTRACT

We present SYMBOLIST, a graphic notation environment for music and multimedia. SYMBOLIST is based on an Open Sound Control (OSC) encoding of symbols representing multi-rate and multidimensional control data, which can be streamed as control messages to audio processing or any kind of media environment. Symbols can be designed and composed graphically, and brought in relationship with other symbols. The environment provides tools for creating symbol groups and stave references, by which symbols maybe timed and used to constitute a structured and executable multimedia score.

1. INTRODUCTION

Contemporary art and music productions frequently rely on automated computer processes with huge sets of data and control parameters; and as in other large-scale data-driven situations, the authoring tools, storage and performance of the data are key design factors which have a marked influence on the aesthetic framework used to compose the artwork [1, 2]. Unlike pen and paper, commercial software authoring tools have been designed based on a set of use-cases and decisions about the composition format and rendering, selected and put forward by different actors in their development process. This situation prompts the question: If tools are a shaping factor in art production, how should authoring environments for artistic production be designed? In what ways can a computational process or mechatronic movement be represented in a score so that it is freely "composable", without presupposing a specific use context, or grammar?

While computational tools for creating and parsing symbolic graphic information are readily available, composition environments which support visualizing, editing, and synchronously executing multimedia control data streams are few to none. There exist no actual notational convention on how to represent control data for computerized automation systems [3, 4]. In electronic music production, most often the "score" is authored in a *digital audio workstation* (DAW) with MIDI note events and breakpoint function automations; while in theater, a *show control* system is typically used to step through a series of cues which send control messages to stage and lighting mechanisms. These tools have proved useful through their longevity over 30+ years, however as compositional frameworks, they proscribe specific ways of thinking about data. Breakpoint function automation works well for situations where you want to control *one* parameter over time, but in multivariate situations, for example spatial location where a position is a vector $\{x, y, z\}$, splitting the values into three separate automation lanes obscures the meaning of the values. [1] In contrast, a well designed symbolic notation could allow users to represent many parameters simultaneously [5].

The SYMBOLIST project addresses these issues by providing composers and media artists with a context-free environment for the authoring of graphical symbolic notation, with tools for displaying, editing and generating arbitrary streams of OSC-encoded data. After a general presentation of the project (Section 2), we will describe the design features and user interface of the software (Section 3), and then detail the execution mechanisms behind its score structure (Section 4). In continuation we will present some use cases and integration in host environments (Section 5), and conclude with an open discussion and some considerations about future work directions (Section 6).

2. FOUNDATIONS

SYMBOLIST was designed to address the practical need of visually representing parameters of electronic performances involving dense streams of control data, first conceived in the context of composing for spatial audio systems [6, 7]. High-dimensional symbolic representation is common in contemporary instrumental writing, and so for many composers it is intuitive to also apply symbolic notation approaches to new kinds of "multimedia instruments".

A first working prototype was implemented using Scalable Vector Graphics (SVG) authored with graphic design software (Adobe Illustrator), which could then be interpreted and performed as a stream of OSC data (Open Sound Control [8]) in the Max environment [9]. By leveraging the tools of a professional graphic design program in connection with the widely supported networking capabilities of OSC, the SVG-OSC project [10] provided a functional model of how graphic objects could be labeled and grouped

[1] The mathematical representation has the same perceptual problem.

semantically in order to be processed by an interpretive engine and used to control multimedia renderers. Building on the SVG-OSC project research, SYMBOLIST integrates the editing and semantic assignment functions into a single workspace specifically designed for maximum flexibility, through a minimum number of predefined object definitions.

SYMBOLIST considers a *score* as a structured set of graphical symbols, where each *symbol* (basic, or compound group of symbols) exists internally as an OSC *bundle* (i.e. a set of OSC messages describing a consistent data structure) which potentially includes both graphical attributes and other musical or control parameters. Although structured on the surface through staves, groupings and nested symbols (as we will see in the next section), the score is therefore viewed (and stored) as a simple, flat and *executable* sequence of OSC bundles.

The SYMBOLIST environment was implemented as a C++ application and built using the Juce framework.[2] It can run as a standalone editor or as an embedded component in another programming environment such as Max (where it constitutes a persistent container — a score — to display, edit and monitor control data streams) or OpenMusic [11] (where scores can be generated and processed through visual programs and algorithms).

3. WORKING IN SYMBOLIST

From the user point of view, the current SYMBOLIST prototype essentially implements a set of utilities for symbol authoring and composition following standard vector-graphic editing techniques.

Symbols. Graphical symbols and their associated semantics are defined by the user through interactive graphic and text-based OSC editing tools. Figure 1 shows a sample view of the main SYMBOLIST window. The left sidebar displays a number of default atomic symbol models (circle, rectangle, triangle, text characters...) which the user can pick and use as templates for the creation of symbols in the score page. On Figure 1, a single, big triangle symbol was added to the score. Score symbols are editable interactively using standard graphic transforms (translation, scaling, rotation, copy/paste, etc.). Their attributes may also be edited directly in the inspector view at the right of the window.

As mentioned above, each symbol is stored as an OSC bundle (i.e. a set of OSC messages), which reflects the set of attributes visible on the inspector view. The basic attributes shared by all symbols are: the *name*, symbol *type*, position (*x*, *y*), size (*w*, *h*), *color*, *staff* assignment, and *id*, a unique identifier of the symbol within the score.[3] Symbols may also include additional attributes. For example, the triangle symbol in Figure 1 includes *fill*, *stroke thickness*, and *rotation* attributes. The listing below displays the OSC representation corresponding to this symbol.

[2] https://juce.com/

[3] By default the *name* value is same as the *type*, and the *id* is the *name* followed by a unique instance number. Once a user-defined *name* is given, the *id* is updated.

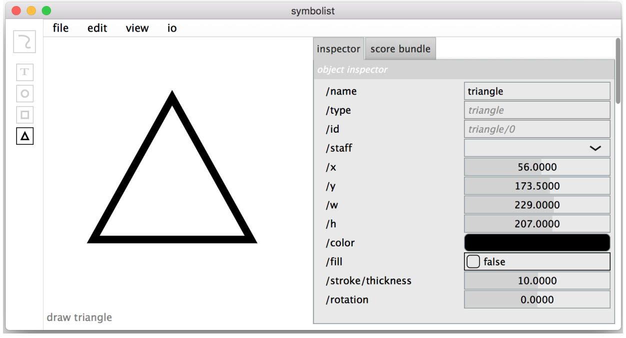


Figure 1. A single triangle symbol in the SYMBOLIST window. The inspector on the right side displays the attribute values of the symbol.

```
{
        /name : "foo",
        /type : "triangle",
        /id : "foo/0",
        /staff : "",
        /x : 47.,
        /y : 134.5,
        /w : 123.,
        /h : 120.,
        /color : [0., 0., 0., 1.],
        /fill : 0,
        /stroke/thickness : 2.,
        /rotation : 0.
}
```

Custom shapes can be drawn and edited using control point handles, and are encoded as *paths*, defined as a sequence of linear, quadratic or cubic bézier curve segments (see Figure 2). The SVG standard is used for storing path drawing commands in string format [12].

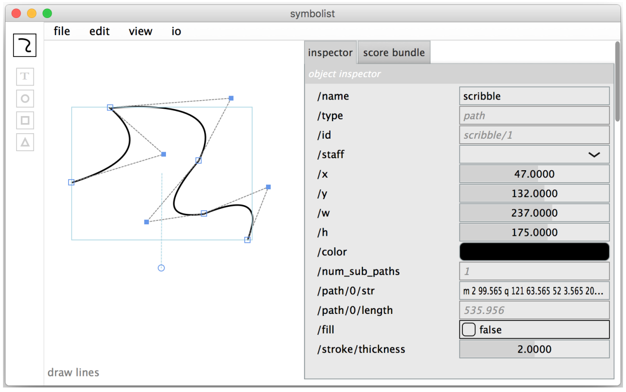


Figure 2. Drawing a custom (*path*) symbol.

The symbol in Figure 2 is represented in OSC as follows:

```
{
    /name : "scribble",
    /type : "path",
    /id : "scribble/1",
    /staff : "",
    /x : 33.,
    /y : 83.,
    /w : 237.,
    /h : 175.,
    /color : [0., 0., 0., 1.],
    /path/str : "m 2 99.565 q 121 63.565 52 [...]",
    /path/length : 535.956,
    /fill : 0,
    /stroke/thickness : 2.
}
```

Templates. Any symbol in the score can be turned into a template via a simple keyboard shortcut. Newly created templates appear in the symbol palette of the left sidebar (see Figure 3). They can then be stored in the application data and potentially shared between scores and projects.

In addition to the set of atomic symbols mentioned previously in this section, user-defined template symbols may therefore be selected and copied anywhere in the score as a new symbol, with all of the same editing and transformation possibilities.

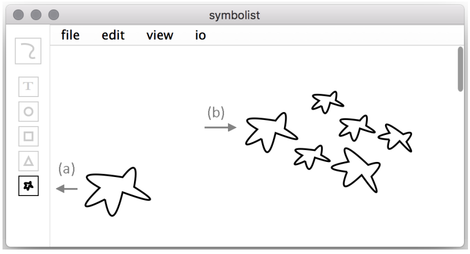


Figure 3. (a) Storing a user symbol as template in the SYMBOLIST palette toolbar (left side of the window). (b) Using this template as a model for creating new symbols.

Compound symbols can be created by graphical composition of simpler ones, through the *grouping* command. Symbols (atomic, custom, or compound) selected for grouping are gathered and converted into a single new symbol (see Figure 4), which can then be positioned, edited, transformed individually, and/or turned into a template in the symbol palette.

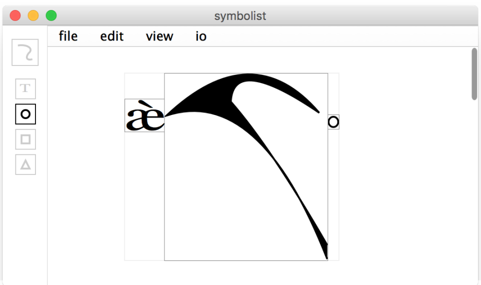


Figure 4. Grouping symbols.

Grouping is a hierarchical operation of unlimited depth and complexity. After grouping, sub-group symbols can still be accessed, recomposed and edited individually at any time, using simple user operations to step through the hierarchy of compound symbols. Below is an excerpted example of OSC representation of a SYMBOLIST *group* symbol, corresponding to the symbol in Figure 4:

```
{
    /name : "group",
    /type : "group",
    /id : "group/0",
    /staff : "",
    /x : 78.,
    /y : 46.,
    /w : 240.,
    /h : 209.,
    /color : [0., 0., 0., 1.],
    /numsymbols : 3,
        /subsymbol/1/name : "path",
        /subsymbol/1/type : "path",
        /subsymbol/1/id : "path/0",
        /subsymbol/1/staff : "",
        /subsymbol/1/x : 45.,
        /subsymbol/1/y : 0.,
        [...]
        /subsymbol/2/name : "text",
        /subsymbol/2/type : "text",
        /subsymbol/2/id : "text/0",
        /subsymbol/2/staff : "",
        /subsymbol/2/x : 0.,
        /subsymbol/2/y : 51.5,
        [...]
        /subsymbol/3/name : "circle",
        /subsymbol/3/type : "circle",
        /subsymbol/3/id : "circle/0",
        /subsymbol/3/staff : "",
        /subsymbol/3/x : 225.,
        /subsymbol/3/y : 54.5,
        [...]
}
```

Staves and score structure. In order to structure symbols into a temporal score, *staff* symbols can be created from any existing symbol (simple or compound). A *staff* symbol is considered as a reference which can be used for global manipulations and creation of polyphonic scores. It is wrapped in a special *staff* OSC bundle, and is automatically assigned time values (see Figure 5).

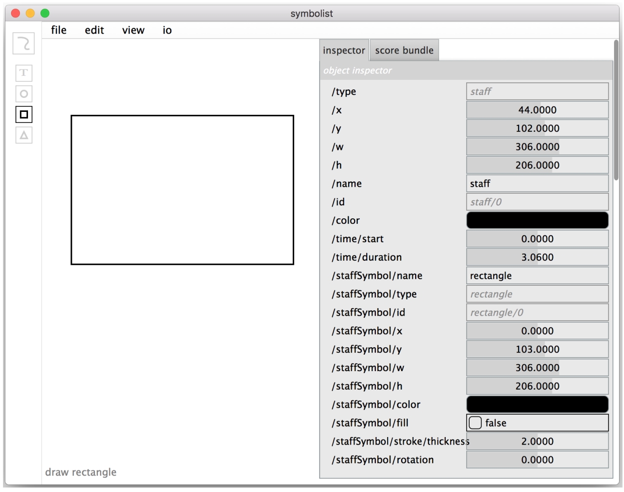


Figure 5. Converting a symbol to a *staff*.

Any symbol can be attached to a stave by setting the */staff* attribute to link to an existing *staff* symbol *id* value. All non-*staff* type symbols include the */staff* attribute in their corresponding OSC bundle. Once a symbol has been linked to a *staff*, this symbol becomes *timed*: it is given a start and duration, to its position and size relative to the stave origin and the stave numbering (see Figure 6).

Stave start and duration values are currently determined by their sequential order on in the score, also following traditional stave system format, reading left to right in lines down the page, and then continuing at the top of the following page. For traditional left to right, top to bottom reading, the symbol's start time and end times are calculated using the left and right edges of the object's bounds. [4]

[4] In the future, we envisage time direction could be a user-definable parameter in the score, for example to facilitate the use of Labanotation [13] or other types of graphic time arrangements such as trajectories through the score, and so on.

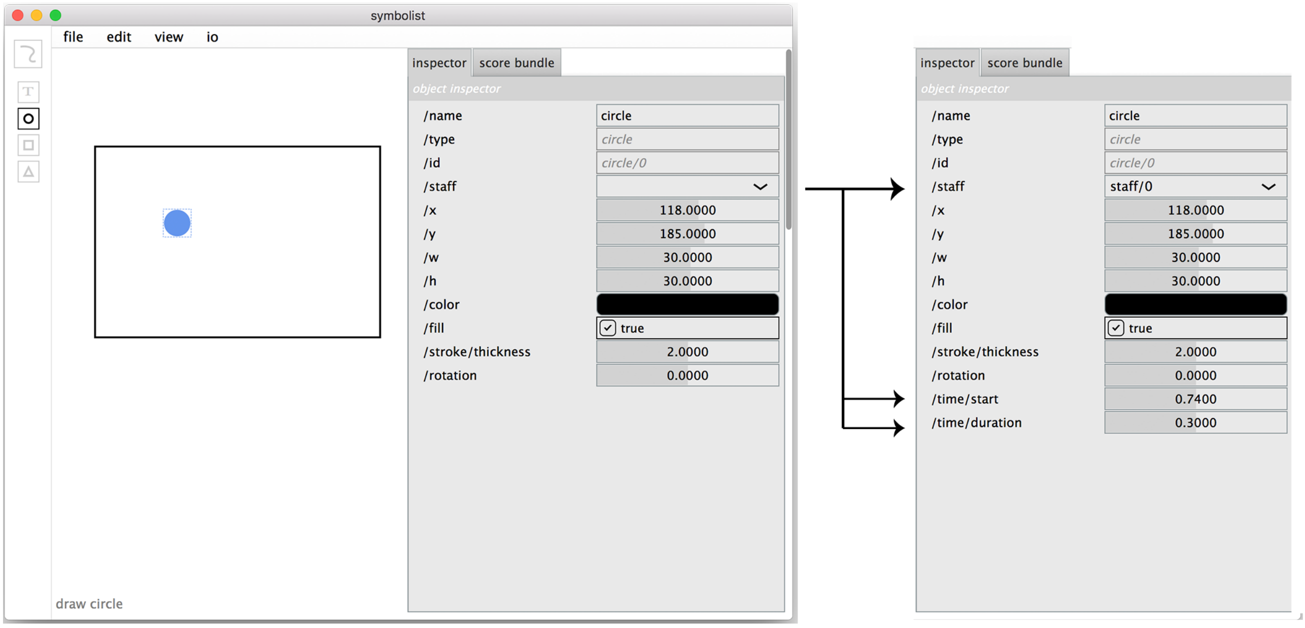


Figure 6. After attaching a symbol to a staff, time information is added to the symbol's OSC bundle.

Depending on the context, other internal parameters could also be effected by the symbol's horizontal and vertical coordinates in the stave reference, for example effecting pitch values for notes on traditional scores.

4. TIME AND SCORE PERFORMANCE

Through the creation of time relationships between symbols and staves, the score becomes "executable", or "performable". As described above, staves are the key temporal marker for score performance: they embed a time referential and a time-map allowing the computation of absolute time from relative graphical distances.

For the performance of the score, SYMBOLIST provides methods for outputting control values as OSC, and includes visual feedback information such as highlighted display for play-heads or cursors, etc.

SYMBOLIST actually does not include its own scheduling engine, but functions by responding to external time requests — e.g. from host environments — in order to retrieve the active symbol(s) at a given time. In response to a time location query, SYMBOLIST outputs an OSC bundle containing the values of all symbol "events" existing at that time in the score (see Figure 7).

To aid with mapping, the output OSC bundle is formatted using the symbol's *name* attribute as user defined identifier. For example in Figure 7, the staff name is "foo" and the group symbol name is "glissnote", which contains "glissando" and "notehead" sub-symbols. Whereas the score is a flat array of symbols/bundles, the contents of the output bundle are formatted in a hierarchical representation, where events are located in the OSC namespace of their associated stave. For example, in Figure 7 note that the active voices in the bundle, are in prefixed by */staff/foo*.

Each event is output with the relative time position within the symbol called the */time/ratio*, where 0 is the beginning of the symbol and 1 is the end.

To assist in handling overlapping polyphonic symbols, which may start and stop independently, a *voice* identifier is assigned to each symbol which stays constant between lookup queries. A */state* value is also provided which identifies the symbol's status: 1 for a new *voice*, 0 when it is continuing from the last lookup, and -1 to identify when a *voice* is no longer present, which can be used for "note off" messages.

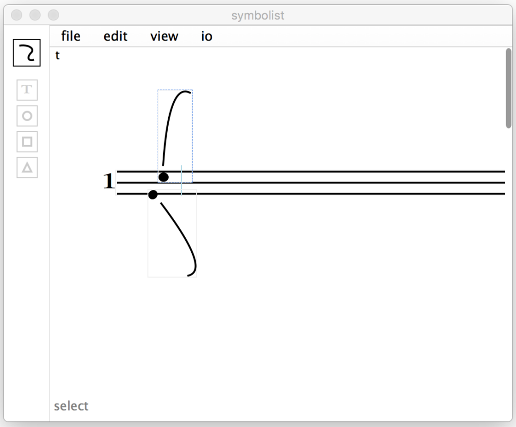
Symbols can also have internal timing and time referential — for example imagine a curve, or another graphic symbol which could represent the evolution of one or several parameters over a given amount of time. In *path*-symbols the relative time position is used to lookup the $\{x, y\}$ location on the path, output at the address */lookup/xy*.

For example, in Figure 8, a "frame notation" is used to control the spatialization of a sequence of events. In this case, a compound symbol is used, consisting of: (1) a 2D spatial region defined by the rectangle frame, (2) a *path* depicting a trajectory moving through the 2D space, and (3) a horizontal line which is used to define the duration of the symbol on the stave. The circle symbols below are sound events which are positioned using the frame notation above.

In order to optimize the processing of the time requests (which can occur at a relatively high rate in playback or score execution contexts), an internal "time-point array" is constructed and maintained along with score editing operations, which stores a sorted reference map of the score symbols' start and end points.

5. HOST ENVIRONMENTS

SYMBOLIST currently exists as a standalone application, and as a static or dynamic library. The main entry points of the application programming interface (API) are read/write




```
{
    /time/lookup : 0.7,
    /time/end : 0.89,
    /staff/foo/voice/1/glissnote/state : 0,
    /staff/foo/voice/1/glissnote/time/ratio : 0.631579,
    /staff/foo/voice/1/glissnote/name : "glissnote",
    /staff/foo/voice/1/glissnote/type : "group",
    /staff/foo/voice/1/glissnote/id : "glissnote/1",
    /staff/foo/voice/1/glissnote/staff : "foo/palette",
    /staff/foo/voice/1/glissnote/x : 46.,
    /staff/foo/voice/1/glissnote/y : −87.,
    [...]
    /staff/foo/voice/1/glissnote/numsymbols : 2,
    /staff/foo/voice/1/glissnote/time/start : 0.46,
    /staff/foo/voice/1/glissnote/time/duration : 0.38,
    /staff/foo/voice/1/glissnote/subsymbol/1/notehead/name : "notehead",
    /staff/foo/voice/1/glissnote/subsymbol/1/notehead/type : "circle",
    /staff/foo/voice/1/glissnote/subsymbol/1/notehead/x : 0.,
    [...]
    /staff/foo/voice/1/glissnote/subsymbol/2/glissando/name : "glissando",
    /staff/foo/voice/1/glissnote/subsymbol/2/glissando/type : "path",
    /staff/foo/voice/1/glissnote/subsymbol/2/glissando/x : 4.,
    [...]
    /staff/foo/voice/1/glissnote/subsymbol/2/glissando/lookup/xy : [0.268815, 0.234897],
    /staff/foo/voice/0/glissnote/state : 0,
    /staff/foo/voice/0/glissnote/time/ratio : 0.648148,
    /staff/foo/voice/0/glissnote/name : "glissnote",
    /staff/foo/voice/0/glissnote/type : "group",
    /staff/foo/voice/0/glissnote/id : "glissnote/2",
    /staff/foo/voice/0/glissnote/staff : "foo/palette",
    [...]
    /staff/foo/voice/0/glissnote/numsymbols : 2,
    /staff/foo/voice/0/glissnote/time/start : 0.35,
    /staff/foo/voice/0/glissnote/time/duration : 0.54,
    /staff/foo/voice/0/glissnote/subsymbol/1/notehead/name : "notehead",
    /staff/foo/voice/0/glissnote/subsymbol/1/notehead/type : "circle",
    [...]
    /staff/foo/voice/0/glissnote/subsymbol/2/glissando/name : "glissando",
    /staff/foo/voice/0/glissnote/subsymbol/2/glissando/type : "path",
    [...]
    /staff/foo/voice/0/glissnote/subsymbol/2/glissando/lookup/xy : [0.657142, 0.553749]
}
```


Figure 7. An example SYMBOLIST OSC output stream for a time point containing multiple timed symbols.

accessors which allow to build, store, process the score symbols in host environments, and perform time point lookup as described above in Section 4. All the data is transferred back and forth through OSC-encoded bundles. Two main host environment are currently supported.

Max. SYMBOLIST was embedded in an object for the Max environment [9], where the score editor can be used to store, generate and monitor timed streams of data (see Figure 8). Score readers can be easily implemented to browse through the score via time requests which output the corresponding symbols and associated data.

OpenMusic. SYMBOLIST was also integrated in the O7 prototype implementation of the OpenMusic computer-aided composition environment [11, 14]. OpenMusic programs can generate scores (sequences of OSC bundles representing staves and timed symbols), which can be connected to interactive, personalized graphical display and editing (see Figure 9). SYMBOLIST in this context offers alternative graphical representations for musical data reaching far beyond the expressive potential of traditional music notation editors or more neutral automation controllers. The editor here also can be easily connected to OpenMusic's embedded scheduling engines through the timed-request function of the SYMBOLIST API, which allows the score be "played", just as any other musical object of the environment, via timed transfer of OSC data.

6. DISCUSSION: TOWARDS EMBEDDED SCORES

An important challenge to be considered for the SYMBOLIST framework is how, if possible, to integrate these new notation tools with the current predominant practices in media art programming. Interactive computer-music and multimedia artists often make use of programming environments such as Max, Pure Data, SuperCollider, Processing, Arduino, Grasshopper, Blender, VVVV, OpenFrameworks, et al., where the compositional thought is directly integrated into the program that renders or performs the work. In these cases, the artist composes the piece directly in the code itself, embedding the artistic intention into the computational process which produces the piece [15].

For example in the "circuit scores" of David Tudor, the "composition" takes the form of an instrument: the instrument's behavior is composed as the result of an interaction between electronic components [16]. Chadabe, di Scipio, Leman, Wessel, and others have discussed this embedded nature of artistic intentions in interactive instrument systems, and its relation to cybernetics, systems theory, and embodied cognition studies [17, 18, 19, 20, 21, 2]. This merging of "instrument" and "composition" can also be observed in the "process scores" of Cage, Feldman, Stockhausen, et al., and all the way back to the *Musikalisches Würfelspiel* pieces by Mozart and Kirnberger, where the score describes a sequence of musical-cognitive processes which led to the production of the piece, rather than describing the results themselves.

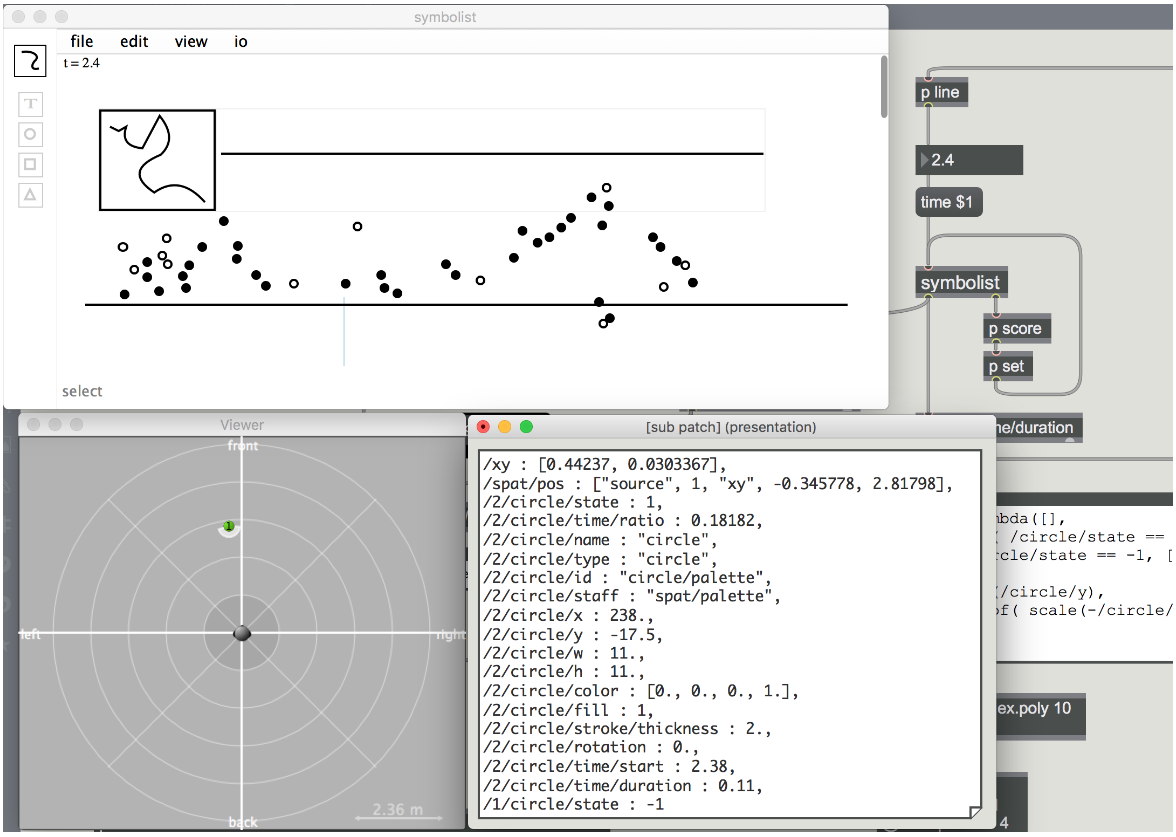


Figure 8. An example using SYMBOLIST in Max. Sending a time value into the SYMBOLIST Max object causes output of an OSC bundle containing the symbol values at that time-point. Spatial location is composed with a frame notation symbol group where the frame represents a given region in space, and the path is the trajectory distributed over the time of the horizontal line. Separately, circular symbols are used to notate sound events.

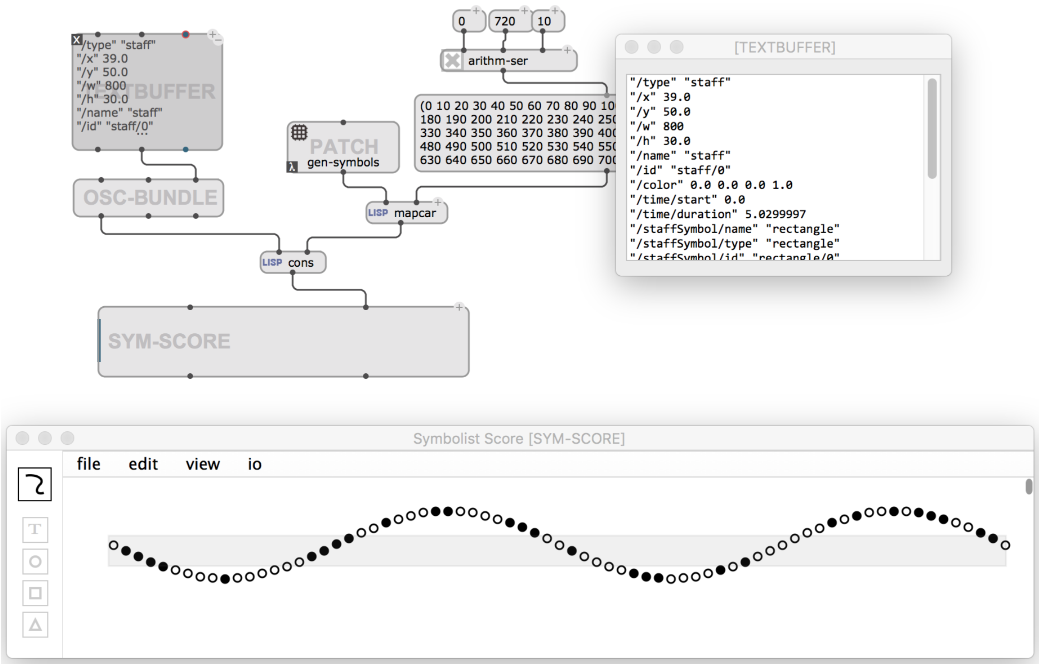


Figure 9. SYMBOLIST integration in OpenMusic (O7). OSC bundles describing symbols are written and generated algorithmically in the computer-aided composition environment (here to produce a sine-shaped sequence of small circle symbols), then displayed and edited in the SYMBOLIST editor.

In all of the above historical examples, the scores were notated to be read and performed by *humans*, which naturally requires them to be readily understood and interpreted by humans in terms of their learned and embodied cultural knowledge. This is no longer necessarily the situation when working with digital performance systems. Control parameters for digital processes need to be in a *computation*-friendly format which can be parsed and interpreted by the *program*, which invisibly transforms the algorithm and score into executable machine-code.

In this context where compositional processes are embedded into an interactive system, there is rarely a "score" separate from the instrument itself. This may well be the most natural approach for this situation, where the code, the instrument, and the score are all intertwined. However, it may be limiting as well, since as we discussed in the introduction, the affordances of a system have a strong influence on the uses of the system. A number of recent projects, such as INScore [22], *bach* [23], or PWGL's Expressive Notation Package (ENP) [24], are similarly navigating this hybrid zone between score and programmatic media generation. The question then for SYMBOLIST is, in what ways could notation function within the context of the embedded score?

A potential route of development could be to include interpretive expressions inside a symbol's OSC bundle which could be evaluated at performance time. Computational expressions could be composed in SYMBOLIST, either symbolically or as text, which could then be transcoded into another environment. The *odot* expression language would be a natural choice since it is specifically designed to operate on OSC messages and is well suited to transcoding between applications [25]. In the simplest case, a symbol could include anonymous functions which when evaluated would map the symbol data to the target rendering system format (spatial audio system, video, motors, etc.).

Attaching expressions to symbols could also be a way for users to create their own custom designed interaction tools. In this case, the expression could be evaluated while editing within SYMBOLIST, to provide additional information relevant to the intended output context (e.g. contextual displays), or used to create interactive drawing tools which could generate other types of symbolic/graphic information.

7. CONCLUSION AND PERSPECTIVES

We presented the first prototype of SYMBOLIST, a software developed for visualizing, editing, and executing control data streams for music and media encoded as OSC bundles. The project was conceived in response to the lack of efficient tools currently available to perform these tasks, and to expand the possibilities for multimedia and electroacoustic scores, which, when they exist, are most often incomplete, non-executable and/or non-editable: there is generally little support to *symbolically notate* computerized music and media control material.

SYMBOLIST aims at completing contemporary artists' and composers' toolboxes with a simple tool used to realize and execute such multimedia scores, and joins a burgeoning landscape of computer platforms for computer aided composition and multimedia notation [22, 23, 26, 27, 28]. As compared to IanniX's 3D timeline orientation [28], or to advanced sequencing tools such as i-Score [29] or Antescofo's Ascograph editor [30], which provide advanced means to program and visualize timing and interactions, SYMBOLIST emphasizes symbolic, graphical drawing/editing for new music and media notation.

The OSC foundation for the SYMBOLIST score data structure is not an arbitrary choice: it is today an established and widely supported format used for media data encoding and interchange, and we believe in the potential for its future development — especially through CNMAT's *odot* library — to greatly improve the expressivity of our software functionality. The planned future work in this project will feature the integration of and embedded OSC server, in order to fully support interaction with external software, as well as advanced embedded expression programming in OSC-encoded symbols, as discussed in Section 6.

Other future work directions are to continue development on the graphical display and rendering of scores, through a number of features related to page formatting and layout, printing, export to graphical formats, etc. Finally, in order to constitute a fully-workable score environment, the software will need to embed the possibility to integrate, edit and merge common music notation with the user-defined staves and symbols of the SYMBOLIST scores.

Acknowledgments

This work was realized in the context of a joint artistic research residency at IRCAM and ZKM.

8. REFERENCES

[1] J. Greeno, "Gibon's Affordances," *Psychological Review*, vol. 101, no. 2, pp. 336–342, 1994.

[2] T. Magnusson, "Of Epistemic Tools: Musical instruments as cognitive extensions," *Organised Sound*, vol. 14, no. 2, pp. 168–176, 2009.

[3] M. Battier, "Describe, Transcribe, Notate: Prospects and problems facing electroacoustic music," *Organised Sound*, vol. 20, no. 1, pp. 60–67, 2015.

[4] K. Stone, "Problems and Methods of Notation," *Perspectives of New Music*, vol. 1, no. 2, pp. 9–31, 1963.

[5] E. R. Tufte, *The Visual Display of Quantitative Information*. Graphics Press, 1983.

[6] R. Gottfried, "Studies on the Compositional Use of Space," IRCAM, Paris, France, Tech. Rep., 2013.

[7] T. Carpentier, N. Barrett, R. Gottfried, and M. Noisternig, "Holophonic Sound in IRCAM's Concert Hall: Technological and Aesthetic Practices," *Computer Music Journal*, vol. 40, no. 4, pp. 14–34, 2017.

[8] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193–200, 2005.

[9] M. Puckette, "Combining Event and Signal Processing in the MAX Graphical Programming Environment," *Computer Music Journal*, vol. 15, no. 3, pp. 68–77, 1991.

[10] R. Gottfried, "SVG to OSC Transcoding: Towards a Platform for Notational Praxis and Electronic Performance," in *Proceedings of the International Conference on Technologies for Notation and Representation (TENOR'15)*, Paris, France, 2015.

[11] J. Bresson, D. Bouche, T. Carpentier, D. Schwarz, and J. Garcia, "Next-generation Computer-aided Composition Environment: A New Implementation of OpenMusic," in *Proceedings of the International Computer Music Conference (ICMC'17)*, Shanghai, China, 2017.

[12] World Wide Web Consortium, "Scalable Vector Graphics (SVG) 1.1 (Second Edition)," *W3C Candidate Recommendation*, 2011. [Online]. Available: https://www.w3.org/TR/SVG11/

[13] A. H. Guest, *Labanotation: The System of aAalyzing and Recording Movement*. Routledge, 2005.

[14] J. Bresson, C. Agon, and G. Assayag, "OpenMusic: Visual Programming Environment for Music Composition, Analysis and Research," in *Proceedings of the ACM international conference on Multimedia – Open-Source Software Competition*, Scottsdale, AZ, USA, 2011, pp. 743–746.

[15] T. Magnusson, "Algorithms as Scores: Coding Live Music," *Leonardo Music Journal*, vol. 21, pp. 19–23, 2011.

[16] R. Kuivila, "Open Sources: Words, Circuits and the Notation-Realization Relation in the Music of David Tudor," *Leonardo Music Journal*, vol. 14, pp. 17–23, 2004.

[17] J. Chadabe, "Interactive Composing: An Overview," *Computer Music Journal*, vol. 8, no. 1, pp. 22–27, 1984.

[18] D. Wessel, "An enactive approach to computer music performance," in *Le Feedback – Acte des Rencontres Musicales Pluridisciplinaires*. Lyon, France: Grame, 2006, pp. 93–98.

[19] M. Leman, *Embodied Music Cognition and Mediation Technology*. MIT Press, 2008.

[20] A. Di Scipio, "'Sound is the interface': from interactive to ecosystemic signal processing," *Organised Sound*, vol. 8, no. 3, pp. 269–277, 2003.

[21] N. Schnell and M. Battier, "Introducing Composed Instruments, Technical and Musicological Implications," in *Proceedings of the conference on New Interfaces for Musical Expression (NIME'02)*. Dublin, Ireland, 2002.

[22] D. Fober, S. Letz, Y. Orlarey, and F. Bevilacqua, "Programming interactive music scores with INScore," in *Proceedings of the Sound and Music Computing conference (SMC'13)*, Stockholm, Sweden, 2013, pp. 185–190.

[23] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015.

[24] M. Kuuskankare, "ENP: A System for Contemporary Music Notation," *Contemporary Music Review*, vol. 28, no. 2, pp. 221–235, 2009.

[25] J. MacCallum, R. Gottfried, I. Rostovtsev, J. Bresson, and A. Freed, "Dynamic Message-Oriented Middleware with Open Sound Control and Odot," in *Proceedings of the International Computer Music Conference (ICMC'15)*, Denton, TX, USA, 2015.

[26] N. Didkovsky and G. Hajdu, "MaxScore: Music Notation in Max/MSP," in *Proceedings of the International Computer Music Conference (ICMC'08)*, Belfast, Northern Ireland / UK, 2008.

[27] C. Hope, L. Vickery, A. Wyatt, and S. James, "The Decibel Scoreplayer – A digital tool for reading graphic notation," in *International conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015.

[28] T. Coduys and G. Ferry, "Iannix-aesthetical/symbolic visualisations for hypermedia composition," in *Proceedings of the Sound and Music Computing conference (SMC'04)*, Paris, France, 2004, pp. 18–23.

[29] J.-M. Celerier, M. Desainte-Catherine, and J.-M. Couturier, "Graphical Temporal Structured Programming for Interactive Music," in *Proceedinsg of teh International Computer Music Conference (ICMC'16)*, Utrecht, Netherlands, 2016.

[30] G. Burloiu, A. Cont, and C. Poncelet, "A visual framework for dynamic mixed music notation," *Journal of New Music Research*, vol. 46, no. 1, pp. 54–73, 2017.

PITCHES IN BACH

Andrea Agostini
Conservatory of Turin
andreaagostini@bachproject.net

Daniele Ghisi
STMS Lab (IRCAM, CNRS, UPMC)
Conservatory of Genoa
danieleghisi@bachproject.net

ABSTRACT

Traditionally, most computer-aided composition environments represent a pitch via a number (typically a MIDI note number or its value in midicents), flattening the enharmonic information onto a single real-valued parameter. Although this choice is convenient in many applications, it can be very limiting in any context where diatonicism, to some degree, matters.

The latest release of *bach*, a library for Max dedicated to musical representation and computer-aided composition, introduces a new 'pitch' data type, designed to overcome this limitation by representing both diatonic pitches and intervals and supporting standard arithmetic operations. In this article we motivate and detail its implementation and its syntax.

As an application, we introduce a new respelling algorithm, also implemented in *bach*, designed to provide an easy-to-read spelling of notes. Differently from most existing pitch spelling algorithms, tailored on the tonal repertoire, our algorithm is targeted to produce a musician-friendly representation of non-tonal music.

1. INTRODUCTION

1.1 The problem

Virtually every software system capable of dealing with symbolic musical information has some kind of representation of pitch. Some tools for computer-aided composition, including OpenMusic [1] and PWGL [2], as well as versions of *bach* [3] prior to 0.8, employ MIDI note numbers or midicents, thus not providing a direct way to express enharmonic information: of course, even in these cases it is always possible to set up custom representations, but manipulating them would require the effort of constructing all the necessary tools. On the other hand, other software systems, such as Abjad [4] and Music21 [5], embed enharmonic information in their basic representation of pitches.

[1] http://repmus.ircam.fr/openmusic/home
[2] http://www2.siba.fi/PWGL/
[3] www.bachproject.net
[4] http://abjad.mbrsi.org
[5] http://web.mit.edu/music21/

Both choices have their own advantages and disadvantages. Reducing pitch to a basic numeric type by eschewing enharmonic information simplifies the system: at the very least, it avoids the need for specific constructors and methods. In some regards, it can also make life easier for users, who do not need to become acquainted with a specific syntax and set of operations.

On the other hand, it is a very limiting choice, for a number of reasons. For one thing, the tuning of pitches depends on the chosen temperament—yet, we shall consider, in this article, only the case of equal temperaments. Even in this case, the choice of dropping enharmonic information is still inadequate, from at least two points of view: a technical one, because there may be good reasons (such as readability) for preferring one kind of enharmonic spelling to another; and a more strictly musical one, because such a representation is strongly connected to a non-hierarchical conception of musical pitches and the networks of significance they form within the musical discourse. After all, in a typical piece of music by Pierre Boulez, Franco Donatoni or even Anton von Webern, the choice of representing a given musical pitch as an F♯ rather than a G♭ is mostly irrelevant, to the point that several composers, Donatoni included, have made very limited use of accidentals other than the sharp. It is not by chance that the three aforementioned composers have a strong relationship with dodecaphony and serialism. On the contrary, a page by Bach or Mozart would be substantially wrong if typeset with all the F♯'s and G♭'s swapped. Moreover, although in most cases this information can be reconstructed, there are instances in which the enharmonic spelling chosen by the composer carries meaning useful to shed light on how a particular chord or passage should be interpreted [1]. A notable example is Richard Wagner's famous Tristan chord, which has been the subject of debate since more than a century: the analytical tools involved are meaningful only if they take enharmonic spelling into account, and the insight they provide is highly relevant to the understanding of late-19th century and early-20th century tonal music.

Several works and sub-genres of contemporary music fall somewhere between these two categories. Whereas music strictly adhering to the tonal system, as found in works by composers from the 18th and early 19th centuries, is now almost solely composed in the context of school exercises, the same cannot be said for music closer, or belonging, to the harmonic traditions of jazz, rock and pop [2]. On the other hand, the tonal syntax of concert music from the 19th and early 20th centuries still forms the harmonic basis for a wide array of contemporary, non-strictly-concert music,

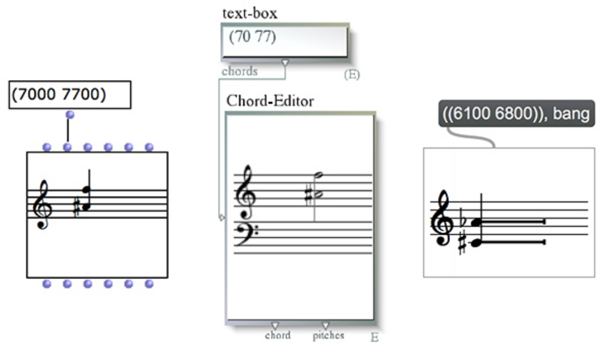


Figure 1. Display of MIDI notes corresponding to perfect fifths in some of the most common computer-aided composition environments (from left to right, OpenMusic, PWGL and *bach*). Each environment has somehow its own 'wolf fifth'.

most notably—but not exclusively—film music. Moreover, although self-described 'art music', roughly over the last century, has distanced itself from the received, historically connoted syntax of tonal language, by no means it has consistently renounced all forms of hierarchical syntax of pitches. This observation refers, in the first place, to various branches of so-called 'neomodal' music, a category that may be applied to works by composers as diverse as Terry Riley, Arvo Pärt and Louis Andriessen, or—more recently—Yannis Kyriakydes, Andrew Hamilton and Nico Mulhy. On the other hand, other sub-genres and individual works in the field of contemporary art music may be described as featuring hierarchical (albeit not modal) pitch structures, including works by composers influenced to various degrees by spectralism, such as Gérard Grisey and Kaija Saariaho, or works explicitly referencing other musical idioms, be they popular, folkloric or historical, such as *Sinfonia* by Luciano Berio, *Professor Bad Trip* by Fausto Romitelli or *Cognitive Consonance* by Christopher Trapani. In all these contexts, the question "Is this an F♯ or a G♭?" is not an idle one, because it alludes to the functional roles that pitches carry within the musical discourse. And anyway, even in more strictly serial or post-serial contexts, there is some sort of consensus on the 'correct' representation of intervals: for instance, it is uncommon to come across diminished sixths where perfect fifths could be used—something composers working with computer-aided composition tools have unfortunately been trained to tolerate (see Figure 1). Effective software tools for musical formalization should take all this into account. Therefore, our aim is to provide a formalization and an arithmetic of pitches in equal temperaments, and implement it in *bach*.

1.2 A proposed solution for Max and *bach*

Max has a very limited focus on symbolic musical representation, and objects that need to represent pitch do it according to the MIDI standard. The *bach* package for Max, conceived specifically to augment Max with advanced capabilities of representation and treatment of musical data [3] has been using midicents as its native way of representing pitches, too, coherently with its main original references (namely, OpenMusic and PWGL). This was also a convenient choice for easing the communication between *bach* objects and native Max objects, as the only conversion tool required was a division or a multiplication by 100, respectively for converting midicents into MIDI pitches, or viceversa. In the latest major version of *bach* (0.8), on the other hand, we felt that this simplistic representation was not adequate to the scope we envisioned. For this reason, we decided to implement in the *bach* system a new data type, aptly called a *pitch*, representing musical pitches and meant to be operated upon through both standard mathematical operators and new, specific tools.

2. REPRESENTATION OF PITCHES

The mathematics of pitch representation is a well-studied field, especially in the context of equal temperament. Although most techniques, influenced by the musical set theory, tend to flatten pitches onto their MIDI note numbers (to the point that nowadays the term 'pitch-class' commonly refers to MIDI note classes rather than diatonic pitch classes), there exist at least two families of approaches that preserve enharmonic information. Models in the the first family represent pitches as belonging to geometrical structures in space (such as the line of fifths, the Tonnetz [4], or the spiral array [6] [6]). Models in the the second family essentially represent pitches as couples $(c, d) \in \mathbf{Q} \times \mathbf{Z}$, where c is the number of chromatic steps or semitones from a reference note, such as middle C, and d is the number of diatonic steps from the same reference note [7, 8]. As an example, the F♯ just above middle C would be represented as $(6, 3)$, while its enharmonic equivalent, G♭, would be $(6, 4)$. Several variants of this representation exist (e.g. using midicents instead of semitones, or choosing C0 as reference note); we will refer to similar encodings as 'chromatic-diatonic couples'.

Both these families of representations have the advantage to make standard operations such as transposition or enharmonic respelling arithmetically trivial—at the expense of making other properties less readable. For example, it is not straightforward to infer the accidental of a pitch from either a spatial position inside a geometrical structure or a chromatic-diatonic couple.

The *bach* library takes advantage of both of these representations (the first one is used, for instance, in pitch respelling algorithms, whereas the second one is used to facilitate some arithmetic operations). However, we have decided to use internally a container whose fields mirror more directly the way we usually think of notes, that is, a degree, an alteration and an octave.

Several models for tridimensional representations of pitches have been proposed. Most of them involve quotienting by an operation of octave transposition, hence disentangling the octave number from a two-dimensional representation of a diatonic pitch-class.

Brinkman's 'binomial representation' [9], represents such diatonic pitch-classes as a combination of a 'MIDI pitch-

[6] The spiral array should not be confused with Shepard's helix [5], which does not distinguish enharmonic pitches.

class' (0 to 11) and 'letter-class' (0 to 6). Brinkman's representation is however not equivalent to chromatic-diatonic couples; namely, as the author recognizes, it has the inelegant disadvantage of allowing ambiguities when more than five accidentals are involved: a MIDI pitch-class of 6 together with a letter-class of 0 may correspond both to C sextuple sharp and to C sextuple flat.

Clement lays in [10] important groundwork concerning the relationship between pitches and intervals, namely asserting that all intervals, and hence all pitches, can be generated via combinations of a chromatic half-step and a diatonic half-step. However, Clement chooses to eventually flatten the pitch parameter onto a single integer, managing to distinguish quite well the most common enharmonic representations, yet still leaving room for ambiguities when larger alterations are involved. Also, Clement uses different names and grammars for pitches and intervals — a distinction that trained musicians usually take for granted but which, in our own view, is unnecessary (as the next section will detail).

Drawing from all these researches and considerations, we have decided to implement our own encoding of pitches in *bach*, as we explain in section 4.

3. ARITHMETIC

3.1 Pitches and intervals

When we say something seemingly trivial like "this is an E♭ at octave 4", we are superposing two kinds of reasoning: on the one hand, the general concept of 'E♭' is a shared, albeit slippery, one, and there is at least partial consensus about what 'octave 4' means. [7] On the other hand, without a reference pitch and tuning system, it is in principle impossible to assign an exact frequency (that is, an exact meaning with respect to sound) to 'E♭ at octave 4'. In this sense, we can say that the name of any musical pitch represents, strictly speaking, an interval with respect to a fixed reference within a certain tuning system. So, according to one of the most widespread practices, 'E♭ at octave 4' means "a tempered augmented fourth below the A4, the frequency of the latter being 440 Hz". In a context of purely symbolic computation, the relation to the exact frequency of a reference pitch may be irrelevant, but the substantial identification of absolute pitches and intervals is an elegant conceptual tool for simplifying the expression of transpositions and other operations.

[7] There are many possible specific definitions and interpretations of 'E♭', both formal (for example, the set of all the notes that can be obtained by stacking three descending fifths starting from a C) and informal (for example, a referral to the embodied cognition of the production of a generic E♭ on a musical instrument, sometimes coinciding with its enharmonic D♯), but most of them share enough common traits to allow both musicians and non-musician to talk practically about E♭'s without worrying about substantial misunderstandings. Octave numbering is usually a more technical matter, and in fact there are several conventions for distinguishing between different E♭'s in the audible range (and, potentially, beyond it). The arabic numeral after the note name is especially used in electronic instruments and music software. The most widespread convention appears to be the one setting C4 as the middle C, written with one ledger line below a treble clef staff and typically corresponding to a frequency of roughly 261.5 Hz (the case with transposing instruments requiring further specification). As we shall discuss below, we chose to adopt a different convention in this regard.

Another way to see this possibly confusing identification is that, on the one hand, we see the musical interval as an essentially spatial measure, and, as such, we typically use it in a relative way (we cannot say that Montreal is located *at 3000 km*, but rather that it is *3000 km away from Albuquerque*). On the other hand, we are somehow used to treat the nomenclature of pitches as just a set of names, not unlike what we do with colors, albeit a very formally defined one. What we are proposing here is that, considering the unambiguousness of pitch names and the trivial and biunivocal relation between absolute pitches and the interval of each pitch from C0, we can actually merge the two concepts and use a single naming scheme for both. This is not too different from what we do when we use Celsius degrees for both measuring the temperature difference between two bodies and expressing absolute temperatures as the distance between a body's temperature and the arbitrarily chosen reference of the water's melting point.

These considerations have informed two fundamental choices at the basis of the pitch representation system in *bach*: first, as hinted above, the same format and data type used for expressing pitches is also used for intervals with respect to a reference pitch of C0. Thus, E♭0 denotes both a very low E flat and an ascending minor third, whereas -F0 (and the equivalent form G-1) denotes a descending perfect fourth. A possibly more rigorous way to see this is considering the system from the point of view of intervals: E♭0 has "minor third" as its first meaning, and we can use it to denote an absolute pitch located a minor third above the C0 reference. This also explains the -F0 = G-1 identity: -F0, considered as an interval, denotes a downward perfect fourth; and a perfect fourth below the C0 reference is G-1. As a side note, *bach* accepts the two representations indifferently, but (since very low values are more often used to express intervals than absolute pitches) returns the 'interval' format, the one with an optional leading minus sign but only non-negative octaves, as the preferred format for textual representation; two objects, bach.write and bach.textout, provide options for returning the other format, potentially with negative octaves. The fact that C0 is the reference for absolute pitches as well as for intervals leads to the second consideration: because we want our system to retain backwards compatibility with *bach*'s previous, midicent-based system of representation of pitches, we now need transposition of pitches to behave consistently with transposition of midicents. This means that transposing a pitch by a minor third must be compatible with summing 300 midicents, which implies that E♭0 must be 300 midicents, C0 (the perfect unison, and the identity element for transposition) be 0 midicents, and C5 be 6000 midicents (that is, middle C). This is a different standard from the two most widely used (placing middle C at the beginning of octave 3 or 4), but there is at least one precedent in Cakewalk Sonar, and there used to be an additional one in older versions of Reaper.

The simplicity and elegance of this architecture have been the two important factors leading to our choice of C5 as middle C in *bach*. On the other hand, it is always possible to express pitch literals according to a different standard,

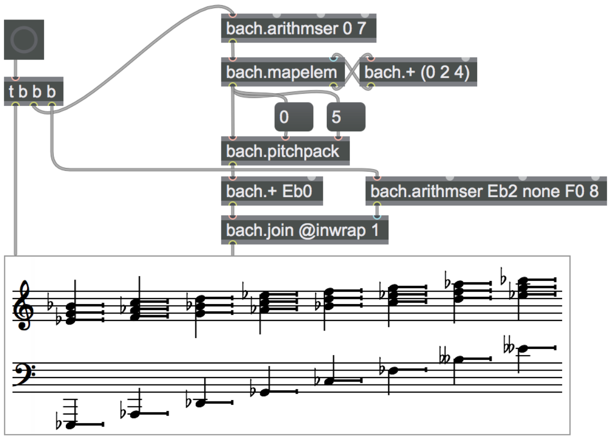


Figure 2. Pitch arithmetic addresses a whole area of diatonic, modal and tonal musical processes.

and applying to them a transposition of one or two octaves. We will hence assume throughout the rest of the article that C0 is MIDI note 0 (and hence C5 is middle C).

3.2 Operations

Algebraic sums and multiplications are meaningful on intervals: for example, a minor third plus a major third is a perfect fifth, and a perfect fifth minus a minor third (that is, plus a descending minor third) is a major third. Within our convention, we may write E♭0 + E0 = G0. This amounts to transposing any of the two pitches by the interval represented by the other one; for instance, summing any pitch to E♭0 will result in a transposition by a minor third (see Figure 2). C0 (unison) is the identity element for the sum.

We have not been able to assign a musical meaning to the multiplication of two intervals in the pitch domain. [8] However, there is a natural external multiplication of an interval by an integer, which can simply be seen as a sequence of sums, with a sign depending on the signs of the factors. As an example, $12 \cdot$ G0 = B♯6. A multiplication by -1 inverts the interval; as previously stated, pitches lower than C0 can be expressed either with a negative octave or with a negative interval (for example, $-1 \cdot$ E♭0 = −E♭0 = A-1).

All the above operations are unambiguous with respect to enharmonicity. Partitive division of an interval by an integer, on the other hand, is generally problematic: what does it mean to divide an augmented fourth by two? The difficulty here arises from the fact that we wish our operations to be meaningful with respect to enharmonic spelling. So, although an augmented fourth is 6 semitones wide, and as such dividing it by two would result in 3 semitones, there are theoretically infinite intervals spanning 3 semitones (minor third, augmented second, doubly diminished fourth, etc.), but none of those, if multiplied by two, will return an augmented fourth. For example, a minor third times two is a diminished fifth, and an augmented second times two is a doubly augmented third. On the other hand, by performing an integer division on the 0-based degree and the integer, and subsequently adjusting correctly the accidental and/or alteration, it is possible to obtain a pitch quotient spanning the correct amount of semitones, or fraction thereof. This pitch quotient, if multiplied back by the original divisor, is an interval possibly different from the original dividend, but enharmonic to it. The difference between the divisor and the product of the dividend and the quotient is the remainder of the division, and it always spans 0 semitones—that is, it is always enharmonic to a perfect unison. So, an augmented fourth divided by two is an augmented second, with a remainder of a diminished second (because an augmented second times two is a doubly augmented third, and an augmented fourth minus a doubly augmented third is just a diminished second). In our pitch syntax: F♯0/2 = D♯0 with the remainder of D♭♭0, because $2 \cdot$ D♯0 + D♭♭0 = F♯0.

Quotative division of an interval by an interval is also possible. It involves promotion of the two terms to midicents, and returns an integer or a rational number. If the second term is C0, the division is indeterminate. The remainder of the quotative division is simply defined as the difference between the divisor and the product of the dividend and the quotient: since the dividend is a pitch and the quotient is an integer, their product is also a pitch and the aforementioned difference is also a pitch. For instance, A1/G0 = 3 with no remainder, while E♯2/G0 = 4 with a remainder of C♯0.

3.3 Comparisons

Comparisons among pitches can also be expressed: given two pitches A and B, we say that $A = B$ iff their degrees, alterations and octaves are the same. Thus, C♯5 is different from D♭5, even if their midicents are the same. In this sense, and differently from what happens (not considering the limitations of numerical representation) when promoting an integer to a float, promoting pitches to rationals may change the result of an equality comparison performed upon them. Moreover, the 'less than' comparison operates lexicographically: $A < B$ if the octave of A is less than the octave of B, or, in case they coincide, if the degree of A is less than the degree of B, or, in case they also coincide, if the alteration of A is less than the alteration of B. Again, an inequality comparison performed on pitches can lead to the opposite result of the same inequality performed upon the midicents of those pitches: for example, B♯4 < C♭5 and E♯5 < F♭5.

These choices have been the subject of careful consideration, and have not been taken lightly. The main reason to choose these seemingly incoherent behaviors as the default is to preserve the richness of the pitch semantics (using the midicents ordering as 'less than or equal to' criterion would imply that all enharmonic spellings are equal). After all, it is straightforward to implement the 'other' behavior (the one according to which B♯4 > C♭5 and E♯5 > F♭5) when needed: all it takes is forcing the conversion to midicents, something *bach* provides various simple options for. All this being said, we are well aware that the

[8] For the sake of clarity, it may be worth recalling that multiplying a frequency by an interval as defined in the frequency domain (that is, the ratio between two frequencies) is perfectly meaningful and corresponds to an equal temperament transposition in the frequency domain. This operation is completely distinct from the meaningless multiplication of two intervals in the pitch domain.

answer most musicians would give to the question "Which is higher, B♯4 or C♭5?" would probably be the opposite of what our system gives.

3.4 Chromatic-diatonic representation

All the aforementioned choices are expressed more concisely using the chromatic-diatonic representation of pitches. Let $A = (c_A, d_A)$ and $B = (c_B, d_B)$ be two pitches such that c_i and d_i are respectively the number of semitones and the number of diatonic steps from the reference point C0 (with midicents 0). Then $A + B := (c_A + c_B, d_A + d_B)$ is the transposition operation, $-A := (-c_A, -d_A)$ is the inversion operation, $n \cdot A := (n \cdot c_A, n \cdot d_A)$ is the multiplication of a pitch by a number $n \in \mathbf{Z}$ (the set of pitches is thus a $\mathbf{Z}$-module). Partitive division is $A/n := (c_A/n, \lfloor d_A/n \rfloor)$ with remainder of $(0, d_A - n\lfloor d_A/n \rfloor)$, enharmonic to C0; quotitive division is $A/B := (\lfloor c_A/c_B \rfloor, \lfloor d_A/d_B \rfloor)$ with remainder of $(c_A - c_B\lfloor c_A/c_B \rfloor, d_A - d_B\lfloor d_A/d_B \rfloor)$.

The standard lexicographical order is defined on pitches: $A \leq B \Leftrightarrow d_A < d_B \vee (d_A = d_B \wedge c_A \leq c_B)$. Any $(c_A, d_A + k), \forall k \in \mathbf{Z}$ is an enharmonic respelling of A.

4. THE BACH IMPLEMENTATION

A pitch in *bach* is a triplet (g, a, o), where $g \in \mathbf{Z}/7\mathbf{Z}$ is the degree, $a \in \mathbf{Q}$ is the alteration (in fraction of tone) and $o \in \mathbf{Z}$ is the octave. In the internal representation, the degree is a number from 0 to 6, representing white keys names from C to B; the alteration is a rational number [9] ; and the octave is an integer, with octave 5 starting with middle C (corresponding to the MIDI pitch 60), and subsequently octave 0 starting with MIDI pitch 0.

Conversions between this chromatic-diatonic representation (c, d) and *bach*'s encoding of pitches as triplets (g, a, o) of degree, alteration and octave are straightforward:

$$\begin{cases} c &= \text{deg2chr}(g) + 2a + 12o \\ d &= g + 7o \end{cases}$$

and

$$\begin{cases} g &= [d]_7 \\ o &= \lfloor d/7 \rfloor \\ a &= \dfrac{c - 12o - \text{deg2chr}([d]_7)}{2} \end{cases}$$

with deg2chr: $\mathbf{Z}/7\mathbf{Z} \to \mathbf{Z}$ mapping $[0]_7 \mapsto 0, [1]_7 \mapsto 2, [2]_7 \mapsto 4, [3]_7 \mapsto 5, [4]_7 \mapsto 7, [5]_7 \mapsto 9, [6]_7 \mapsto 11$.

A pitch, according to the above definition, is stored in a double word, according to the computer architecture in use. Under a 32-bit architecture, a pitch is stored in 8 bytes (64 bits): 2 bytes for the degree (which of course is overkill, since its value is limited to the 0-6 range), 2 bytes for the octave (hence limited to the enormous range -32768 to 32767), and 4 bytes for the alteration (2 bytes for the numerator and 2 for the denominator, allowing for an extremely precise representation). Under a 64-bit architecture, a pitch is stored in 16 bytes (128 bits), thus doubling the size of all its fields with respect to the above.

[9] Rational numbers and arithmetic operations upon them are introduced in Max by *bach*.

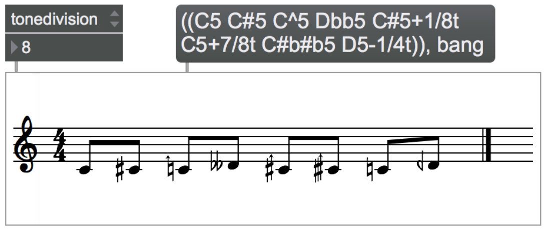


Figure 3. Some examples of pitch syntax in *bach*.

There is no explicit concept of a 'pitch constructor' in *bach*: the simplest way to construct a pitch is just typing its textual representation into a message object and passing it to a *bach* object. The textual syntax of a pitch is structured as follows (brackets delimit optional elements):

$$(\pm)\langle\text{degree}\rangle[\langle\text{accidental}\rangle]\langle\text{octave}\rangle[\pm\langle\text{alteration}\rangle\text{t}]$$

where the degree is a letter corresponding to an Anglo-saxon note name (from A to G); the accidental is a combination of the characters # (sharp), b (flat), x (double sharp), q (quartertone sharp), d (quartertone flat), ^ (eighth-tone sharp), v (eighth-tone flat), whose values are summed together; the octave is a positive or negative integer; and the alteration is a signed integer or rational number expressing a deviation in tones (or a fraction thereof) from the pitch as defined by the degree / accidental / octave representation. Both the accidental and the alteration are optional, but the degree and the octave must always be present (for instance C is not a pitch). The leading unary minus or plus is also optional: the plus sign has no effect, whereas the meaning of the unary minus flips the interval direction, as explained above. Examples of properly formatted pitches, as typed into a message box, are: C0, D#3, E-1, Fbbb6 (an F triple-flat at octave 6), Abv5 (an A flat minus an eighth tone at octave 5), B5-1/2t (a B minus a half tone, equivalent to a B♭5), C♯4+1/10t (a C sharp plus one tenth of a tone). Also see Figure 3 for an illustration.

The same representation is essentially used when a pitch is returned as text. As hinted at above, the same pitch can be represented through several representations: for example, B5-1/2t and Bb5 represent the same pitch, and the same goes for C#v3 and Cq^3. It is also possible to invent 'absurd' representations, such as C#b#b2 for C2, or Dvvvv4 for Db4. In principle, for each pitch there are infinite representations. Among those, each pitch has a 'normal form', that is, the representation with the shortest combination of same-direction accidental signs and the alteration with the smallest absolute value (or, if possible, no alteration at all: accidentals are preferred over alteration).

The musical notation editors of *bach* (namely, the bach.roll and bach.score objects) are now capable to accept pitches as their input. [10] Mathematical expression among pitches

[10] This is not completely new, as in previous versions of *bach* there was a way to assign a specific enharmonic spelling to a note, but it was a cumbersome one: besides entering the pitch in midicents, it was (and, for backwards compatibility sake, still is) possible to specify that the graphical representation of the note was composed by a given 'white key' pitch and a given alteration. There was even a sort of 'shortcut' for this, in that, by entering, say, 'Db4', the appropriate pitch and graphical information

can be evaluated via the usual *bach* arithmetic modules [11] : the *bach* evaluator can now perform operations on pitches, just like it does with regular numerical types, following the explanation provided in section 3.2. In order to handle results of indeterminate operations (such as divisions by C0), a special NaP ('not a pitch') value is returned. In addition to the set of functions explicitly supporting pitches, any mathematical operator and function can accept pitches, which are implicitly promoted to integer, rational or floating-point midicents and operated upon as such: for instance, calculating the square root of E8, corresponding to 10000 midicents, returns the floating point value 100., as the sqrt function only operates upon floats, and promotes to a float all the other number types.

5. PITCH SPELLING ALGORITHMS

Finding the best possible spelling for sequences of notes and chords is far from a trivial issue, requiring knowledge of the musical context as well as computation time—which is why essentially all computer-aided composition environments tolerate default awkward spellings such as the ones in Figure 1. A certain number of pitch spelling algorithms have been proposed in the last few decades [11], aiming at finding, to some respect, the 'best' spelling of notes, given their MIDI numbers, onsets and durations.

In the new *bach* release, both bach.roll and bach.score feature three pitch spelling algorithms, triggered via a 'respell' message:

- a trivial algorithm, providing automatic note-by-note respelling, without any context or memory. For each step in the semitonal (or microtonal) scale, a 'standard' enharmonic representation is used. Either such representation is provided by the user (via an enharmonic spelling table), or a hard-coded choice, depending on the current key signature, is used;
- the algorithm proposed by Chew and Chen [12] in [12], based on Chew's spiral representation of pitches [6];
- a new 'atonal' algorithm, described in section 5.1.

Each of these algorithms can operate either voice by voice (so as to provide consistent readability for single, specific voices) or globally (so as to provide diatonic consistence across different voices). They can also limit their scope to subsets of the line of fifths (see Figure 4), by defining a 'sharpest' and/or a 'flattest' representable pitch (Figure 6).

5.1 General outline of the atonal algorithm

Although it is true that pitch spelling is imperative in tonal music (as stated in the introduction, an F♯ might be substantially wrong inside a piece in G♭ major), it also plays a crucial role in the portion of non-tonal music where diatonicism has some importance. And yet, all the pitch spelling algorithms compared in [11] are essentially designed to work with tonal musical data, and they are hence only compared on historic tonal works. An important part of their workings deal with detecting harmonic modulations as precisely as possible.

The algorithm we propose is not tailored for this purpose—which is also why any comparison with the existing algorithms would be meaningless—but is rather meant to make general, non-tonal sequences of notes and chords 'as readable as possible' for musicians. In this context, detecting the precise position of a modulation is not a concern, whereas it is decisive to provide the players with a simple-to-read spelling for sequences of notes. We have developed our 'atonal' pitch spelling algorithm with these considerations in mind. As a side note, it should be remarked that the atonal algorithm can of course be applied to portions of modal and tonal music—which is why key signatures are also accounted for.

The idea at the basis of the atonal respelling algorithm is that notes that are close in time should be transcribed with pitches as close as possible on the line of fifths. [13] Therefore, the general outline of the algorithm is the following:

1. The notes belonging to the voice to be respelled (or to the entire score, depending on the chosen operation mode) are rearranged in a tree data structure, so as to reveal the proximity of notes in time. More specifically, the tree is structured so as to allow being traversed as follows: the couple of notes that are closest in time in the original voice or score (let us call them A and B) is encountered and evaluated first, thus forming a "core couple"; then the note closest to the previous pair is encountered, thus allowing it to be evaluated alongside A and B; and so on. In this way, increasingly large temporal windows of the original voice or score are taken into account. If, at any point, two notes not having been considered yet are closer to each other than either of them is closer to the current window, then the current window is put aside and the two new notes are considered as a new core couple, and the process moves forward from there. Further on, the new window may grow large enough to enclose the previous one, and in any case at the end of the process a single window containing the whole voice or score will be formed.
2. The rearranged tree is traversed according to the pattern described above, and over each step of the traversal the line-of-fifths distance of the pitches contained in the currently evaluated window is minimized, by searching for the combination of enharmonic representations with the smallest line-of-fifths standard deviation while respecting some ancillary constraints. If such standard deviation is within a given range, then the new enharmonic spelling is accepted, otherwise the algorithm settles upon the previously found

was automatically set. On the other hand, this kind of representation did not allow to perform arithmetic operations on pitches, and the extra information made the structure of the score more complex and less readable.

[11] In the actual implementation, integer division is performed towards zero and the remainder has the sign of the dividend, mirroring the behavior of the corresponding C functions.

[12] The algorithm was chosen based on [11], also considering the fact that Meredith's pitch spelling algorithms are subjected to patents.

[13] Following [12], a version with a spiral array representation had also been tested, to replace the line of fifths, with no significant improvement.

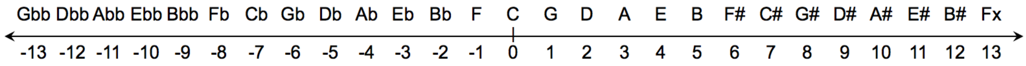


Figure 4. The line of fifths.

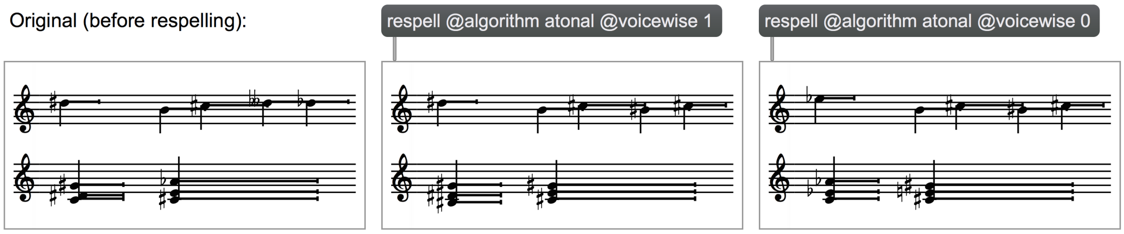


Figure 5. Voicewise versus non-voicewise respelling.

one and the traversal skips to the next core couple. The process goes on until there are core couples to be found.

A more detailed description of the algorithm, providing all the details needed for reimplementing it, and a practical example, are given below.

5.2 Detailed description of the atonal algorithm

A detailed description of the atonal respelling algorithm follows:

1. Respell the notes one by one via the aforementioned 'trivial' algorithm, providing a first rough spelling to be refined. This guarantees stability, since the rough spelling does not depend in any way from the original enharmonies, but only on the MIDI numbers. For instance, in a context with no key signature, spelling of portions of melodies in A♯ major, or in C♭♭ major, would be all equally respelled in B♭ major.

2. Build a list with all the notes of the voice, if the algorithm operates in a voice-wise fashion, or of the entire score otherwise. Chords are unpacked into notes with the same onset. At this stage, the list is flat; the next steps will reshape it into a tree, adding hierarchical levels (parens levels in bach *llll*s). Each node of the list contains some metadata, namely a 'starting time' s, an 'ending time' e (which at this stage both coincide with the onset of each specific note [14]) and a 'number of notes' n (at this stage $n = 1$).

3. Reshape the list constructed at point 2 into a tree, in the following way:

   3a. If the root level has a single node, then jump to step 4; otherwise find the closest nodes in the root level of the note list, i.e., find the two nodes such that the ending time of the first is closest to the starting time of the second. If there is a tie, take the first couple in temporal order.

   3b. Wrap the two nodes found in 3a in a new level (i.e., add a hierarchical node). If (s_L, e_L, n_L) and (s_R, e_R, n_R) are the metadata, respectively, of the earliest (left) and latest (right) node, then set the metadata of the new node to $(s_L, e_R, n_L + n_R)$.

   3c. Go to step 3a.

4. Perform the actual respelling. Obtain the list of nodes of the constructed tree via reversed breadth-first search and traverse it (deepest nodes are processed first). Process each node in the following way:

   4a. Let n be the number of notes of the node and $\mathcal{M} = (m_1, \ldots, m_n)$ be the list of MIDI numbers of the notes of the node. Also let $\mathcal{K} = (k_1, \ldots, k_n)$ be the key signatures of the voices to which the notes belong, and let $\mu_{\mathcal{K}}$ be the average of such signatures. Each $k_i \in \mathbf{Z}$ represents the number of sharps (if positive) or flats (if negative) of the key.
   If a node has a single note ($n = 1$), do nothing and jump to processing the next node. Otherwise continue to 4b.

   4b. Obtain the list of enharmonic possibilities for each $m_i \in \mathcal{M}$, in the form of an integer number (the position on the line of fifths, Figure 4) accounting for the 'sharpest' and 'flattest' parameters. Suppose that m_i has p_i enharmonic possibilities: let $\mathcal{C}_i = \{c_{i,1}, \ldots, c_{i,p_i}\}$ be the set of such numbers, $c_{i,j} \in \mathbf{Z}$. Let

$$\mathcal{C} = \bigcup_i \mathcal{C}_i = \{c_{1,1}, \ldots, c_{1,p_1}, c_{2,1}, \ldots, c_{2,p_2}, \ldots, c_{n,1}, \ldots, c_{n,p_n}\}$$

   be the collection of the enharmonic possibilities for each note.

   4b1. Consider each one of $c_{i,j} \in \mathcal{C}$ as a candidate 'center of effect' on the line of fifth, and respell each element of $\mathcal{M}$ so that its position on the line of fifths is as close as possible to $c_{i,j}$. Let

$$\mathcal{S}_{c_{i,j}} = (s_{c_{i,j},1}, \ldots, s_{c_{i,j},n})$$

   be the array of respelled positions, $s_{c_{i,j},k} \in \mathbf{Z}$.

[14] Notice that we call 'ending time' the largest note onset inside the hierarchical level, hence not accounting for note durations.

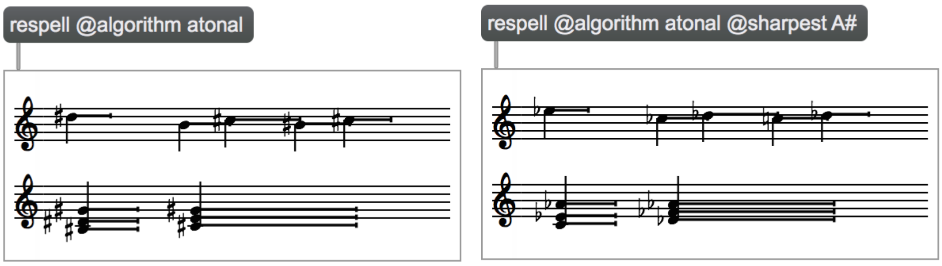


Figure 6. Defining 'sharpest' or 'flattest' notes has a global influence on the spelling algorithm.

4b2. Get the average $\mu_{c_{i,j}}$ and the standard deviation $\sigma_{c_{i,j}}$ of the $s_{c_{i,j},k}$'s. Normalize $\mu_{c_{i,j}}$ by subtracting the average of the key signatures $\mu_{\mathcal{K}}$ and add an additional bias, by default set to -2, accounting for the fact that first flat note appears at -2 on the line of fifths, while the first sharp note appears at 6 (flat and sharp notes are hence equally distant from the origin).

4b3. Determine whether the respelling $\mathcal{S}_{c_{i,j}}$ is acceptable. Only three conditions would make a respelling not acceptable:

- if any note is sharpest than the 'sharpest' acceptable or flattest than the 'flattest' acceptable;
- if altered repetitions (such as the sequence E♭-E♮) appear in $\mathcal{S}_{c_{i,j}}$ — but only in case a specific parameter to discard altered repetitions of the same pitch is set.
- if the standard deviation $\sigma_{c_{i,j}}$ is above a certain threshold $\widetilde{\sigma}$ (the threshold is a user-definable formula, defaulting to $\widetilde{\sigma} = \frac{21}{n+1}$). In other words, by default the threshold decreases as the number of notes of the set $\mathcal{M}$ increases.

If the $\mathcal{S}_{c_{i,j}}$ is not acceptable move to 4b5.

4b4. Determine if $\mathcal{S}_{c_{i,j}}$ is the 'best spelling' so far, i.e., the one having the smallest $\sigma_{c_{i,j}}$. In a tie, the spelling with the smallest $|\mu_{c_{i,j}}|$ is retained. If $\mathcal{S}_{c_{i,j}}$ is the best spelling, keep it as candidate.

4b5. Test the next possible candidate 'center of effect', i.e., go back to point 4b1 and move to testing $c_{i,j+1}$, or, if $j+1 > p_i$ then move to the element $c_{i+1,1}$; if $i+1 > n$, i.e., if all c_i's have been tried, move to 4c.

4c Once all $c_{i,j}$'s have been tested, there may or may not be a candidate for the respelling.

If there is no candidate, the node cannot be respelled, and all nodes containing it in the list of point 4 are dropped from the search.

If there is a candidate $\mathcal{S}_{c_{\bar{i},\bar{j}}}$, perform the respell of all notes according to it.

4d Jump back to point 4a and continue with the next node, until all nodes are completed.

This algorithm roughly provides a natural-to-read respelling of group of notes.

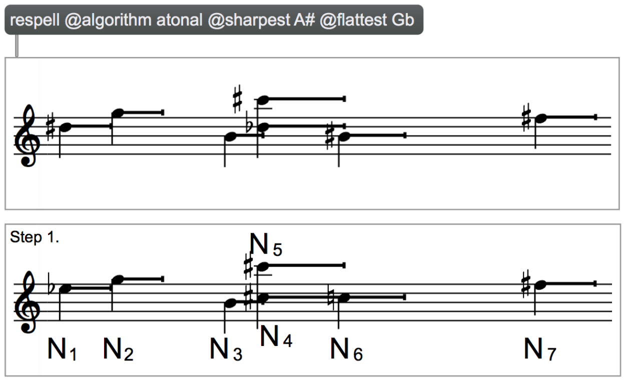


Figure 7. A simple example as a test for the algorithm.

5.3 An example case

To follow the behavior of the algorithm in a simple, concrete case, consider the score in Figure 7 and let $N_1, \dots, N_7$ be the notes to be respelled. As per step 1, we respell each note with standard enharmonic tables. Then, as per step 2, we obtain the list of individual notes N_i, and via step 3. we arrange it in tree form according to their distances. Since the two notes forming a chord are the nearest ones (according to their onsets), they will be the first to be wrapped in a level, yielding $N_1N_2N_3(N_4N_5)N_6N_7$. Then, the two nearest nodes are the note N_3 and the node (N_4N_5), hence we wrap them yielding $N_1N_2(N_3(N_4N_5))N_6$. We repeat the process, until we have a single node at the root level of the list, yielding the list displayed in Figure 8.

Once the tree is constructed, we apply step 4 and build the list of nodes to be visited, in reversed breadth-first search. This list is (due to 4a, we can drop the final nodes having a single note): (N_4N_5), $(N_3(N_4N_5))$, $((N_3(N_4N_5))N_6)$, (N_1N_2), $((N_1N_2)((N_3(N_4N_5))N_6))$, $(((N_1N_2)((N_3(N_4N_5))N_6))N_7)$.

We start with (N_4N_5). The set of possible positions on the line of fifths for each note is $\mathcal{C} = \{7, -5\}$, representing a C♯ and a D♭. No other options are possible, given our choice of 'sharpest' and 'flattest' pitches. Since N_4 and N_5 are the same note, $\sigma_7 = \sigma_{-5} = 0$, while $\mu_7 = 7 - 2 = 5$ and $\mu_{-5} = -5 - 2 = -7$, given a bias of 2. We accept $c_1 = 7$ as center of effect, and spell both notes as C♯.

We move to $(N_3(N_4N_5))$. The set of possible positions on the line of fifths is $\mathcal{C} = \{7, -5, 5\}$, corresponding to C♯, D♭ and B (no other enharmonic option is possible for

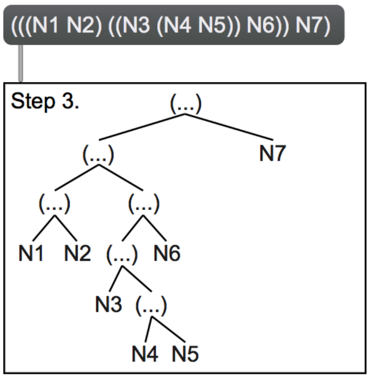


Figure 8. Tree of notes obtained after step 3.

B, given our choice of 'sharpest' and 'flattest' pitches). Again: $\sigma_7 = 0.94$, $\sigma_{-5} = 4.71$, $\sigma_5 = 0.94$, hence we choose $c_1 = 7$ as center of effect (since $\sigma_7 = 0.94 < 21/(3+1) = 5.25$ the solution is acceptable), and spell N_4 as B and both N_4, N_5 as C♯.

We move to $((N_3(N_4 N_5))N_6)$, with $\mathcal{C} = \{7, -5, 5, 0\}$, corresponding to C♯, D♭, B and C. Using $c_i = 7$ would respell the consecutive notes N_5 as C♯ and N_6 as C♮, which is unacceptable if (as is by default) we choose to discard altered repetitions. The best acceptable scenario is hence $c_i = -5$, $\sigma_{-5} = 4.15 < 21/(4+1) = 4.2$. We hence respell N_3 as B, both N_4 and N_5 as D♭ and N_6 as C.

We move to $(N_1 N_2)$, with $\mathcal{C} = \{9, -3, 1\}$, corresponding to D♯, E♭ and G. The best solution is $c_2 = -3$, with $\sigma_2 = 2 < 21/(2+1)$ yielding N_1 as E♭ and N_2 as G.

We move to $((N_1 N_2)((N_3(N_4 N_5))N_6))$, with $\mathcal{C} = \{9, -3, 1, 7, -5, 5, 0\}$, which on the other hand has no acceptable solutions, either because of the altered repetitions, or because the standard deviations being greater than $21/(6+1) = 3$. We do not respell this node, and we also delete from the list all nodes containing this one, i.e., the node $(((N_1 N_2)((N_3(N_4 N_5))N_6))N_7)$. This concludes our process (the final result is displayed in Figure 9).

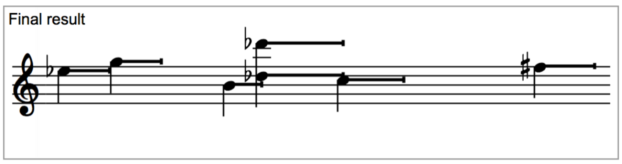


Figure 9. Final result.

5.4 Final considerations

The algorithm works for both bach.roll and bach.score and depends on the standard deviation threshold $\widetilde{\sigma}$. Such threshold can be set by the user, as shown in Figure 10. Higher values (or equations yielding higher values) for $\widetilde{\sigma}$ will allow respelling of larger temporal windows, at the expense of the quality of the transcription on smaller temporal windows (and at the expense of computation time).

Parameters for the $\widetilde{\sigma}$ function are 'numnotes' (the number of notes in the node to be respelled) and 'extension' (the temporal extension of the node in milliseconds). Among other things, one can fix spelling of chords only (as the ones in Figure 1) by providing a sufficiently high value for $\widetilde{\sigma}$ when the extension is 0, and a 0 value otherwise, e.g. $\widetilde{\sigma} = 1000000 * (\text{extension} == 0)$.

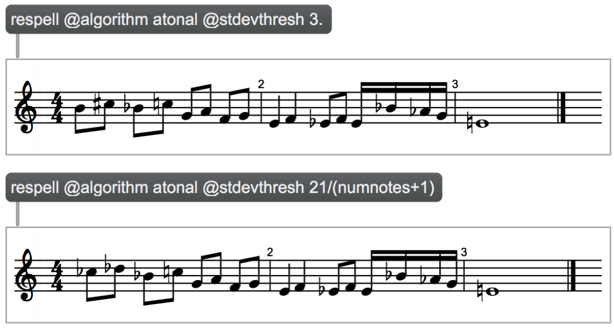


Figure 10. Different thresholds for $\widetilde{\sigma}$ affect the outcome.

Although the described algorithm provides a roughly natural respelling of general diatonic material, it also has two important shortcomings. For one thing, it is computationally expensive; notice, for instance, how respelling is performed multiple times on the notes N_4 and N_5 in the example above. The algorithm has been tailored for small portions of raw material and for short scores; as a consequences, for medium or large scores, the algorithm is essentially unusable in real time. To mitigate this issue, one can, however, adapt the equation for $\widetilde{\sigma}$ to only account for time extensions up to a certain threshold. In addition, given that the algorithm is based on a representation of diatonicism related to the line of fifths, it extends poorly on microtonal scenarios. The extension to microtonal music is of little concern for algorithms tailored on tonal music, such as the one by Chew and Chen; however, in our case, the possibility to improve the readability of microtones may constitute an important topic for future research.

6. CONCLUSION

We have presented a new framework for pitch representation in the *bach* library for Max, whose defining features are the ability to represent pitches with full enharmonic information, and the identification of pitches and intervals, meant to simplify and generalise the expression of arithmetic operations upon them. We also have described a novel algorithm for pitch respelling in the context of non-tonal music. We are aware of the fact that some aspects of this new system (in particular, the representation of intervals) might appear somehow confusing at first sight, but we hope that the simplification and generalisation they afford will outweight the initial difficulty, and that, overall, they will prove useful for implementing meaningful musical processes in a more straightforward and correct way than what the previous versions of *bach*, as well as other software tools, allow.

7. REFERENCES

[1] R. T. Kelley, "Reconciling tonal conflicts: Mod-7 transformations in chromatic music." [Online]. Available: http://www.robertkelleyphd.com/mod7.htm

[2] P. Tagg, *Everyday Tonality II*. The Mass Media Scholars Press, Inc., 2014-2016.

[3] A. Agostini and D. Ghisi, "Real-time computer-aided composition with *bach*," *Contemporary Music Review*, vol. 32, no. 1, pp. 41–48, 2013.

[4] R. Cohn, "Introduction to Neo-Riemannian Theory: A Survey and a Historical Perspective," *Journal of Music Theory*, vol. 42, no. 2, pp. 167–180, 1998.

[5] R. N. Shepard, "Circularity in judgements of relative pitch," *The Journal of the Acoustical Society of America*, vol. 36, no. 12, pp. 2346–2353, 1964.

[6] E. Chew, "Towards a mathematical model of tonality," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000.

[7] E. Agmon, "A Mathematical Model of the Diatonic System," *Journal of Music Theory*, vol. 33, no. 1, pp. 1–25, 1989.

[8] R. T. Kelley, "A mathematical model of tonal function," in *Annual Meeting of Music Theory Southeast*, Chapel Hill, NC, USA, 2006.

[9] A. R. Brinkman, *PASCAL Programming for Music Research*. University of Chicago Press, 1990.

[10] P. J. Clements, "A System for the Complete Enharmonic Encoding of Musical Pitches and Intervals," in *Proceedings of the International Computer Music Conference (ICMC)*, Den Haag, Netherlands, 1986.

[11] D. Meredith and G. A. Wiggins, "Comparing Pitch Spelling Algorithms," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, UK, 2005.

[12] E. Chew and Y.-C. Chen, "Determining context-defining windows: Pitch spelling using the spiral array," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Baltimore, USA, 2003.

A COGNITIVE DIMENSIONS APPROACH FOR THE DESIGN OF AN INTERACTIVE GENERATIVE SCORE EDITOR

Samuel J. Hunt
Creative Technologies Laboratory
UWE Bristol
Samuel.hunt@uwe.ac.uk

Tom Mitchell
Creative Technologies Laboratory
UWE Bristol
Tom.mitchell@uwe.ac.uk

Chris Nash
Creative Technologies Laboratory
UWE Bristol
Chris.nash@uwe.ac.uk

ABSTRACT

This paper describes how the Cognitive Dimensions of Notation can guide the design of algorithmic composition tools. Prior research has also used the cognitive dimensions for analysing interaction design for algorithmic composition software. This work aims to address the shortcomings of existing algorithmic composition software, by utilising the more commonly used score notation interfaces, rather than patch based or code based environments. The paper sets out design requirements in each dimension and presents these in the context of a software prototype. These principles are also applicable for general music composition systems.

1. INTRODUCTION

The primary focus of this research is to engage traditional composers with generative music systems. Existing tools for generative music composition require that users transition from traditional score editor workflows [1], into programming based environments, either graphical (like Max [2]) or textual (like Sonic Pi [3]). Little research has explored how algorithmic music techniques can be integrated into contemporary digital music composition work-flows (e.g. score editor-based, and digital audio workstation-based composition).

This paper discusses an ideal usability profile for a system that integrates generative music elements into a score editing workflow, considering the features required, under the Cognitive Dimensions of Music Notation framework. A prototype generative music system called the Interactive Generative Music Editor (IGME), is presented that considers the Cognitive Dimensions of Music Notation in its design in order to support interactive generative music. The research also shows directions in which digital score editors might develop to improve general usability.

2. COGNITIVE DIMENSIONS FRAMEWORK

Green and Petre [4] proposed the Cognitive Dimensions of Notations framework, as an evaluation technique for visual programming environments, interactive devices and non-interactive notations. Nash [5] has adapted this framework for use in designing and analyzing music notations and user interfaces for digital and traditional music practice and study. Bellingham [6] presents similar work, using the dimensions approach for analyzing a representative selection of user interfaces for algorithmic composition software. Finally, the cognitive dimensions can also be thought of as discussion tools for designers [6].

3. INTRODUCTION TO IGME

IGME is a score editor-based music sequencer that incorporates using generative and algorithmic techniques, designed to promote a human and computer cooperative creative system. A more detailed overview of IGME (previously named IGMSE) is given in [1].

The core design principles of IGME are as follows:

- Integrates algorithmic techniques for musical composition inside familiar score editing and music sequencing workflows.
- Provides full version control, for revisiting and comparing material.
- Uses graphical controls (WIMP) rather than code based interfaces.
- Takes a modular approach to composition, while retaining a linear timeline.
- Uses a multi-layered assembly stage that assembles the final score from individual parts.

IGME considers composition in terms of three distinct musical parts: human created content, computer generated content, or a mixture of both. The IGME program is divided into two main views: the arrange view (Figure 1) and edit view (Figure 2). The arrange view focuses on arranging and sequencing individual parts, using design principles found in other sequencers such as Logic Pro X [7]. The edit view (or detail view) allows the user to edit the individual music sequences, and/or specify the algorithmic effects for each part. A range of algorithmic effects are implemented by IGME, that can either augment human composed music, or generate computer created music. These are presented using an audio plugin metaphor, whereby control is given though simple graphical controls (see figure 4).

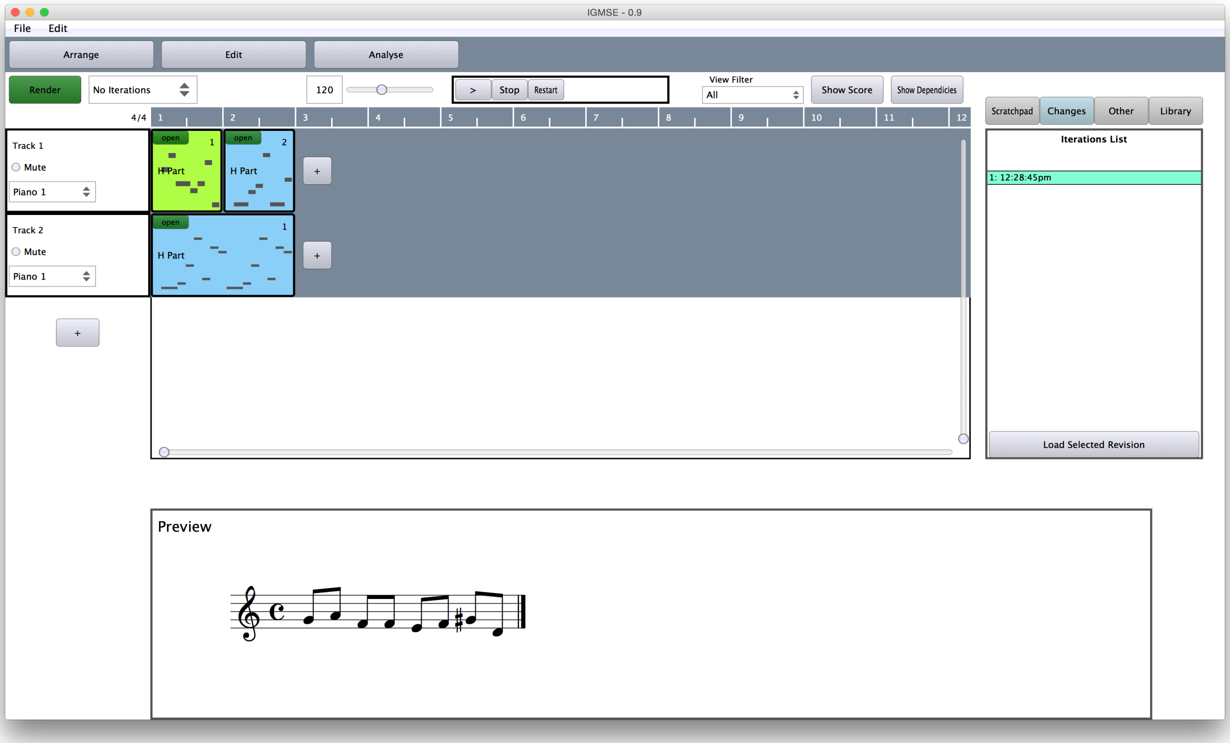


Figure 1. The arrange view in IGME.

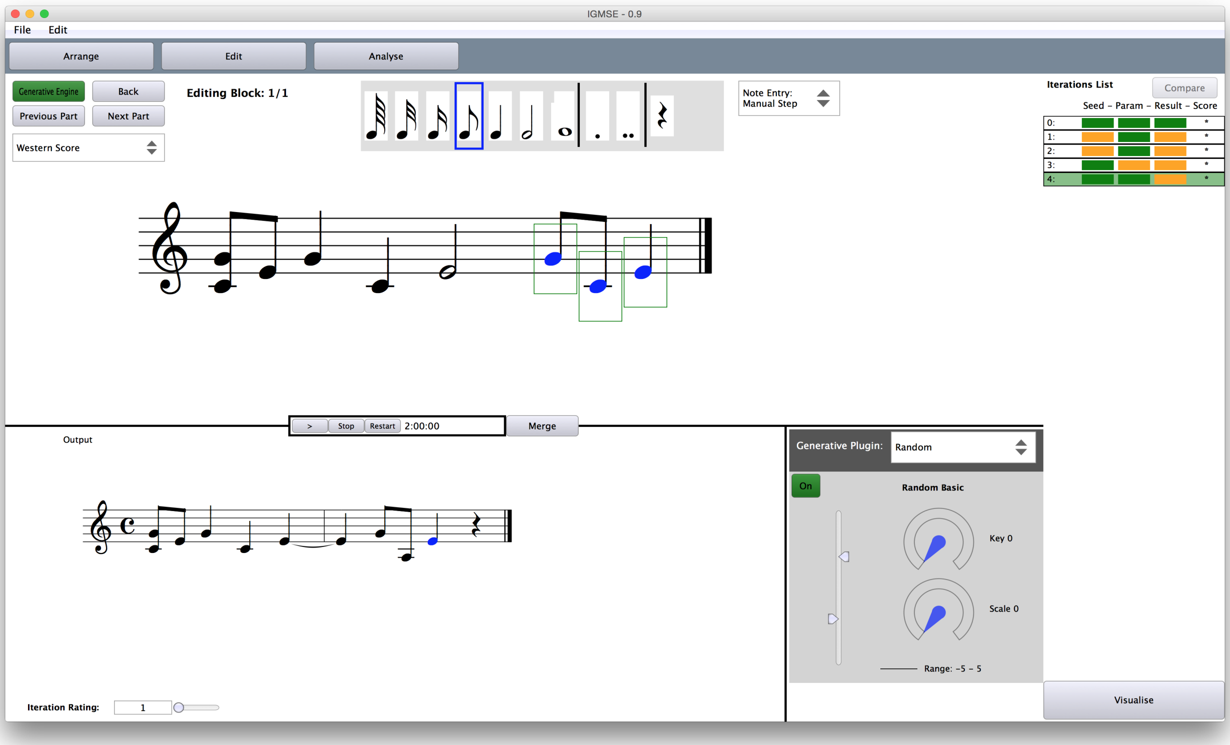


Figure 2. The edit view in IGME.

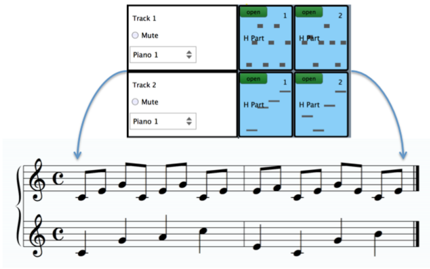


Figure 3. Each of the parts gets concatenated to form a final score.

Once the individual musical parts have been arranged on the timeline and any generative systems set up, the final score can be assembled. This process, referred to as the assembly stage, evaluates all of the computer generated parts and along with any human parts assembles them into a final score that can be auditioned and further inspected (see Figure 3).

A prominent feature necessary for supporting IGME is the version control system (Figure 1 & 2). This feature keeps track of all of the various edits made at both the individual part level and global level. This encourages experimentation with the generative system as it is not possible for the user to overwrite or loose material. This feature also allows the user to revisit previous material at will. Such a feature was originally proposed by Duignan [8], noting that such features are non-existent in music composition systems.

Two related and existing systems worth noting that also use score interfaces and have similar design themes are InScore [9] and OpenMusic [10]. InScore is an environment for the design of interactive augmented music scores, it is mainly used for the real-time playback of dynamic music. OpenMusic is a visual programming language, that is a convenient environment for music composition. Although both systems use score interfaces, InScore requires the learning of syntax, and OpenMusic has many of the issues associated with patch environments [6].

4. DIMENSIONS OF MUSIC NOTATION

The remainder of the paper takes each of the dimensions in turn and discusses them in context. The description for each dimension is taken from Nash's [5] work.

5. VISIBILITY

"How easy is it to view and find elements or parts of the music during editing?"

The software presents varying hierarchical levels of visibility, these can be loosely thought of as the detail view, arrange view, and global view. The detail view (Figure 2) shows the individual notes within each part, the processes attached to them, and the net result. The arrange view (Figure 1) shows the sequential order between parts, giving an indication of which parts are human created, or computer generated. Finally the global view (Figure 3) shows the final output as a score, after all processes have been applied, and is the musical content auditioned by the user, this is very similar to the kind of view seen in programs such as Sibelius [1] . There are few steps required to get from the arrange view down into the edit view. A criticism of patch based interfaces as stated by Bellingham [6] is that this structure is not often clear with many layers often spread across different windows. Finally selecting an individual note in the global view will highlight its parent part, so that it can be edited.

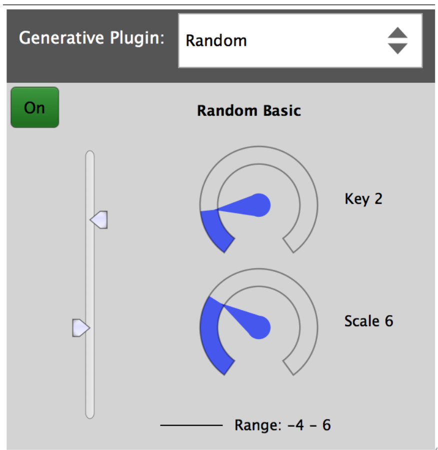


Figure 4. Random algorithmic effect, alters the pitch within a given range, also applies key/scale quantisation.

Generative effects in IGME are shown with graphical controls (Figure 4) similar to those found in audio/MIDI plugins, rather than using variables or number boxes. The values of the plugins are easier to see, compared with programming based systems. In programming based generative systems the value of a variable can sometimes be defined far away from where it is used, thus presenting additional debugging challenges.

Following design principles in similar programs, different tracks can be isolated in the view and shown against each other, giving great control over what is shown at once. The program can be split into two windows, one for the arrange and edit view and the other for permanently showing the overall score.

6. JUXTAPOSABILITY

"How easy is it to compare elements within the music?"

Specific iterations can be compared with a dedicated compare command. As each iteration and edit is retained, it allows different parts to be swapped in and out quickly and the result auditioned in context with the parts around it. The included *diff* tool (Figure 5) shows the explicit difference between parts. Bellingham et al [6] note that form-based systems such as Tune Smithy and the Algorithmic Composition Toolbox do not allow users to see older entries as they are replaced, this having a negative effect on

[1] http://www.avid.com/sibelius

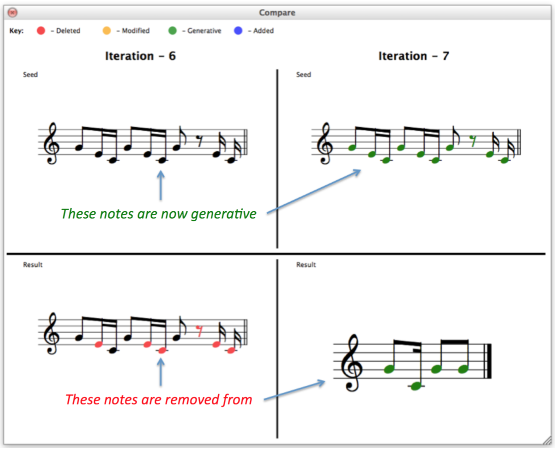


Figure 5. Diff tool comparing two different iterations.

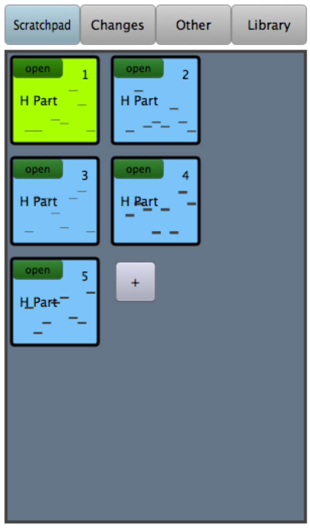


Figure 6. Scratchpad for creating musical ideas.

juxtaposability. Thus users are reliant on either using an undo, or keeping everything in working memory, increasing *hard mental operations*. These issues are mitigated by the version control techniques introduced in IGME.

Nash [5] notes that some musical characteristics are not obvious purely in the visual domain. For example some notes are dependent on key and transposition (for certain instruments) to fully decode, requiring the user to understand these relationships. Two solutions can be proposed respectively. First, the colour or shape of the note head could be modified to explicitly show that a note is for example not an E but an E flat, however this feature is possibly only useful for novice composers. Second, by using the IGME assembly process, transposed instruments can be notated as a normal instrument, and then be transposed into the correct transposition for the overall score during the assembly stage, this does however increase *hidden dependencies*. Both of these features have been encoded into IGME but remain optional at this stage.

It may be beneficial to compare elements in the music in a different domain to assess for example why a generative part is not suitable. IGME includes built in analysis tools that allow for the analysis and comparison of musical elements, showing for example the difference in pitch distributions between different sections. This may for example show quickly that track A is clashing with track B, because of the increased use of accidentals. This feature would be useful for normal, purely human compositions, but increasingly useful for compositions with elements of computer generated parts. For more information relating to visualizing musical elements in reference to the cognitive dimensions see [11].

Using programming based environments for generative music makes comparing elements in the music especially difficult. The program would need to be recompiled and the runtime output recorded into a third-party program for offline comparison. This issue is also true of environments such as Max/MSP. The ability for programming based environments to achieve the same effect in multiple ways can be prohibitive for this dimension. IGME supports the auditioning of music within the software itself; however the music can also be output as MIDI for synthesis or analysis in an external client.

7. HARD MENTAL OPERATIONS

"When writing music, are there difficult things to work out in your head?"

Existing systems for generative music often require the user to design and implement algorithmic techniques using either code or a graphical programming environment (e.g. Max/MSP), therefore placing a high mental load on the user. This software provides such techniques out of the box, focusing on using, rather than designing algorithms for music composition. A knock on effect is that these techniques end up being black boxes, where the user has little knowledge of what each technique is actually doing. There is a certain trade off between making each technique internally accessible, and making it simple to use on the outside.

Scores unlike other forms of notation and music software, require the user to have a fairly high literacy threshold of score notation [5]. As this software is designed for composers familiar with western notation, this issues is not a prominent one. However it is possible to switch from a score editor to a piano roll editor for those more familiar with the MIDI editing workflow, although the focus at this stage is on score editing.

The scratchpad feature of IGME helps reduce the cognitive load further by allowing users to offload their musical ideas, without needing to think about their final location in the overall structure. This feature works by simply allowing the user to create musical fragments in a separate window. These fragments can then later be dropped into the final arrangement (Figure 6). Finally the rapid entry methods discussed in section 12 also aid this dimension even further, allowing users to capture core primitive elements of the music and focusing later on the exactness of these elements.

Bellingham et al [6] stress the need for a clear visualization showing the signal flow between components. These issues are a common problem for systems without a clear indication of a timeline, for example coding, patching and offline systems. As IGME uses the timeline metaphor control moves from left to right, therefore the user is not required to predict control flow, reducing cognitive load.

8. PROGRESSIVE EVALUATION

"How easy is it to stop and check your progress during editing?"

Individual parts can be rapidly auditioned, however users of this system are required to manually iterate the assembly process so that individual parts can be heard in the context of other parts, adding a small amount of delay to the process. The user is able to toggle these arrange level iterations to happen automatically, mitigating this delay, but it must be first explained to each user. Mute and solo controls are present in IGME and have the same usage as the majority of music software.

Collins [12] states that evaluating the material is obviously important for music software, as iteration is a primary concept in composition. A user interacting with any form of composition software is likely to apply a trial and error approach, testing many different ideas and combinations. Nash [13] notes that a rapid edit-audition cycle contributes to having a high state of *flow*, a desirable mental state for users engaging with creative exercises such as music. The affordances offered by score editing software make it very easy to stop and audition parts at any point, and for the most part make quick edits. Code based environments, make such editing processes more complicated, often due to the need for recompiling. The introduction of compile time errors, can cause the composition workflow to stop all together, such compile and run time errors are prevalent in patch and code based environments. Compile time errors are eliminated with IGME, as all generative effects are pre-compiled. Like other non-generative music software, erroneous data is prevented from being entered by the restrictions imposed by the UI. The software makes heavy use of pop-up warnings. These features of IGME reduce *error-proneness* and increase *provisionality* also.

The version control system further aids in this dimension, as users can not only revisit their previous work, but also see how there compositions have progressed over time using the diff tool (Figure 5).

9. HIDDEN DEPENDENCIES

"How explicit are the relationships between related elements in the notation?"

An important consideration is the relationships between different musical elements at varying structural levels. A specific use case of IGME is that the individual parts can reference other parts. For example part 2 on track 1 can take its initial content from the output of part 1 on the same track. This facilities simple repeats, or more complex processed based music. To reduce the complexity of using reference parts, the specific dependencies can be highlighted by using unidirectional coloured arrows, an example of this is given in Figure 7. This feature is similar to the patch cable metaphor used in Max/MSP [2] and Reason [14]. Deleting events that have references triggers a warning ensuring the user is aware of the knock on effect of doing this.

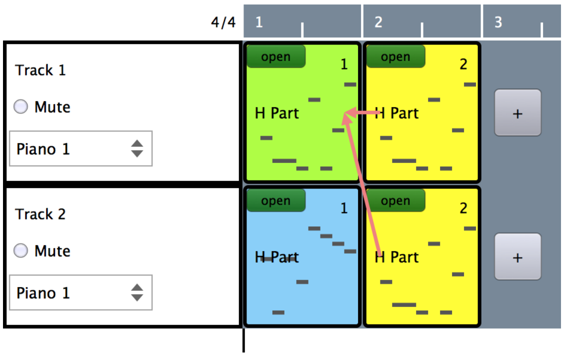


Figure 7. Part 1 on track 1 is being repeated (referenced) by 2 other parts.

Figure 8. The show dependants features, highlights all parts that are dependant on part 1 track 1.

Generative systems such as Max make dependencies more explicit by using patch cables [6]. However, the variables used in code based music systems, have more complex hidden dependencies. For example without manually searching for a variable it can be a challenge to see exactly where it is later used, and what effect changing it has for the overall output. Changing the internals of a particular function or patch can introduce knock-on effects if other parts of the program are dependant on the original behaviour. Even though IGME does not use code or patch based workflows, it can still have dependencies, especially for more complex arrangements. For example some of the purely generative effects work by analysing the surrounding musical content, changing or removing a musical part will alter the output of such generative processes. By shift clicking on a part in the arrange view, it will show what other parts are dependent on this part (Figure 8). Furthermore, pop-up warnings are presented at the point where making an edit would have a knock-on effect.

10. CONCISENESS / DIFFUSENESS

"How concise is the notation? Does it make good use of space?"

Nash [5] notes that western scores remain a concise form of notation interface, with the material shrinking and growing depending on the number of notes in a bar. However, issues present themselves in that score notation requires expert knowledge to decode the symbolic encoding of time. These decoding issues are not an issue for other forms of digital notation such as a step sequencer or piano roll; however both of these can take up considerable amounts of space.

The multiple views offered by IGME allows the representation of music at different hierarchical levels, improving the *conciseness* of the notation, also aiding with *visibility*.

Bellingham [6] notes that one should increase the verboseness in the language for variable names in coding environments, even if this has a negative effect on this dimension. This idea has been incorporated by giving parameter names more verbose names, for example"num of notes" gets expanded into "number of notes to be generated". The effect on overall user interface space usage is negligible.

11. PROVISIONALITY

"Is it possible to sketch things out and play with ideas without being too precise about the exact result?"

The most prominent idea introduced in IGME is the in-built version control technologies. This encourages experimentation as ideas cannot fundamentally be overridden or lost.

The *assembly* process allows users to rapidly enter note sequences, without the restrictions imposed by bar lines or time signatures. For example eliminating bar lines in the initial note entry process removes the need for tied notes that cross bars, as notes can simply be displayed as their absolute length. The later assembly stage takes care of creating these formalisms (see Figure 2).

A *scratchpad* feature (Figure 6) is presented in IGME that supports the user creating parts outside the scope of the arrange view. In most other software this could only be achieved by placing content many bars in the future or creating a new session entirely. This ability to create provisional material is a feature of Presonus [2]. Older iterations in a given part can be placed into the scratchpad so they can be later re-purposed.

A prominent feature is the note step option. When entering a new pitch for a selected note the user can switch between auto and manual step. Manual step means that entering a new note alters only the selected note, and the cursor does not increment to the next note. Auto step allows rapid entry as the cursor increments each time a new note is entered.

Nash [5] states that digital score notation interfaces are weak for supporting this dimension compared with paper

[2] https://www.presonus.com/products/Studio-One

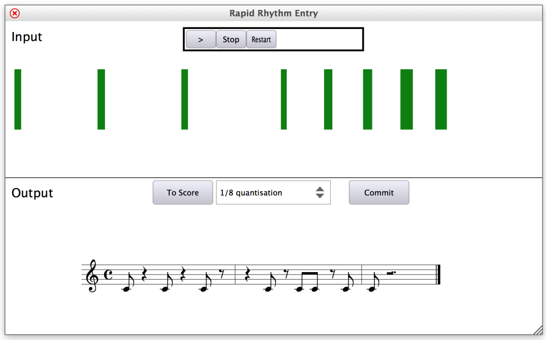


Figure 9. The Raid Rhythm Entry window allows user to enter rhythm patterns quickly.

notation which is far more flexible, as it allows for informal sketching. Programs such as Sibelius are beneficial for preparing final scores, and not necessarily for rapidly experimenting with ideas. A feature of IGME is the ability to rapidly enter notes, and this is supported by providing interfaces supporting a sketching metaphor. See *secondary notation* (Section 12) for more information.

Finally, Bellingham [6] notes that Impro-visor's [15] preset algorithms can be used for quickly creating musical sketches based on chord progressions. A prominent use case hypothesis for algorithmic music is that it can be used to generate new ideas, or to suggest augmentations to existing ideas, the extent to which such a feature is useful, is one of the major objectives of planned future research in evaluating IGME. Bellingham also notes that Logic Pro's in-built loops facilitate *provisionality*, as the content can be used as a place-holder and replaced later.

12. SECONDARY NOTATION

"How easy is it to make informal notes to capture ideas outside the formal rules of the notation?"

Nash notes that handwritten scores have an almost unlimited ability to make informal notes that can be interpreted by the user across the score. However, digital scores make this much harder through limited interaction with the keyboard and mouse. Staff Pad [3] offers a trade off between paper based and digital notation, in that scores can be sketched on a digital screen using a stylus, and the content typeset via advanced handwriting recognition, this is perhaps only useful for those already familiar with handwritten notation, it is also unclear how high the error rate would be. A key principle of IGME is providing ways to quickly enter sequences of notes, as briefly discussed in the previous dimension (*provisionality*).

IGME has a range of in-built tools for quickly capturing ideas, that can be formalised at a later stage. Figure 9 shows an example of one of these techniques that use secondary notation to quickly input material. The "Rapid Rhythm Entry" window allows user to simply tap a rhythm in using the space bar. The ascoiated pitches and exact rhythm can be edited later. These techniques support a

[3] https://www.staffpad.net/

kind of sketching [5] metaphor where informal ideas can be recorded quickly and effortlessly.

Bellingham [6] notes that adding colour to different elements of the interface can aid the usability of a program. For example logic pro can display each track as a different colour. Max allows different patch cables to be given different colours which could for example represent different types of signal flow (e.g. MIDI, mathematical, GUI controls). Bellingham [6] and Nash [5] both note that programming and patching environments support adding comments that can better explain the program between users and subsequent uses. Nash emphasises that limited provision is made in digital audio workstations for annotating music in any of the sub-notations or views. Both the parts and generative effects in IGME allow the user to write informal notes about their current choice of arrangement.

13. CONSISTENCY

"Where aspects of the notation mean similar things, is the similarity clear in the way they appear?"

A key design feature of IGME is that its design elements are consistent with other music sequencers. For example the arrange view and edit view are borrowed from similar elements in Logic Pro X, Cubase and Pro Tools. Editing music in score notation shares many features with Sibelius and other notation packages. An important design principle is that someone who is familiar with Logic Pro and Sibelius should find it easy to pick up and use IGME. The generative effects operate much like plug-ins, with presets and graphical controls.

Bellingham [6] notes that a consistent interface is easy to learn, so for IGME it is important that the interface is consistent so the new features introduced are easily picked up, so they can instead focus on exploring the novel features of the software. A criticism of many existing generative music systems is that they require the user to learn a new, often unfamiliar workflow.

This is one dimension where the changes made for other dimensions have had a knock on negative affect in this area. For example offering different forms of input notation to rapidly record new ideas, reduces *consistency* (as there are now multiple ways to achieve similar things) for an improvement in *provisionality* and *closenss of mapping* (to other interfaces more familiar to users).

14. VISCOSITY

"Is it easy to go back and make changes to the music?"

The editing stage in IGME has been designed with low *viscosity* in mind. The removal of bar lines for editing notes, means that users are not required to supply tie lines for notes that cross bar lines. Attempting to increase the length of a note in existing score editors, has knock-on *viscosity* [5], where the resultant effect will often discard notes from the end of a bar. Guitar pro [4] solves this in a novel way, by not removing anything, instead highlighting the bar as an error (in red), requiring a manual fix from the user. This suggestion of removing bar lines for speeding up the edit process is not novel, as it is also used in Dorico [5] .

Repetition viscosity [6] becomes an issue where sections of the music are copy and pasted to create repeats, and the user wishes to change the initial content. IGME's reference feature can mitigate this issue, as changing the initial part, causes all parts that reference it to update. A downside is there is a slight increase in hidden dependencies, although this has been addressed in other ways (see Section 9). By providing users with an easy way to use repeats, it could have consequences for both the musical quality and variety. The non-automated method may create happy accidents, whereas the proposal here may simply encourage rigid repetition. The exact effects cannot be determined at this time, but will be considered in future analyses of user interaction with the software.

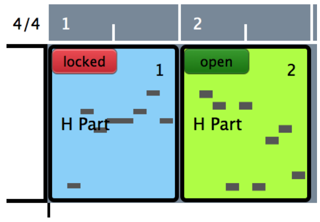


Figure 10. Part 1 is locked and cannot be moved or edited, whereas part 2 is open.

The inbuilt source control technologies inside IGME ensure that going back to make changes is encouraged. Some issues can arise whereby the user inserts a note that expands the part , for example from 1 bar into 2 bars, therefore requiring the surrounding parts to be moved around in context. However sync points can be used to ensure that future sections are preserved. The lock feature of IGME (Figure 10) prevents the position or internal content of a part being modified. These sync points and part locks create a trade off between *viscosity* and *provisionality*. The software does not therefore impose either option but hands over responsibility to the user.

Certain parts of the software have a *viscous* workflow, for example the generative plugins have a limited degree of control. Bellingham [6] notes highly viscous workflows can improve stability and create well defined use cases. This is not to say the effects are not powerful but have carefully designed user interfaces that facilitate a fluid experience for the user. Finally, Nash [5] notes increasing *viscosity* is a trade-off for avoiding *hidden dependencies*, this can be observed in patch and code based environments.

15. ROLE EXPRESSIVENESS

"Is it easy to see what each part is for, in the overall format of the notation?"

A key design requirement of IGME is that each design element makes use of existing metaphors, including staff no-

[4] https://www.guitar-pro.com/en/index.php?pg=buy-guitar-pro

[5] https://www.steinberg.net/en/products/dorico/start.html

tation (Sibelius), arrange/edit view (digital audio workstations), graphical controls (plug-ins) and part editing (most music sequencers). Using metaphors and graphical controls, allows a user to quickly understand the potential uses of each editor [5] [6].

In general, code and patch based environments are not as role expressive, unless the user is already familiar with the system. For a traditional score editor user, transitioning into a code/patch based environment presents a considerable learning curve. For example variables replace graphical controls, and without ensuring these have suitable names, confusion can arise in each variable's role. This issue can be prominent in MAX/MSP where some objects appear as text boxes [5].

A key consideration of integrating generative music is ensuring the user is aware of what notes are going to be altered by any generative process. This is done by changing the colour of the note, where green is used to indicate notes that may be processed, and black to show ones that are fixed (Figure 2). Parts are also coloured depending on their use, i.e. blue for a normal part or orange for a purely generative part (Figure 1).

16. PREMATURE COMMITMENT

"Do edits have to be performed in a prescribed order, requiring you to plan or think ahead?"

The timeline metaphor in IGME somewhat encourages a linear left-to-right workflow. However like score editing IGME supports various forms of development, including part-by-part, bar-by-bar or top down arrangements (form) [5]. The scratchpad (Figure 6) feature allows parts to be created offline and then placed back into the arrangement later on. The flexible editing methods offered by not enforcing bar lines, ensures that for example notes in the middle of bar can be edited easily at a later stage.

Bellingham [6] stresses that it should be important to have an option that states I don't know what is going here. Many sequencer based systems including score editors, inherently allow such gaps in the arrangement. More specialised generative music systems without the concept of a timeline are weaker in this category, and in general code based systems are not supportive of more structured or orchestrated compositions. Nash [5] notes that an advantage of using an arrange view metaphor, is it allows musical parts to be easily inserted, moved and copied.

Digital score editors inherently force several commitments from the outset, for example tempo, key and time signature. A planned feature of IGME is these can be autocompleted, whereby the user enters a sequence and the software analyses the musical features to predict tempo, key and the time signature. See Temperley [16] for more work in this area.

17. ERROR PRONENESS

"How easy is it to make annoying mistakes?"

As all generative effects are presented through a graphical UI rather than patching or code, users are generally protected from doing things that would otherwise break the underlying generative models. The assembly process takes care of adding formalisms that might otherwise be seen as errors, for example fixing bar lines with tied notes, and ensuring harmonic consistencies.

A feature of any generative music system is that the algorithms themselves can generate annoying musical mistakes, or wildly inappropriate musical material. The version control tracker features of IGME ensure many iterations can be experimented with, encouraging the user to tune the model to produce a more desired effect. It is difficult to appropriately tackle a subjective area such as music, as musical qualities deemed annoying mistakes by one composer, may be wholly appropriate by another.

18. CLOSENESS OF MAPPING

"Does the notation match how you describe the music yourself?"

A key advantage of the IGME assembly stage, is that different parts can have different notations for representing the musical material during the editing stage. At the assembly stage these parts can be converted into a single notation format for viewing and performing. Nash [5] notes that score representation is not an intuitive representation, but remains widespread especially for performers. It is therefore important to ensure that whatever notational interfaces are offered by the program it can still produce a score based output, especially when the programs musical output is to be performed by musicians. IGME does not yet support any novel notation interfaces but is considered for future versions.

As the generative effects are similar in nature to plugins, the individual effects values are expressed as simple GUI controls rather than variables or number boxes. For certain effects, terminology that aligns with how the sound is described is used. The use of graphical controls aligns with the visual metaphors offered by audio plugins and virtual instruments. Nash [5] notes that DAWs score highly in this dimension due to having interaction paradigms based on recording studio workflows.

19. ABSTRACTION MANAGEMENT

"How can the notation be customised, adapted, or used beyond its intended use?"

Green and Blackwell [17] describe three classes of software; abstraction-hungry systems, abstraction-tolerant systems and abstraction-hating systems. Programming and patch based environments rely heavily on abstractions, which has a negative effect on usability, especially for traditional composers transitioning into those new types of interfaces. Nash [5] notes that composers using paper scores are free to invent new notation techniques to describe music more concisely, digital score editors tend to be more limited. In addition Bellingham [6] states that Abstractions can be used to make software more effectively match the users mental model of the music they are notating.

Bellingham [6] states that *"An effective design would be for the software to have a low abstraction barrier but be abstraction-tolerant. Such a design would allow new users to work with the language without writing new abstractions, while more advanced users could write abstractions when appropriate."*

In general IGME's inbuilt processes for generative music are abstraction-hating as they cannot be customised internally, but can only be control through exposed GUI controls. IGME abstracts sequences of events into parts that can have further processes applied to them, and also reference and reuse each other. Overall the system is therefore abstraction tolerant. It is unclear without more conclusive user studies how powerful IGMEs inbuilt generative features and part referencing will be.

20. CONCLUSION

This paper has shown how many of the issues associated with generative music systems can be mitigated by transitioning into more traditional music sequencing workflows, eliminating the deficiencies offered by patch and code based environments. Many of the suggestions made in this paper, for example the version control system, would be beneficial for different music sequencing and composition software in general.

Future work will focus on testing the IGME software with composers, both in longitudinal studies and shorter workshop sessions, then evaluating each cogntive dimension in a similar way to Nash's [5] research. Another theme for future research will be integrating more advance algorithmic techniques for music creation, the primary aim of the overarching research objectives.

21. REFERENCES

[1] S. Hunt, C. Nash, and T. Mitchell, "Thoughts on Interactive Generative Music Composition," in *2nd Conference on Computer Simulation of Musical Creativity*, Milton Keynes, UK, 2017.

[2] V. J. Manzo, *Max/MSP/Jitter for Music: A Practical Guide to Developing Interactive Music Systems for Education and More*. Oxford University Press, 2016.

[3] S. Aaron and A. F. Blackwell, "From Sonic Pi to Overtone: Creative Musical Experiences with Domain-Specific and Functional Languages," in *Proceedings of the first ACM SIGPLAN workshop on Functional Art, Music, Modeling and Design (FARM'13)*, 2013, pp. 35–46.

[4] T. R. G. Green and M. Petre, "Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework," *Journal of Visual Languages & Computing*, vol. 7, no. 2, pp. 131–174, 1996.

[5] C. Nash, "The Cognitive Dimensions of Music Notations," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015.

[6] M. Bellingham, S. Holland, and P. Mulholland, "A cognitive dimensions analysis of interaction design for algorithmic composition software," in *Proceedings of Psychology of Programming Interest Group Annual Conference*, Brighton, UK, 2014.

[7] K. Anker and O. Merton, *Logic Pro X Power!: The Comprehensive Guide*. Cengage Learning, 2014.

[8] M. Duignan, J. Noble, and R. Biddle, "Abstraction and Activity in Computer-Mediated Music Production," *Computer Music Journal*, vol. 34, no. 4, pp. 22–33, 2010.

[9] D. Fober, Y. Orlarey, and S. Letz, "Inscore - an environment for the design of live music scores," in *Proceedings of the Linux Audio Conference*, CCRMA, Stanford University, USA, 2012, pp. 47–54.

[10] J. Bresson, C. Agon, and G. Assayag, "OpenMusic: Visual Programming Environment for Music Composition, Analysis and Research," in *Proceedings of the ACM International Conference on Multimedia: Open-Source Software Competition*, Scottsdale, AZ, USA, 2011, pp. 743–746.

[11] S. Hunt, C. Nash, and T. Mitchell, "How can music visualization techniques reveal different perspectives on musical structure," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017.

[12] D. Collins, "A synthesis process model of creative thinking in music composition," *Psychology of music*, vol. 33, no. 2, pp. 193–216, 2005.

[13] C. Nash and A. Blackwell, "Liveness and Flow in Notation Use," in *International Conference on New Interfaces for Musical Expression (NIME)*, Ann Arbor, MI, USA, 2012.

[14] M. Duignan, J. Noble, P. Barr, and R. Biddle, "Metaphors for Electronic Music Production in Reason," in *Proceedings of the Asia Pacific Conference on Computer Human Interaction (APCHI)*, Rotorua, New Zealand, 2004.

[15] R. M. Keller and D. R. Morrison, "A Grammatical Approach to Automatic Improvisation," in *Proceedings of the Sound and Music Computing conference (SMC)*, Lefkada, Greece, 2007.

[16] D. Temperley, *Music and probability*. The MIT Press, 2007.

[17] J. M. Carroll, *HCI models, theories, and frameworks: Toward a multidisciplinary science*. Morgan Kaufmann, 2003.

MAXSCORE: RECENT DEVELOPMENTS

Georg Hajdu
Hamburg University of Music and Drama
georg.hajdu@hfmt-hamburg.de

Nick Didkovsky
Algomusic.com
nick@didkovsky.com

ABSTRACT

This paper presents recent development in MaxScore and its peripheral applications. These developments include:

- Adding new functionality to the core mxj object including details on our implementation of an undo/redo stack, new licensing models, custom beam groups, and other new features.
- Strategies to achieve proportional notation, with a look to the future.
- Expanding the feature set of the MaxScore and LiveScore Editors which include new style editors for the design of non-standard clefs, tablature notation and Bohlen-Pierce microtonality.
- Providing tools for greater compatibility with other third-party developments such as bach, Mira, the Scala Archive as well as the conTimbre sample library and its ePlayer.
- New peripheral components for guided improvisation and situated scores.
- Strategies to achieve proportional notation, with a look to the future.

1. INTRODUCTION

MaxScore is a notation package for Max consisting of a core mxj object referred to as “MaxScore object” implementing the Java Music Specification Language, and a number of peripheral abstractions and devices [1]. A complete music editor with menus and floating palettes exists in form of the MaxScore Editor. Some of MaxScore’s functionality has been integrated under the moniker LiveScore into Ableton Live via the Max for Live API. MaxScore shares some features with the bach and cage computer-aided composition packages for Max [2] and to a lesser extent with Inscore [3], but is set apart from them by its capability to render to arbitrary contexts as the engines for data handling and graphics rendering are separate entities.

2. RECENT ADDITIONS TO THE MAXSCORE OBJECT

A number of new features have been added to MaxScore. Some of these, like the Undo/Redo stack, were implemented in the core JMSL engine (Java Music Specification Language) that powers MaxScore, while others like new low-cost licensing options, primarily affect MaxScore.

2.1 Undo/Redo

JMSL’s Score package originally implemented a fine-grained undo/redo mechanism using a Command Pattern [4]. With this scheme, a user action that affected a *score*, such as doubling the duration of a *note*, was encapsulated in a *command*. The *command* included an undo operation, in this case, halving the duration of the note. *Commands* were added to a stack, and a user’s Undo action would pop the top *command* off the stack and execute its custom *undo* operation.

The Command Pattern implementation of an Undo stack worked well for the subset of actions that had *commands* implemented for them. However, as JMSL and MaxScore expose a general API to the user as well as a user interface (indeed Max itself is a GUI), it became difficult to decide at what level undo/redo should be implemented. If the user mouse-clicks a staff, a new *note* is inserted and was undoable because the UI action was wrapped into the Command Pattern. However, the same user may use the API to insert a *note* using the *addNote* message in Max. The *addnote* message is an elemental API call that does not trigger an undoable *command*. Furthermore, the Max user may patch together an arbitrarily complex network of similarly fundamental API procedures which insert, delete, and transform existing notes. We wanted to give that user a functioning *undo/redo* stack and decided something closer to the “Memento Pattern” would be appropriate.

The current *undo/redo* scheme in JMSL’s Score package addressed these issues by building a stack of *score* clones instead of a stack of undoable commands. Actions that altered the contents of a *score* trigger the saving of the entire Score to a cache. Undo replaces the current score with the clone at the top of the stack. At first, we were concerned that the user would experience unacceptable pauses while editing as the score was being written to the cache, but in practice we discovered that writing a *score* to the disk cache is almost unnoticeable, even with large scores. The MaxScore user has also been given more control over the *undo/redo* stacks, with new saveToUndoStack, undo, and redo messages. The saveToUndoStack message takes a snapshot of the score's current state and saves it to the undo stack. This allows the user to make arbitrary programmatic changes to the score, i.e. non-UI commands that do not trigger UNDO stack snapshots, save to the undo stack and undo the activity if desired.

This scheme required a layer of programming to provide the user with the sense that the score displayed by an undo/redo event felt like the same score, even though it was actually replaced with a clone. The starting measure of the layout, for example, had to be cached along with the score to restore the current layout. We may add other such features such as restoring the current note selection.

2.2 Proportional Notation

User demand for proportional notation is currently addressed with two different strategies. One method is using linear measure widths and adjusting the base of Blostein/Haken justification algorithm, which is responsible for the influence of duration on horizontal note placement [5]. Another technique is to use invisible rests to fill in time space. The latter is more accurate but a more cumbersome solution.

Figure 1-3 show various layout schemes in JMSL. The default layout uses flexible measure widths, where measures are algorithmically widened to accommodate denser note layout. Changing the layout to linear measure widths ensures that all measures have the same width, a prerequisite to proportional notation. Changing the Blostein/Haken justifier algorithm's base from a default of 0.7 to 0.4 results in a layout that comes close to proportional notation.

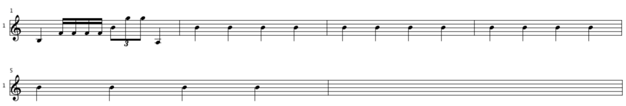



Figure 1. JMSL Score's default layout uses flexible measure widths and a Blostein/Haken justification base 0.7.

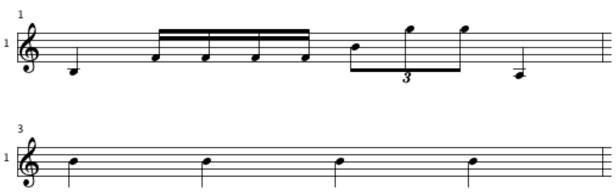



Figure 2. Changing JMSL's measure width to Linear results in all measures being the same width. Notice that the quarter notes in the measure 3 do not align with the notes in measure 1 due to justification algorithm JMSL uses.

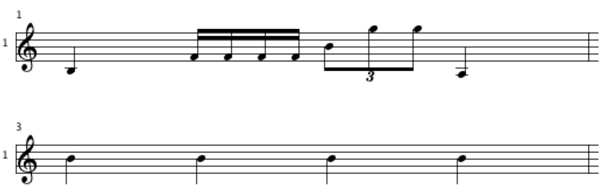



Figure 3. Linear measure widths and a Blostein/Haken justification base of 0.4 comes close to proportional notation. Notice that quarter notes in measure 3 align closely with the notes in measure 1.

An alternative technique to achieve proportional notation is to choose a fine time granularity, say 64th notes, and filling a measure with Linear width with these notes. These notes will all be spaced evenly, and the user may change some of them to invisible rests, either by hand or preferably using a straightforward algorithm. Figure 4 shows this technique, which additionally made stems and beams invisible.

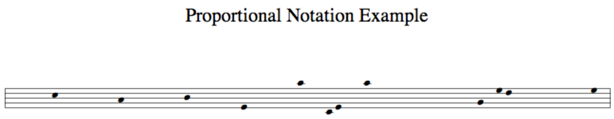


Figure 4. This example ensures accurate time-based note placement required by proportional notation. Horizontal space between note heads is occupied by evenly spaced invisible rests.

A useful plug-in to generate proportional notation using this technique could be created in a straightforward way. With traditionally notated durations as input, the plugin ought simply to quantize their durations to the nearest 64th note to fit into this scheme. JMSL and MaxScore's "Unary Copy Buffer Transform" API (reference to JMSL paper) would serve well as the plug-in platform.

We are developing a new strategy to achieve proportional notation using an underlying data representation and a layout manager that is robust and flexible. This approach will address the shortcomings of the two approaches discussed above, and will be visually precise and free of an underlying quantization grid.

2.3 New MaxScore licenses

We have developed new license levels to accommodate users' needs. The latest is the low-cost ($9.90) LIVE_LITE license, used by composers who wish to use MaxScore in the Ableton Live environment without the ability to edit in MaxScore or develop using JMSL's Java API. JMSL's license scheme accommodates new license types transparently, whose semantics are interpreted programmatically. We have found that a fair number of new users have been very satisfied with the limited but focused functionality of using the new LIVE_LITE license to bring traditional notation into Ableton Live.

2.4 Beam Grouping

Beam grouping is a new MaxScore feature, delivered by JMSL's "BeamGroupTransform", a NotePropertiesTransform which is addressed from Max via a few simple messages sent to MaxScore. A BeamGroup is a specification of how to group notes in a particular time signature. Notes in a measure of 7/8, for example, may be beamed as groups of 2+3+2 or as 3+2+2 or other combinations. The MaxScore user specifies a BeamGroup with the message addBeamGroup <timeSigUpper timeSigLower g1 g2 g3 g4...> , where g1+g2+...+gn add up to the number of beats in the measure as specified by the upper number in the time signature. For example, the following message:

addBeamGroup 7 8 2 3 2

…specifies that a measure of 7/8 should be grouped as 2+3+2 while the following message:

addBeamGroup 6 8 3 3

…specifies that a measure of 6/8 be grouped as two groups of three. Once the user specifies all such beam grouping preferences, the beamGroupTransform message executes this custom beaming on all selected notes, as Figure 5 illustrates.

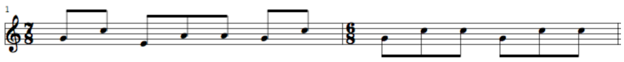



Figure 5. Results of beam grouping, where a measure of 7/8 is grouped as 2+3+2 and a measure of 6/8 is grouped as 3+3.

3. MAXSCORE EDITOR: NEW FEATURES AND TOOLS

3.1 Staff Styles

Staff Styles have been implemented in the MaxScore Editor to enable different representations of musical content, primarily for non-standard notation. Staff Styles rely on a plugin structure which has been described in [6]. The plugins talk to the MaxScore object via a JavaScript object mapping pitch to an arbitrary position of on a staff irrespective of its actual frequency and keeping track of the latter by using a MaxScore *note dimension* called originalPitch. Plugins for notation in the context of the Bohlen-Pierce scale and other microtonal scales have already been created, yet, recently, three new Staff Styles editors (which allow greater variability and flexibility) have been added to the repertoire.

3.1.1 Clef Designer

The JMSL API features a limited number of clefs, namely treble, alto, tenor, bass and percussion clef. The Clef Designer (Figure 6) was created to overcome this impasse by adding another 15 clefs or multi-clef staves (such as the OpenMusic-style FFGG staff [7], Figure 7) as well as providing an interface for the creation of non-standard, user-defined clefs (see Figure 8).

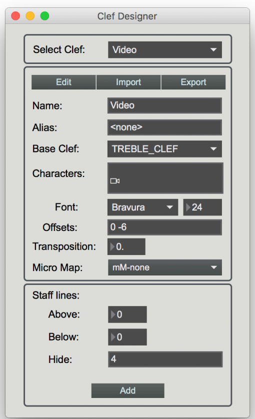


Figure 6. Screen shot of the Clef Designer GUI.

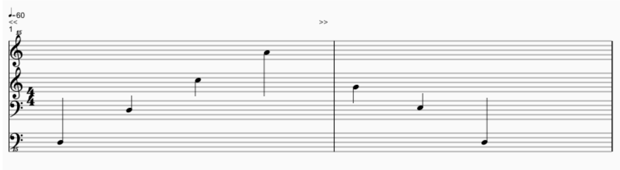



Figure 7. The FFGG staff settable in the Clef Designer.

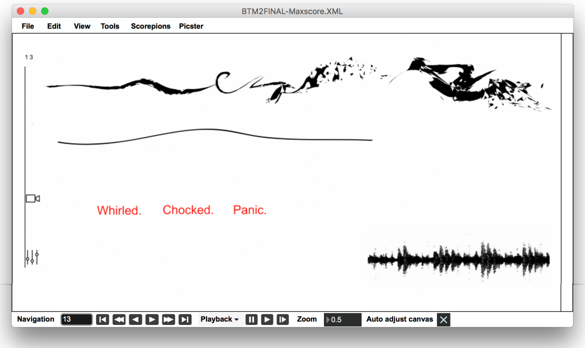


Figure 8. A score by Vietnamese composer Luong Hue Trinh using a non-standard clef for text display on the third staff.

3.1.2 Tablature

Tablature is supported by another editor allowing users to define an arbitrary number of strings as well as fret intervals, both set to pitches in floating-point precision. This, for instance, permits tablature notation of the 10-string 41-tone guitar used in Hajdu's piece *Burning Petrol* [8] (Figure 9).

The editor features 21 presets from monochord to 19-course theorbo which can be used as templates for user-defined tablatures. As with the Clef Designer, user-defined tablatures can be saved into scores they been created for, from where they can be exported as files and imported to other scores. Notes can be dragged to other strings for alternate fingerings and shifted up and down by using arrow keys.

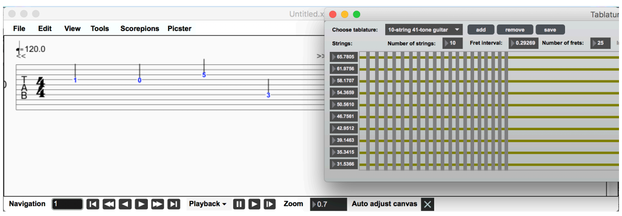


Figure 9. The GUI to the Tablature editor featuring the preset for the 10-string 41-tone guitar (foreground). A short score in the corresponding tablature is seen in the background.

Two things are still on our agenda:

1. The implementation of an intelligent algorithm for fingering, both vertically (chords) and horizontally (melody and chord progression) using constraints and/or neural nets [8].
2. Adaptation of the editor to just-intonation instruments with individual, unequal fret positions. This

poses a particular challenge as two frets can be close together representing tuning alternatives for the same scale degree (e.g. 16/9 and 9/5) or spaced widely apart, possibly even skipping a scale degree. Personal communication with guitarist John Schneider emphasized that just-intonation guitar community is still far from defining a common standard for such scenarios.

3.1.3 Bohlen-Pierce microtonal notation

The Bohlen-Pierce scale is a *macrotonal* tuning dividing the just twelfth (“tritave”) into 13 steps. It exists in just and equal-tempered versions, the latter with a step size of 146.3 cents. The chromatic Müller-Hajdu notation has been described in [10] and implemented in the MaxScore Editors as Staff Style. Two subdivisions of the BP scale deserve particular attention:

1. BP triple scale also known as 39ED3
2. BP quintuple scale (65ED3) (whose step size deviates just 0.03 cents from 41-tone equal temperament)

We created an editor which accommodates the aforementioned microtonal BP scales using accidentals from the Bravura font set (Table 1), as partially suggested by clarinetist Nora-Louise Müller.

Steps (195ED3)	Glyph	Reference (see SMUFL [9])
0		natural
5		accidentalXenakisOneThirdToneSharp
10		accidentalXenakisTwoThirdTonesSharp
-5		accidentalWyschnegradsky3TwelfthsFlat
-10		accidentalWyschnegradsky9TwelfthsFlat
3		accSagittal11MediumDiesisUp
6		accSagittalSharp
9		accSagittalSharp11MUp
12		accSagittalDoubleSharp
-3		accSagittal11MediumDiesisDown
-6		accSagittalFlat
-9		accSagittalSharp11MDown
-12		accSagittalDoubleFlat

Table 1. The accidental set for the 39ED3 and 65ED3 Bohlen-Pierce microtonal scales. The indices in the left column refer to the LCM of both scales.

3.2 Expressions

The MaxScore object offers a variety of options to expand its feature set via *note dimensions* and *rendered messages* [1]. Note dimensions are referred to by a numeric or symbolic index and a floating-point value and need to be defined before notes are added to a score. These values are being added to a note event in the order of their index and sent out of the object during playback. In turn, rendered messages consists of single strings (or symbols in Max lingo) applied to notes, intervals, staves as well as measures and are sent out when the object renders to its drawing context. Expressions offer a way to combine the two, so that the messages to be sequenced (an action) are associated with a graphical element symbolizing the action to be performed. As an action can be more than just a single float (e.g. an OSC message with a number of values) the built-in limitation of the MaxScore object was overcome by writing messages into a buffer (a Max coll object) and referring to them by an index sent out during playback (the floating-point value). The buffer is created and updated whenever a score is loaded or events or Expressions are added to it. Expressions are created via the addRenderedMessageTo… family of messages, e.g. addRenderedMessageToSelectedNotes 0 0 "expression Coda[0] 153.3ocSN1kCBBCDD1ixDN.jVPdw27B3c.jARSvVrs.IR3EuQdx7J31HlXxlM67s+MdZaoms3DVgRxc99Fnx2ihppbnSXmMhryCNSfVbgKYn3HJjNKdSjQWZ5RcNd+7EJE7HsAyCJKqwFB79DsW+6OqfslnyKkMic3FCg5dJpHCQLWOLkDZk5qQz+bjZmk.XvXYsWt+1gOvm.fiM". The two zeros in the message refer to the initial coordinates of the part to be rendered (i.e. graphics), which can later be adjusted by dragging the graphics to another position, while the long string after Coda(0) is a Base64-compressed Max dictionary (Example 1).


```
{
	"rendered" :	{
		"0" : [ "frgb", 0.0, 0.0, 0.0, 255.0 ],
		"1" : [ "font", " Times Italic", 18 ],
		"2" : [ "writeto", 0.0, 31.0, "dal niente" ]
	}
,
	"sequenced" :	{
		"0" :		{
			"editor" : "bpf",
			"message" : "/amplitude",
			"value" : [ 1000.0, 0.0, 1.0, 0.0,
0.0, 0, 1000.0, 1.0, 0, "linear" ],
			"autorender" : "false"
		}
	}
}
```


Example 1. An Expression consists of “rendered” and “sequenced” messages.

Example 1 shows an example for an Expression in JSON format. It consists of the two keys “rendered” and “sequenced”, each holding an arbitrary number of entries. The sequenced dictionaries contain the keys “editor”, “message”, “value” and “autorender”, with the latter denoting whether or not the MaxScore editor should try to render the values irrespective of the drawing instructions given by the “rendered” dictionary.

The MaxScore Editor currently offers three editors (Figure 10) for the creation of sequenced messages:

- a generic editor with a text field for the message name and another for its values;
- an editor for breakpoint functions;
- an editor for DJster [11] parameter settings.

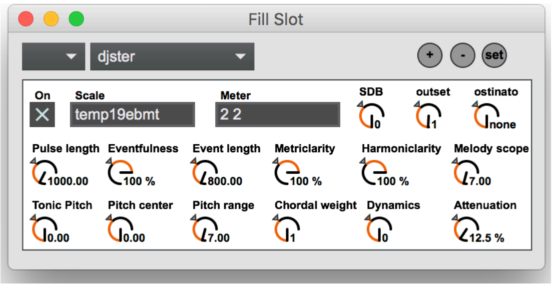


Figure 10. Example of an editor for sequenced messages.

In a MaxScore XML file, Expressions are stored as the Message attribute of a <userBean> element (of a <note> or <interval> parent element). It consists of a string containing a compressed Max dictionary preceded by "expression" and a symbolic reference (Example 2).


```
<note NOTEDUR="2" TUPLET="0" DOTS="0" ACCINFO="0" DURATION="1.0" PITCH="71.0" VELOCITY="0.5" HOLD="0.800000011920929" BEAMEDOUT="false" GLISSOUT="false" TIEDOUT="false" ACCPREF="0" ACCVISPOLICY="0" ALTENHARMONIC="false" DYN="0" SLUROUT="false" ISGRACENOTE="false" GRACENOTESEPARATIONSCALER="2.0" LEDGERLINESVISIBLE="true" WEDGE="none" OTTAVA="none" MARK="0" TEXTOFFSETX="0" TEXTOFFSETY="0" NOTEHEAD="0" NOTEHEADSCALE="1.0" VISIBLE="true" NOTEHEADVISIBLE="true" STEMVISIBLE="true" OVERRIDELEVEL="-1" ISOVERRIDELEVEL="false" STEMINFOOVERRIDE="false" STEMINFO="2" TEXT="" >
<dim index="4" value="0.0" name="EventFlag" />
<dim index="5" value="71.0" name="originalPitch" />
<dim index="6" value="1.0" name="index" />
<userBean CLASSNAME="com.softsynth.jmsl.score.util.RenderedMessageBean" Name="RenderedMessageBean_note-sel" Message="expression Coda[0] 159.3ocSO9lBBCCCE2ixid.lsSFHdC76dAxVyjBccy9G8Cic2MUmfPHj7KgWdIxAKGYKtfUnk7X7dOzM6QaWWCLU7bHC0M2Dmv0L4cCJXNiVYzqnKy4455mLMPYIOBNNjYE1PheT3vveWXEr0kmiRY+xHDESzcV5NRSKdWtXY7j7kJxn0eMh4miz6rJ.dWfoHnhH2mGo5TxmX4vaGdyE.8yM" Xoffset="0.0" Yoffset="0.0" >
</userBean>
</note>
```


Example 2. Expression are stored as compressed Max dictionaries. The <dim> element with index="6" attribute contains the reference (value="1.0") to the message contained in the buffer.

3.2.1 Button Mode

MaxScore has two modes for mouse interaction:

- one for editing notes and other score elements;
- one for repositioning and deleting Expressions and Picster [1] elements.

These modes can be toggled by using caps lock. When clicking on a graphics element in Picster mode, it is highlighted by a red bounding rectangle. We are planning to implement a *button mode* which would allow a Mira user (see section 4.2) to use Expressions as interactive score elements, sending out "sequenced" messages upon clicking on them—thus bridging the gap between score and interface[1]. Jacob Sello's Hexenkessel project [12] is an excellent example for such an approach developed at the HfMT Hamburg.

3.3 Searchable Scala database

The Scala Archive is the world's largest collection of microtonal scales maintained by Manuel op de Coul[2]. It is supported by an increasing number of third-party applications such as the Kontakt sampler and the MuseScore notation editor, among many others.

The Scala Archive currently contains more than 4500 scales and tunings—a number that makes informed choices staggeringly difficult. We have therefore created a searchable database via the Max SQLite JavaScript implementation. Searches can be performed according to

- number of *steps*,
- *pitch content* in terms of both floating-point and rational numbers
- *strings* contained in the comment section of a Scala file.

The database is integrated into the Scala Browser, a Max patch we refer to as a "virtual keyboard". The interface of the Scala Browser displays notes of the scales which can be clicked on to add notes to a score or change their pitch (Figure 11). A MIDI keyboard can be used instead of or in addition to mouse clicks.

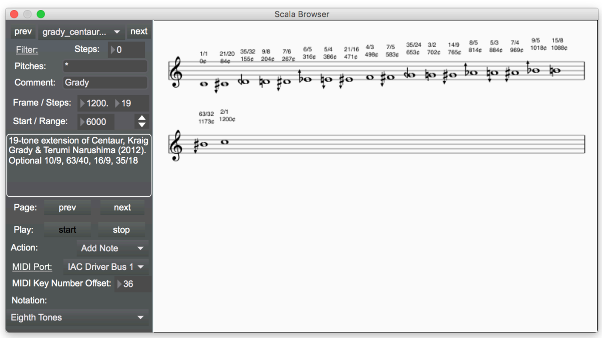


Figure 11. The Scala Browser virtual keyboard containing the Scala Archive as searchable database. The browser filters all scales containing the "Grady" search string in their comment section.

[1] At the HfMT, we have dedicated the UMIS research project (Unified Musical Instrument Surfaces) to the idea that an instrument can also act as a controller and score display.
[2] http://www.huygens-fokker.org/docs/scales.zip

3.4 MaxScore.NetCanvas

MaxScore.NetCanvas is a Java-based peripheral component of MaxScore developed by Benedict Carey, designed to render scores in web browsers via web socket connections [15].

In the latest update the communication between MaxScore and MaxScore.NetCanvas now occurs entirely within the Max environment, doing away with the previous reliance on inter-application messaging. This has the effect of speeding up communication between Max and remote clients, and simplifies the setup procedure for MaxScore users. The set of messages accepted by the MaxScore.NetCanvas object has expanded to include messages specific to part rendering, the behavior of cursors and control of the server (configuration, starting and stopping the websocket server and the new fileserver for serving the html client files). Max users can run multiple instances of MaxScore.NetCanvas concurrently. The new helpfile (Figure 12) contains information about how to use the new MaxScore.NetCanvas abstraction and accompanying mxj object; the source is available on Github.

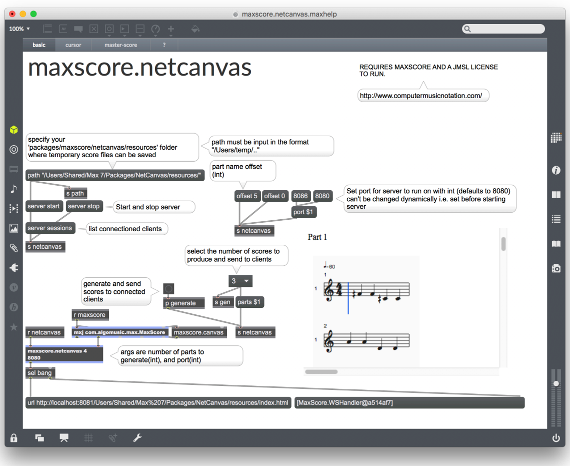


Figure 12. The helpfile to MaxScore.NetCanvas.

4. INTERFACING WITH OTHER THIRD-PARTY DEVELOPMENTS

4.1 bach compatibility

bach is a Max package developed by Andrea Agostini and Daniele Ghisi which has become the de facto standard for computer-aided composition in Max for its thorough integration and plethora of tools such as the bach.roll and bach.score notation objects [2]. Being modelled after IRCAM's OpenMusic environment its externals implement a data format which due to its similarity to the LISP syntax is called llll (lisp-like linked lists). Despite some overlap, bach.score and MaxScore occupy different niches of real-time notation ecosystem. While bach.score excels at continuous tempo changes, polymeter, nested tuplets and some GUI operations, MaxScore shows more flexibility in how it performs graphics rendering. By separating the mxj object designated for data handling from its drawing context, MaxScore can be used for generating PDFs, for data mapping as well as rendering to various 2D and 3D contexts. We therefore have created the maxscore.bachScoreToMaxScore abstraction capable of bridging bach.score with the MaxScore Editor with the aim to preserve as much information as possible (i.e. by translating bach.score's slot and pitch-bend data into corresponding note attributes and userBeans, see Figure 13).

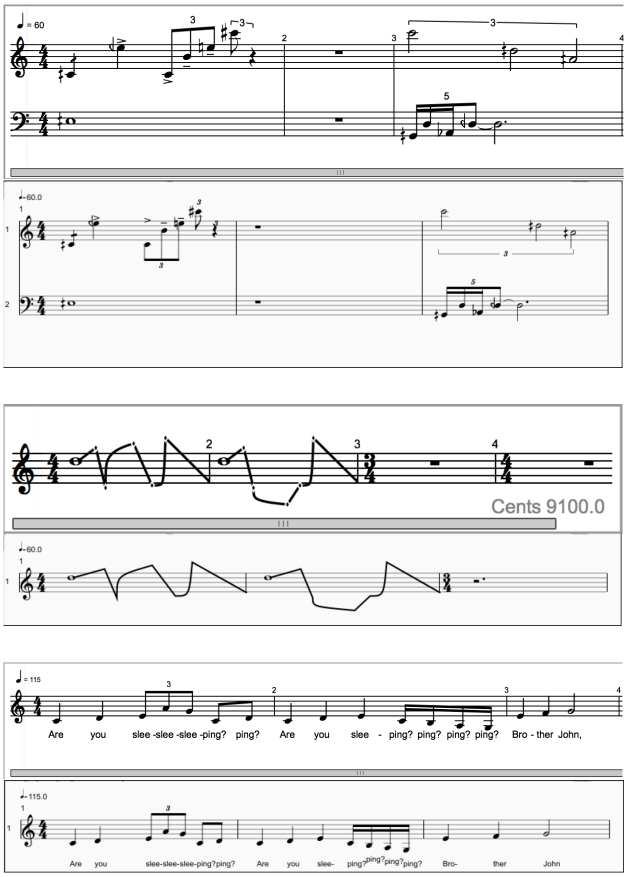



Figure 13. Translations between bach.score and MaxScore, featuring microtones (top), break-point functions (center) and text slots (bottom) performed by the maxscore.bachScoreToMaxScore abstraction.

4.2 Mira and MiraWeb

Mira and MiraWeb are technologies (the latter based on xebra.js) developed by Florian Demmer for Cycling '74, capable of mirroring Max GUI objects such as sliders, buttons, messages and comments in a dedicated iOS application and/or web browsers[3]. It therefore constitutes a perfect companion to Max by harnessing the multi-touch power of iPads, tablet computers or smartphones. The built-in *zeroconf* technology and automatic mirroring (Max objects simply need to be dragged onto an object called mira.frame) make Mira and MiraWeb a superior choice in comparison to alternative applications (e.g. TouchOSC, Lemur or C74).

[3] https://cycling74.com/articles/content-you-need-miraweb

Yet, dedicated notation objects mirroring objects such as nslider or bach.roll are currently out of reach as this would require a major development effort on the side of Cycling ’74 or third-party developers. However, Mira and MiraWeb support the fpic object capable of dynamically loading and displaying images. We have taken advantage of this by creating an abstraction called MaxScore.toMira. In this scenario, MaxScore renders to a Jitter matrix object via the embedded jit.render2MaxScore abstraction (Figure 14). Upon exporting the matrix as an image to a temporary location, the fpic object is prompted to load and display this image after a short delay. This image is then transferred over the network to the Mira client. MaxScore.toMira performs adjusts automatically to score dimension and dynamically scales the multi-touch information it receives from the mira.touch object to support user interaction with a MaxScore object.

This approach is efficient enough to create the illusion of a dedicated notation object and thus offers the only seamless solution to date enabling users to interface with Max through music notation also supporting bach.score via the maxscore.bachScoreToMaxScore abstraction (see Figure 15.

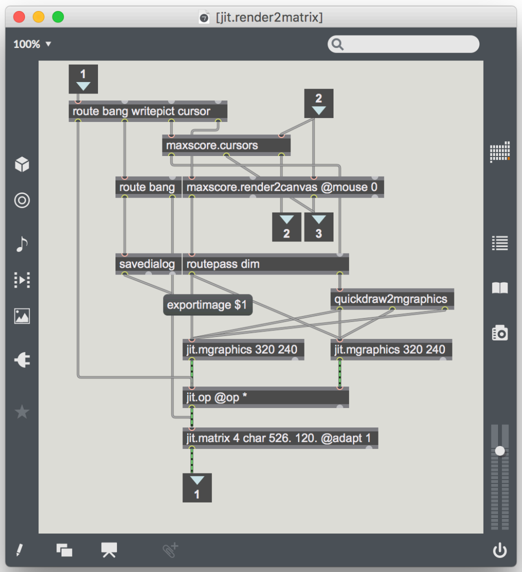


Figure 14. Rendering to Jitter enables MaxScore to save a score as an image.

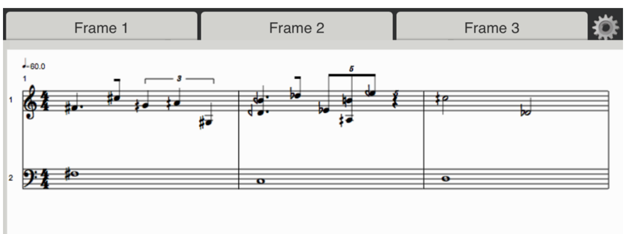


Figure 15. The MaxScore.toMira abstraction allows users to display scores on multi-touch devices. This example displays the content of a bach.score object via translation to MaxScore.

4.3 conTimbre playback

conTimbre is a sound bank of orchestral instruments created by Thomas Hummel. With more than 80000 individual samples—many of them performed with extended techniques—it is a tool becoming increasingly popular amongst new music composers and electronic musicians alike[4]. Using copy protection and a proprietary file format, it requires dedicated software to play back these samples. However, Hummel has implemented a suite of OSC message for interaction with its Max-based ePlayer. We have thus created new abstractions for Max and Max for Live exploiting the power of the conTimbre library and enabling multi-timbral microtonal playback. The MaxScore.2conTimbre module, complementing the MaxScore.Sampler and MaxScore.Fluidsynth2 playback devices, reads ePlayer settings files, which fills the menus with (and sets them to) the current instrument names.

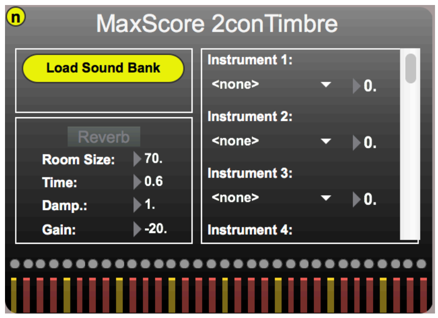


Figure 16. Screenshot of the MaxScore.2conTimbre module, a playback device for MaxScore.

5. NEW TOOLS FOR GUIDED IMPROVISATION AND SITUATED PERFORMANCES

5.1 Cursors

In 2006, Marlon Schumacher, then-member of the European Bridges Ensemble, asked Hajdu to implement cursors in the Quintet.net Client (a software for networked multimedia performance) for his piece *Fire* [14] which were to travel independently of each other at different speeds across its notation display to guide the performance of electronic musicians. In 2016, cursors were also added to the MaxScore Editor and more recently to MaxScore.toMira through the MaxScore.Cursors abstraction. The behavior of those cursors (a maximum of 20 per score) can be controlled with a variety of messages for which the Max @ attribute notation was adopted.

Each message contains the message name cursor, an instance number and any of the following @ attributes:

cursor 0 @begin 0 0 @end 0 1 @runs 1 @countdowncolor 1. 0. 0. 1. @countdown line @color 0. 0. 1. 0.7 @interval 2000 @ timestretch 2. @shape line

[4] https://www.contimbre.com

Depending on the @begin and @end attributes, the length of a cursor will be adjusted to the span of the specified staves. For this, the *getStaffBoundingInfo* query is performed to yield the bounding rectangle around a particular measure/staff (our term for the cross section of a measure and a staff, for lack of a better term). In addition, tempo and time signature are queried to determine the speed of the cursor travelling across the canvas.

Furthermore, cursors can also be controlled with the start, stop, resume, blink, unblink und hide messages. Instance number -1 can be used if all cursors are to be affected at once.

There is a difference between rendering cursors in MaxScore and Mira: While in MaxScore cursors are rendered just like any other graphics elements, they are represented by the actions of GUI objects in Mira (such as the visible line of a transparent multislider). This way, the network load can be decreased dramatically as only control messages need to be sent to the clients to adjust the position and size a multislider and move its line.

In 2016, one of us (Hajdu) participated in an academic exchange with Cat Hope, Lindsay Vickery and other members of the Decibel Ensemble (all at WAAPA, Edith Cowan University, Perth, at the time). The aim was to mutually expose ourselves to the developments of the other group [20]. A concert was organized at the end of our first stay. For this, the piece "Carnage" was written as a guided improvisation for the Decibel ensemble (flute, bass clarinet, viola, cello, percussion) and premiered on July 29, 2016. The piece (based on the eponymous film by Roman Polanski and Yasmina Reza) featured five lines of emoticons. The musicians were instructed to interpret the moods represented by the emoticons and were guided by the movements of the cursors. The notation was read from a single projection of the MaxScore canvas.

In November of 2017, this piece was performed again in Tel Aviv by the Meitar ensemble (featuring flute, bass clarinet, violin, cello and piano). During this performance, the musicians read the music from individual iPads running the Mira app (Figure 17).

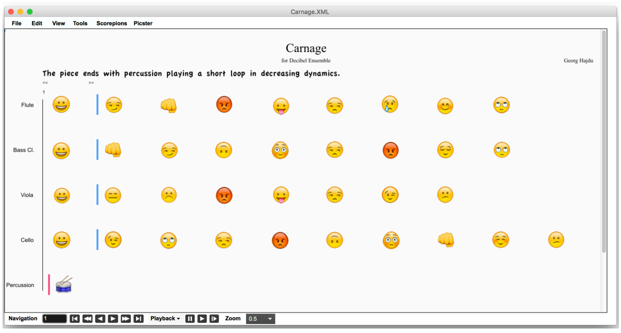


Figure 17. Hajdu's piece Carnage consists of a one-page score with individual cursors guiding the performance.

6. OUTLOOK AND CONCLUSIONS

MaxScore development is currently focusing on areas including network connectivity (MaxScore.NetCanvas and MaxScore.toMira), guided improvisation (MaxScore.Cursor) as well as compatibility with other software developments (bach and conTimbre). This has been facilitated by changes to the MaxScore object itself. A promising door has been opened by the introduction of Expressions which will allow users to pursue ideas akin to the Spatialization Symbolic Music Notation [16], which combines a language of icons with clearly defined spatial trajectories. For this, we will be working on a GUI accommodating a number of editors both in the graphical and control domains. It is also planned to use MaxScore and its peripheral components in a performance in the St. Pauli Elbtunnel in Hamburg—a 100-year old tunnel under the Elbe river, involving 144 musicians reading their music off portable devices in 2019. Until then, further strides will have to be done towards robustness and efficiency.

Acknowledgments

The authors would like to thank the Hamburg *Ministry for Science, Research and Equality* (BWFG) and the *Federal Ministry for Education and Research* (BMBF) for their support within their *Landesforschungsförderung* and *Innovative Hochschule* frameworks.

7. REFERENCES

[1] G. Hajdu, and N. Didkovsky, "MaxScore – Current State of the Art," in *Proceedings of the International Computer Music Conference,* Ljubljana, Slovenia, 2012, pp. 156–162.

[2] A. Agostini and D. Ghisi, "Bach: An Environment for Computer-Aided Composition in Max," in *Proceedings of the International Computer Music Conference*. Ljubljana, Slovenia, 2012, pp. 373–378.

[3] D. Fober, Y. Orlarey, S. Letz, "INScore – An Environment for the Design of Live Music Scores," in *Proceedings of the Linux Audio Conference*, CCRMA, Stanford University, 2012, pp. 47–54.

[4] Erich Gamma and Richard Helm, *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1994.

[5] D. Blostein and L. Haken, "Justification of Printed Music," *Communications of the ACM*, vol. 34, no. 3, pp. 88–99, 1991.

[6] G. Hajdu, "Dynamic Notation – A Solution to the Conundrum of Non-Standard Music Practice". *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015.

[7] C. Agon, G. Assayag and J. Bresson, *The OM Composer's Book Vol. 1*, Editions Delatour France / IRCAM, 2006.

[8] G. Hajdu, "Die Transkription des Poème Vers la flamme op. 72 von Alexander Skrjabin für Bohlen-Pierce-Ensemble," in A. Bense, M. Gieseking, B. Müßgens (Eds.), *Musik im Spektrum technologischer Entwicklungen und Neuer Medien. Festschrift für Bernd Enders*, Osnabrück: epOs, 2015, pp. 65-80.

[9] D. R. Tuohy and W. D. Potter, "Generating Guitar Tablature with LHF Notation Via DGA and ANN," in *Advances in Applied Artificial Intelligence. IEA/AIE 2006. Lecture Notes in Computer Science,* vol. 4031. Springer, pp. 244–253, 2006.

[10] N.-L. Müller, K. Orlandatou and G. Hajdu, "Starting Over – Chances Afforded by a New Scale," in M. Stahnke and S. Safari (Eds.), *1001 Microtones*, Neumünster: Von Bockel, 2014, pp. 127–172.

[11] G. Hajdu, "Resurrecting a Dinosaur – The Adaption of Clarence Barlow's Legacy Software AUTOBUSK," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, UK, 2016.

[12] J. Sello, "The Hexenkessel: A Hybrid Musical Instrument for Multimedia Performances," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Brisbane, Australia, 2016.

[13] B. Carey and G. Hajdu, "Netscore: An image server/client package for transmitting notated music to browser and virtual reality interfaces," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, UK, 2016.

[14] G. Hajdu and N. Didkovsky, "On the Evolution of Music Notation in Network Music Environments," *Contemporary Music Review,* vol. 28, nos. 4/5, 2009, pp. 395-407.

[15] S. James, C. Hope, L. Vickery, A. Wyatt, B. Carey, X. Fu, and G. Hajdu, "Establishing connectivity between the existing networked music notation packages Quintet.net, Decibel ScorePlayer and MaxScore," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017.

[16] E. Ellberger, G. Toro Pérez, J. Schuett, G. Zoia, and L. Cavaliero, "Spatialization Symbolic Music Notation at ICST," in *Proceedings of the joint International Computer Music Conference / Sound and Music Computing conference ICMC|SMC|2014*, Athens, Greece, 2014, pp. 1120-25.

DADA: NON-STANDARD USER INTERFACES FOR COMPUTER-AIDED COMPOSITION IN MAX

Daniele Ghisi
STMS Lab (IRCAM, CNRS, UPMC)
Conservatory of Genoa
danieleghisi@bachproject.net

Carlos Agon
STMS Lab (IRCAM, CNRS, UPMC)
carlos.agon@ircam.fr

ABSTRACT

This article introduces the *dada* library, providing Max with the ability to organize, select and generate musical content via a set of graphical interfaces manifesting an interactive, explorative approach. Its modules address a range of scenarios, including, but not limited to, database visualization, score segmentation and analysis, concatenative synthesis, music generation via physical or geometrical modelling, wave terrain synthesis, graph exploration, cellular automata, swarm intelligence, and videogames. The library is open-source and extendable; similarly to *bach*, it fosters a performative approach to computer-aided composition (as opposed to traditional off-line techniques): the outcome of all its interfaces can be recorded in scores, or used in real time to drive, among other things, digital signal processes, score transformations, video treatments, or physical actuators.

1. INTRODUCTION

Real-time computer-aided composition is a relatively recent and promising field of study. In particular, the development of the *bach* library [1] for Max [2] has made possible to operate on symbolic scores scores as interactively as on sound buffers. Although *bach* features a certain number of interactive, graphical objects, all of them essentially implement established representations of music, be they traditional scores or alternative but widespread representations such as the clock diagram or the Tonnetz [3]. This is both a strength and a limitation: it is a strength, inasmuch as it allows *bach* to be a general-purpose, highly adaptable tool; it is a limitation, inasmuch as it limits the scope of *bach* as a toolbox for experimental, non-standard musical practices and research.

This article introduces a new library, *dada*, based on the *bach* public API, meant to fill this gap, focusing on real-time, non-standard graphical user interfaces for computer-aided composition. Hence, most of the modules in *dada* are interactive user interfaces; nonetheless the library also features a small number of non-UI modules designed to complement the operation of some of the interfaces in the library. The *dada* library is the third library in the "*bach* family" [4] (the *cage* library being the second [5]). It is part of the PhD thesis of one of the authors, and although it has been widely used in his recent musical production, for lack of space, this article will not describe such examples of usages (while the full PhD thesis does [6]).

2. MOTIVATION AND RATIONALE

The philosophy behind *dada* is profoundly different from the one which informed *bach*: *dada* is to *bach* what a laboratory is to a library. Under the umbrella of non-standard, strictly two-dimensional graphic user interfaces, all of its components participate of a ludic, explorative approach to music; most of its components also refer to the fields of plane geometry, physical modelling or recreational mathematics.

A preliminary alpha version of *dada* (0.1) is available on its official website [1] . The modules included in the *dada* library can be roughly divided into three categories: tools for corpus-based composition (including database interfaces and score analysis mechanisms), tools for physical or geometrical modelling of music (including gravity-based models, pinballs, kaleidoscopes and wave terrain synthesis), and tools to handle rule-based systems and games (including cellular automata, swarm intelligence models and platform videogames). Before providing, in the next few sections, a detailed overview of the modules, we would like to motivate our development choices.

Differently from *bach* and *cage*, *dada* is a personal library, tailored on the compositional needs of one of the authors. Essentially all implementation choices have been taken with this consideration in mind, a fact that is most notable in some specific modules (such as *dada.bodies* or *dada.music*∼). In other words, the choice of what to develop has not been influenced by the needs of the computer music community, but rather by a very personal effort to experiment with geometry-based musical ideas.

That being said, *dada* is by design an open box: it is open-source [2] , and we hope that other interested musicians and developers will contribute with new modules. Such additions will be facilitated by the *dada* API, implementing a set of common operations (to provide, among other things, support for graphic display, selection handling and undo mechanisms).

[1] http://www.bachproject.net/dada
[2] https://github.com/bachfamily/dada

It should be remarked that most processes in *dada* are not new—some of them have also been implemented and distributed as third-party Max externals. In particular: the portion of the library dealing with database visualization has been inspired by the CataRT library for concatenative synthesis [7] (which has two different Max implementations [8]); the swarm intelligence module relates to the Boids library [3]; the wave terrain synthesis module relates to WAVE [4] and to Stuart James's work [9]; and there is a large number of implementations of cellular automata in Max, including Bill Vorn's Life Tools [5]. There are, however, two good reasons for our choice to re-implement these tools inside *dada*.

Firstly, all *dada* modules follow the *bach* paradigm of real-time computer-aided composition [4]. The contribution of *dada* is hence novel, inasmuch as it builds on top of the rich hierarchical representation and algorithmic manipulation afforded by *bach* and *cage*, integrating its processes within a single, unified, coherent system. As an example, all *dada* modules are designed to be easily used in combination with *bach.ezmidiplay*, to obtain a quick MIDI rendering of the musical outcome, and with *bach.transcribe*, to record the outcome in proportional notation in a *bach.roll*.

Secondly, the *dada* implementation is more general, more customizable or has a different scope. As an example, the *dada.catart* module, a two-dimensional interfaces of datasets, differently from CataRT is not limited to audio datasets; on the contrary, it is able to organize on the cartesian plane entries of a generic SQLite database—and its focus is, most notably (but not uniquely), on score datasets, providing mechanisms to segment and analyze symbolic scores. As another example, the *dada.boids* object, differently from the Boids library, allows for customized rules to be set via snippets of C code, compiled on-the-fly. The same is true for *dada.life*, dealing with cellular automata.

3. TOOLS FOR CORPUS-BASED COMPOSITION

The tools in this category are primarily designed to handle scores databases, but can be more generally applied to the creation and visualization of general datasets. Some of the modules in this category were already introduced in [10], and have been, since then, improved and extended.

The overall system relies on four different modules: *dada.segment*, performing score segmentation and feature extraction; *dada.base*, implementing the actual database engine; *dada.catart* and *dada.distances*, two-dimensional graphic user interfaces capable of organizing and interacting with the extracted grains.

3.1 Segmentation

The *dada.segment* module performs the segmentation of a score, contained in a *bach.roll* (as proportionally notated musical data) or a *bach.score* (as classically notated musical data, see Figure 1), in one of the following manners: using the markers in the original score as slice points; defining an equation for the size of each grain; using labels assigned to notes and chords (outputting one grain for each label).

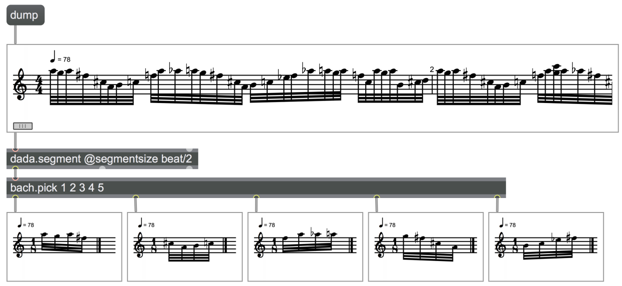


Figure 1. Segmentation of a *bach.score* into grains having length equal to half of the beat (i.e. an eighth note).

The segmentation can be carried on with overlapping windows, both on proportional and classically notated scores, and standard windowing techniques can be applied to MIDI velocities, if desired.

3.2 Analysis

Grain analysis is performed during the segmentation process. On one side, *dada.segment* is capable of adding some straightforward metadata to the segmented grains, such as their duration, onset, index, label (if segmentation is carried out via label families) and notation object type (either 'roll' for *bach.roll* or 'score' for *bach.score*); in case the grain comes from a *bach.score*, tempo, beat phase, symbolic duration and bar number can also be added.

On the other hand, *dada.segment* allows the definition of custom features via a loopback patching configuration named "lambda loop" [11]: grains to be analyzed are output one by one from the rightmost outlet, preceded by the custom feature name; the user should provide a subpatch to extract the requested feature, and then plug the result back into *dada.segment*'s rightmost inlet. Feature names, defined in an attribute, are hence empty skeletons which will be "filled" by the analysis implementation, via patching. This programming pattern is widely used throughout the *bach* library (one can compare the described mechanism, for instance, with *bach.constraints*'s way of implementing custom constraints [1]), and allows users to implement virtually any type of analysis on the incoming data.

Some ready-to-use abstractions are provided for quick prototyping, whose terminologies are mostly borrowed from the audio domain, even if they are applied to symbolic data; hence *dada.analysis.centroid* will output an average pitch, *dada.analysis.spread* will output the standard deviation of the pitches, and so on. The reason behind this choice is to underline the duality between this symbolic framework and the digital signal processing approach. Moreover, since analysis modules are standard Max patchers, it is easy for users to inspect and adapt them to different behaviors.

Analyzed features are collected for each grain, and output as metadata from the middle outlet of *dada.segment*.

[3] http://s373.net/code/
[4] http://www.noisemaker.academy/blog/
[5] http://billvorn.concordia.ca/research/software/lifetools.html

3.3 Database

Once the score grains have been produced and analyzed, they are stored in a SQLite database, whose engine is implemented by the *dada.base* object. Data coming from *dada.segment* are properly formatted and fed to *dada.base*, on which standard SQLite queries can be performed.

Some higher-level messages are provided to perform basic operation and to handle distance tables (i.e. tables containing distances between elements in another table, useful, for instance, in conjunction with the *dada.distances* module, as explained below).

Databases can be saved to disk and loaded from disk.

3.4 Interfaces

Two objects provide graphic interfaces for the database: *dada.catart* and *dada.distances*.

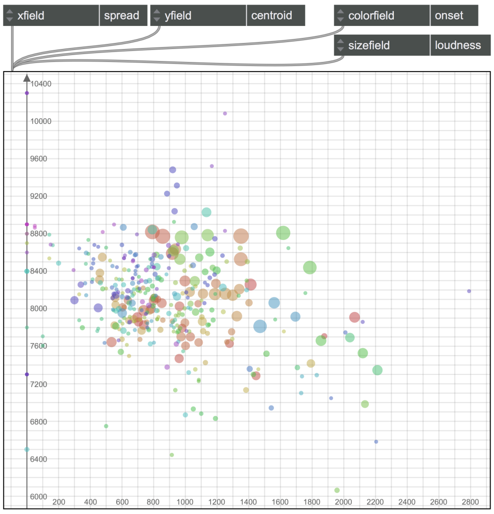


Figure 2. The *dada.catart* object displaying a database of score fragments. Each element of the database (grain) is represented by a circle. On the horizontal axis grains are sorted according to the spread, while on the vertical axis grains are organized according to their centroid. The colors scale is mapped on the grain onsets in the original file, while the circle size represents the grain loudness.

The *dada.catart* module provides a two-dimensional Cartesian graphic interface for the database content. Its name is an explicit acknowledgment to the piece of software which inspired it [7]. Grains are by default represented by small circles in a two dimensional plane. Two features can be assigned to the horizontal and vertical axis respectively; two more features can be mapped on the color and size of the circles. Finally, one additional integer valued feature can be mapped on the grain shape (circle, triangle, square, pentagon, and so forth), adding up to a total number of five features being displayable at once (see Figure 2).

The *dada.distances* module provides a distance-based representation of the database content. Points are the entries of a table, characterized via their mutual distances, contained in a different table. They are represented in a two-dimensional plane via the multidimensional scaling algorithm provided by [12]. Edges are drawn only if the corresponding distance is below a certain threshold (see Figure 3). The resulting graph is navigable in a Markov-chain fashion, where distances are interpreted as inverse probabilities. As for *dada.catart*, features can be mapped to colors, sizes and shapes.

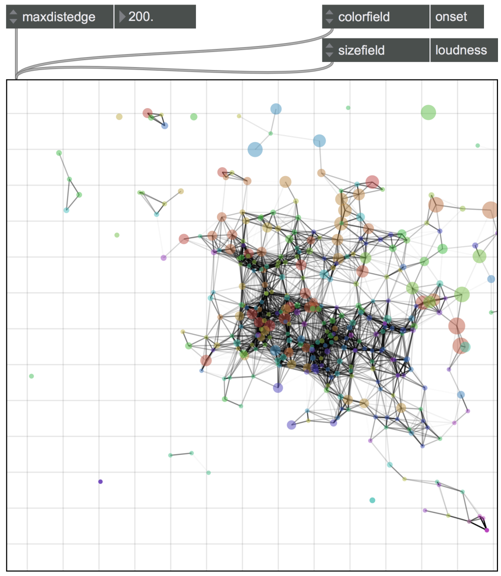


Figure 3. The *dada.distances* object displaying a database of score fragments. As for the *dada.catart* case (Figure 2), each element is represented by a circle. Grains are only positioned only according to a certain defined distance function (in this case, the distance of their centroids, spreads and loudnesses, as tridimensional vectors), the positioning in the plane is carried out via multidimensional scaling.

Both in *dada.catart* and in *dada.distances* each grain is associated with a "content" field, which is output either on mouse hovering or on mouse clicking. The content is usually assigned to the *bach* list representing the score. The sequencing can also be beat-synchronous, provided that a tempo and a beat phase fields are assigned: in this case the sequencing of each grain is postponed in order for it to align with the following beat, according to the current tempo (obtained from the previously played grains).

A *knn* message allows to retrieve the k nearest samples for any given (x, y) position. A system of messages inspired by turtle-graphics is also implemented, in order to be able to move programmatically across the grains: the *setturtle* message sets the turtle (displayed with an hexagon) on the nearest grain with respect to a given (x, y) position; then the *turtle* message moves the turtle of some $(\Delta x, \Delta y)$, choosing the nearest grain with respect to the new position (disregarding the original grain).

The database elements can be sieved by setting a *where* attribute, implementing a standard SQLite 'WHERE' clause. The vast majority of the display features can be customized, such as colors, text fonts, zoom and so on. In combination with standard patching techniques, these features also allow the real-time display, sequencing and recording of grains.

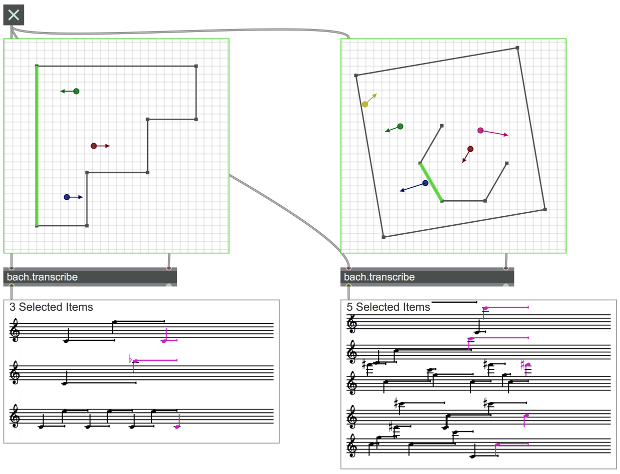


Figure 4. Two *dada.bounce* objects producing respectively a polyrhythm (left) and a more complex pattern (right). Each edge is mapped on a note which can be played or recorded as soon as the the edge is hit.

4. TOOLS FOR PHYSICAL OR GEOMETRICAL MODELLING OF MUSIC

The interfaces in this group share the idea that objects in space can lead to music generation by means of geometry and motion.

4.1 Pinball-like bouncing

The *dada.bounce* module suggests a pinball-like scenario, where a certain number of balls move inside a space delimited by a user defined graph, called "room". The ball movement is uniform (constant speed [6], no gravity), except when a ball bounces off an edge. Each edge contains metadata either as a couple of MIDI pitch and velocity, or as a complex score; such metadata will be output whenever a ball hits the corresponding edge. Information about the collision (identifying the point, the edge and the ball) can be retrieved. Ball and room properties and metadata can be changed dynamically.

Simple room configurations may lead to loops or polyrhythmic patterns; more complex results are achievable by modifying the geometry of the room and the number of balls (see Figure 4), or by using feedback loops as programming patterns—e.g., by adding edges at each hit.

4.2 Gravitation

A different paradigm is enforced by the *dada.bodies* module, modelling a two-dimensional universe with gravity, containing two types of objects: "stars", fixed circles, from which a certain number of radii stand out, each representing a note (see Figure 5); and "planets", which orbit around the stars according to a customizable gravitational law, triggering the playback of radial notes whenever they orbit "close enough" to a star. The MIDI velocities are scaled according to the distances between planets and stars. As

[6] In order to avoid confusion with MIDI velocities, the term "speed" is used in this context also to refer to the velocity vector, and not just to its scalar intensity.

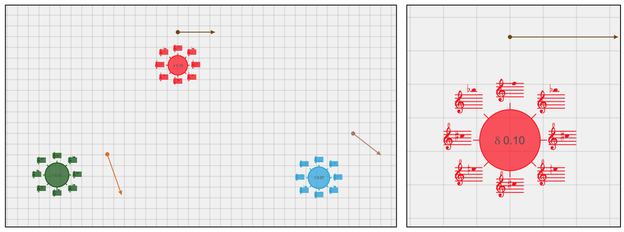


Figure 5. Configuration of *dada.bodies* gradually distorting the loops of Gerard Grisey's *Vortex temporum*. At right: a zoomed version of one of the stars (corresponding to the flute's notes).

a metaphor, one could imagine "stars" as being "radial aeolian harps", played by the planets whenever they circle around them.

This model is a convenient representation to handle continuous modification of loops. In a situation with a single star and a single planet, one could set the distances and speeds so that the planet motion around the star is circularly uniform (convenience methods are provided), resulting in a perfectly looping pattern. Modifying the planet position or speed, ever so slightly, results in a time warping operation on the loop. Adding more stars will trigger complex scenarios. Chaotic loops and attractor-like situations can be achieved via this system.

4.3 Kaleidoscopes

The *dada.kaleido* module traces the disposition and movement of a certain number of polygons in a kaleidoscope-like container. A certain number of shapes (polygon or ellipses) are positioned inside a 2- or 3-mirror chamber. The 2-mirror chamber has a couple of mirrors of equal length hinged at the origin, producing circular "snowflake"-like patterns. The angle between the mirrors is set by the user via the *count* attribute, an integer number $n \geq 2$ relating to the mirror angle α in the following way: $\alpha = \pi/n$: for $n = 2$ mirrors are at right angles, for $n = 3$ they are two sides of an equilateral triangle, and so on (see Figure 6). For $n = 2$ and $n = 3$, a third mirror can be introduced [13, p. 210], closing the triangle formed by the other two, hence extending the tiling to the whole plane.

The shapes inside the chamber can be modified either via the interface or via a set of messages, such as 'move', 'rotate', 'scale' and 'shake'. A combination of rotation with a certain amount of shaking will result in an elementary yet effective modelling of a hand rotating the body of a kaleidoscope.

Users can assign test points on the plane, so that the object may report whenever any of the polygons, during a movement, hits a point (i.e. when the point enters a polygon or any of its kaleidoscopic reflections) or releases a point (i.e. when the point is no longer on the polygon, or on any of its kaleidoscopic reflections). Information about the distances between test points and polygons can also be retrieved, and can be used as control for symbolic or DSP processes. As an example of application, one might associate each shape with a portion of audio file, which, like a

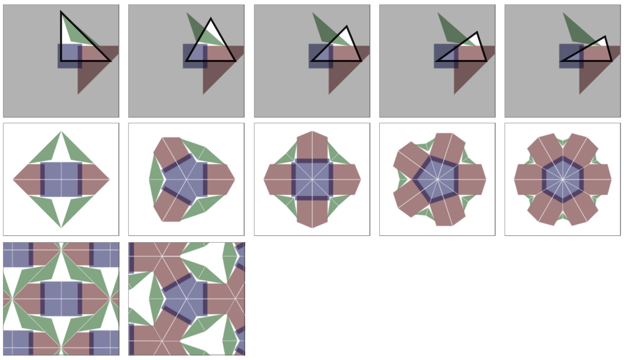


Figure 6. Same shapes reflected into different chambers of a *dada.kaleido* object, for increasing values of the *count* attributes. Last row shows the 3-mirror version of the patterns, only available for $n = 2$ and $n = 3$.

vinyl, is only read, with variable speed, when a certain test point (the "stylus") is positioned over the shape.

4.4 Wave terrain synthesis

The *dada.terrain∼* module implements wave terrain synthesis [14, pp. 163–167]: a function $z = f(x, y)$ yields the "height" of the terrain for each point of a plane. Evaluating the function f on a specific path $p : x = x(t), y = y(t)$ produces a one-dimensional function $z = g(t) = f \circ p(t)$, which represents the wave terrain synthesis along the path p. Wave terrain synthesis essentially constitutes an extension of the ordinary wavetable synthesis to bidimensional lookup tables, and it is traditionally implemented in this way, in order to lower computational costs. A typical scenario is when the surface f is a direct product of sinusoids, such as $f(x, y) = sin(n\pi x)cos(m\pi y)$: in this case, by sampling the terrain on circular or elliptic orbits p, one obtains FM-like timbres.

In the *dada.terrain∼* module, the function $f(x, y)$ is however not defined via a wave table, and is set via an explicit portion of C code compiled on-the-fly (see Figure 7). The wave terrain is displayed so that black corresponds to $z = -1$, white corresponds to $z = 1$, and 50% grey corresponds to $z = 0$.

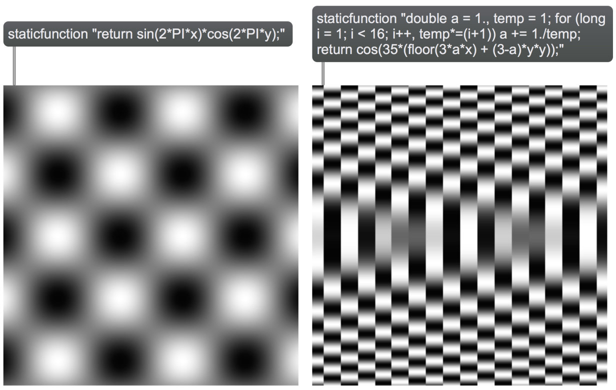


Figure 7. Two wave terrains displayed in *dada.terrain∼*.

Four auxiliary modules help producing specific paths, namely: segments, rectangles, ellipses and spirals; such modules produce coordinates at sample rate, to be used as input for the wave terrain module.

The *dada.terrain∼* module also supports the a "buffer wheel" mode, where the terrain is the result of a morphing between radially arranged buffers. Such morphing could be additive (result being a simple crossfade) or multiplicative; the equation for the contribution of each buffer can be set as a portion of C code compiled on-the-fly. As an example, consider Figure 8, where four instruments playing the same notes are arranged radially, and a spiral path samples the wave terrain, yielding a morphing between the four sounds.

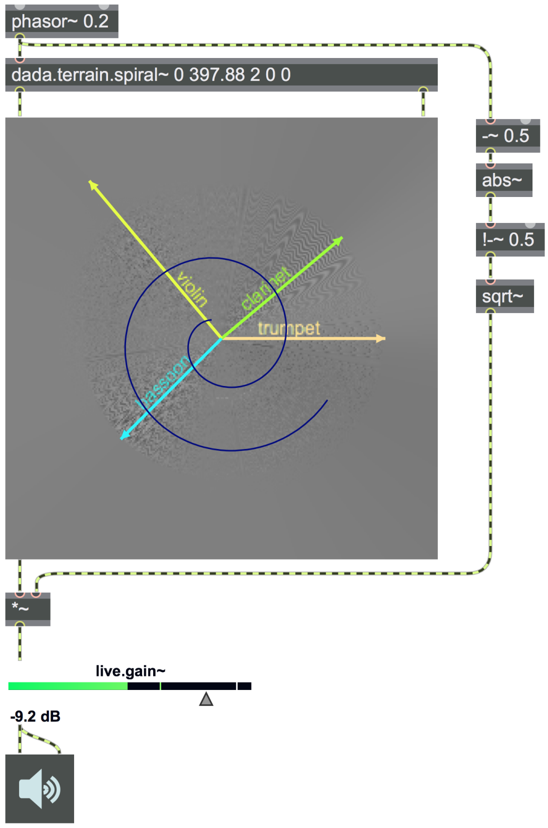


Figure 8. Four buffers, each containing an instrument playing an A3 in pianissimo, are arranged radially on a *dada.terrain∼*. The terrain is then sampled via a spiral path, yielding a morphing between the four sounds.

5. RULE-BASED SYSTEMS, GRAPHS, AND MUSIC AS A GAME

A certain number of tools explore the relationship between music, mathematics and games, and how this relationship ramifies towards combinatorics, algebra, topology and computer science (the link between canonical processes and topology being of course well known [15], further interesting examples can be found in tools such as origami [16] or juggling patterns [17] [7]).

[7] See for instance Tom Johnson: *Three notes for 3 jugglers* (2012).

The modules in this family share two important ideas. The first one is that interesting emergent behaviors may arise from dynamical systems even when their agents adhere to sets of extremely simple rules; this is well known, for instance, in the study of cellular automata, swarm intelligence and in Chaos Theory. The second idea is that digital scores may somehow be harbringers of a form of "gamification", i.e. the usage of game design elements in non-game scenarios.

After all, there are fundamental similarities between musical scores (in any form) and digital games [18]. Playing videogames often resolves in following a (graphically notated) rhythmical score, not dissimilar to a percussionist playing his or her own part in an orchestra: in both cases, the ability to stay within an acceptable level of precision affects the outcome. If the score is hard-coded, gamers can progressively learn the precise timing for their actions; if the score is open, gamers are obliged to play *a prima vista*.

5.1 Cellular automata

The first module in this family is *dada.life*: a graphical interface for two-dimensional cellular automata, on square or triangular grids. Cellular automata are rule-based systems, consisting of a regular grid of cells, each in one of a finite number of states (such as "alive" and "dead"). A set of cells called "neighborhood" is defined relative to each specific cell. Given a configuration of states, a new generation can be created according to a given rule, usually a mathematical function, determining the new state of a cell depending on the current states of the cells in its neighborhood. The most famous cellular automaton is arguably Conway's *Game of Life*. Extremely complex patterns can arise in cellular automata, even from simple rules.

A Max module handling two-dimensional cellular automata was already included in *cage* [5]; nevertheless the *dada.life* object improves the approach, by making it interactive, more customizable and faster. The customization possibilities are not limited to colors and sizes: rules themselves can be defined either via attribute combinations (for simple scenarios similar to Conway's *Game of Life*) or via a portion of C code, compiled on-the-fly—a more agile approach than *cage.life*'s Max patchers. [8]

Automata in *dada.life* can live on square or triangular lattices, such as the Tonnetz [3]. One can use the Tonnetz grid as basis for a two-states cellular automaton (see Figure 9): cells can be 'on' (playing) or 'off' (silent). Pattern hence result in musical sequences; for instance, oscillators (patterns that repeat after a finite number of steps) yield harmonic or melodic loops.

5.2 Swarm intelligence

The *dada.boids* module investigates swarm intelligence models. The object contains a certain number of "swarms" or "flocks", each containing a certain number of "birds" or "particles", singularly represented on the screen as points or arrows. The movement of each particle is dictated by a

[8] On the other hand, the fact that *cage.life* is an abstraction is consistent with the design of the whole *cage* project.

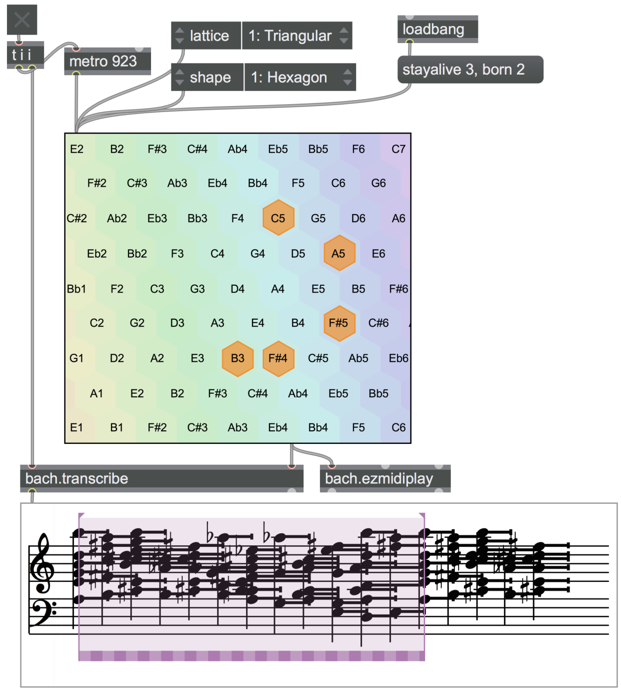


Figure 9. The harmonic cycle for the third movement in *Come un lasciapassare*, by one of the authors, as an oscillator of a two-dimensional cellular automata played on the Tonnetz.

sequence of higher-level rules, usually in the form of differential equations, accounting for the global behavior of the flock. Particles are traditionally called "boids" [19], a shortened version of "bird-oid objects".

In the traditional boids scenario, three rules apply: separation (particles steer to avoid crowding local flockmates), alignment (particles steer towards the average heading of local flockmates) and cohesion (particles steer to move toward the average position of local flockmates). The *dada.boids* module is able to account for such rules, as well as a for the presence of external barriers (obstacle avoidance) and winds. Moreover, each user can define his or her own set of rules, by compiling on-the-fly a portion of C code. Rules can have parameters, defining their position (such as the location of an obstacle), their orientation (such as the wind direction), their intensity (such as the wind speed, or the strength of a barrier), or, more generally, their behavior (such as a threshold for particle separation). Some of these parameters can also be associated to editable graphical user interface elements, such as points, vectors or lines—for instance, users can modify the direction of the wind by dragging the tip of the corresponding arrow, or the position of a barrier by dragging the corresponding horizontal or vertical line (see Figure 10).

In addition to their position and speed, particles can have a scalar intensity value, and custom rules can be set to modify intensities along with speeds. In practice, both built-in and user-defined rules are compiled functions that, for each particle, take as input its state, together with the state of the entire flock (coordinates, speeds and intensities of each particle), and yield as output, according to the current

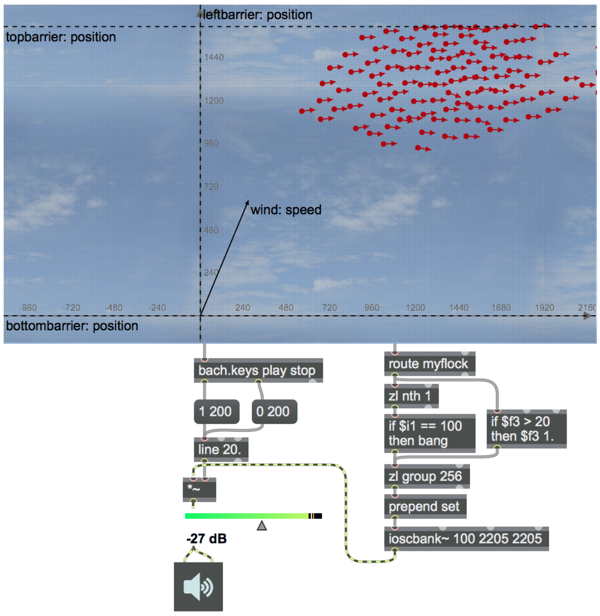


Figure 10. A *dada.boids* object where the vertical position of each boid is mapped on the frequency of a sinusoidal oscillator.

value of their parameters, a speed vector, to be added to the current particle speed (a "steering" vector), and possibly a value to be added to its intensity. By summing the contributions of all rules, one gets the discrete derivative of the particle speed (and intensity).

5.3 Graphs

The *dada.graph* module (see Figure 11) is a simple graph interface and editor, also featuring two automatic node placement algorithms provided by the Boost library [20]: the Fruchterman-Reingold force-directed layout [21] and the Kamada-Kawai spring layout [22]. Similarly to *dada.distances*, the graph can be also navigated in a Markov-chain fashion, starting at a given point, and then choosing each following steps according to the edge probability distribution (weights) and to a desired memory length.

A variation on *dada.graph* is the *dada.machines* module (see Figure 12), essentially a graph where each node represents some "machine", i.e. a simple, prototypal operation to be performed on one or more inputs. By default these operations are elementary symbolic score transformations, such as transposition, retrogradation, circular shift, splitting, merging, and so on; user-defined operations are also supported. In a way, *dada.machines* represents a patch inside a patch, taking a score as input, processing it via the transformation graph, and outputting the result; however its spirit is more peculiar, and it was designed to be used with randomly generated graphs (the 'random' message produces graphs where the number of machines of each type matches a desired distribution). Via *dada.machines* one can apply a performative, exploratory paradigm to music, somehow reversing the functional and ergonomic relationship between algorithm and data.

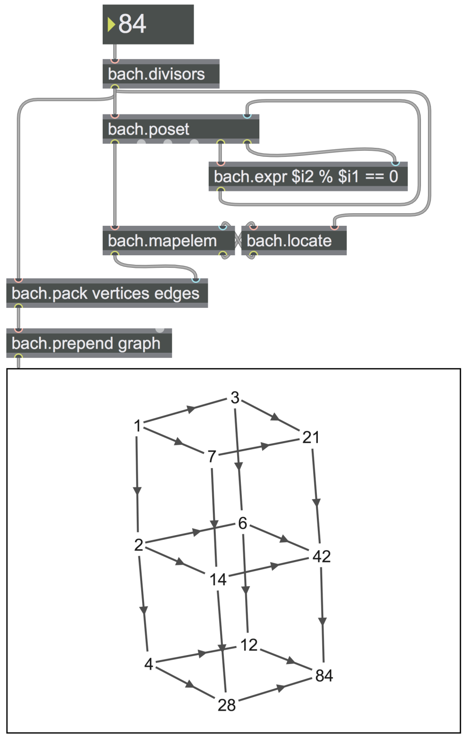


Figure 11. A simple patch displaying, via *dada.graph*, the lattice of divisors for an incoming natural number.

We are used to operate on data via carefully designed functions, and to modify them if the output result on a certain input is different from what we desire. As an example, to create a symbolic distorted granulation of a given Mozart sonata, one would spend quite some time designing the way the symbolic granulation should be achieved and the type of distorsion modelling needed. Nonetheless, one might reverse the principle, taking a random algorithm for granted, and carefully exploring input data in order to see if the results are interesting. If the algorithm is "complex enough", one might attempt to detect simple patterns (such as scales or counting-like patterns) along with more complex ones. (Of course, operatively, it makes little sense to search for a counting machine by tweaking inputs of a complex, random algorithm—which would categorize *dada.machines* module more as a mental experiment than a practical tool.)

5.4 Videogames

Developing a game engine in Max might seem awkward; and indeed there is a large number of environments specifically dedicated to the task (Unity probably being one of the most popular [9]). Max is neither designed nor optimized for such scenarios.

It can however be interesting to have a (crude, primitive) game engine natively coded in a Max external, since Max is a general purpose environment, and its visual paradigm can be applied to a large number of scenarios (digital audio, video, lighting, actuators...), making it easier to communicate between different media and techniques.

[9] `https://unity3d.com`

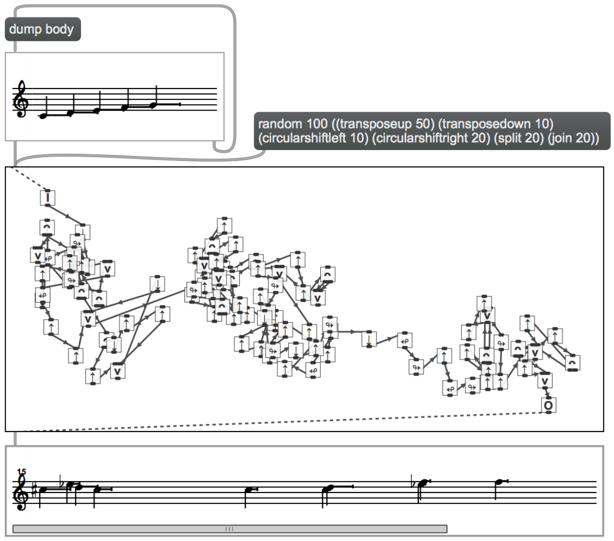


Figure 12. A patch featuring a *dada.machines* interface, generating network graphs containing 100 machines according to a distribution of some "atomic" score operations (transposition, circular shift, splitting and joining). The incoming score is processed via the randomly generated graph, and the result is output.

The *dada.platform* module, allowing the design of graphical interactions inspired by platform videogames, has been imagined and developed with these considerations in mind. Due to the complexity of designing a usable game engine, the module is currently in a prototypal phase, slightly more than a "proof of concept". Nevertheless *dada.platform* already supports four categories of objects:

Blocks: fixed objects which can possibly be broken;

Coins: fixed objects which can possibly be taken;

Game characters: moving elements which can interact with any other element in a more complex way. Game characters' motion is governed by a crude physical modelling: characters may possess the ability to jump, run, fly, swim, fire, glide, break elements, kill other characters, be killed by other characters. Game characters, in turns, belong to one of the following categories: 'usercontrol' (currently at most one character can be controlled by the user, also called 'hero'); 'idle' (do-nothing characters); 'food' (characters feeding the hero); 'enemy' (characters with the ability to harm or kill the hero); 'bullet' (projectiles potentially killing the hero);

Portals: objects which can dislocate the 'hero' to a new position in the same level, or to a brand new level.

All the properties of each object (such as its position, dimension, speed, abilities, image or sequence of images used to display it, and so on) can be set or fine-tuned via a dedicated inspector (see, for instance, Figure 13).

Linking game actions to musical events can be done in two ways. On one side, some of the objects' properties are musical scores (in *bach.roll* or *bach.score* syntax), output from a dedicated outlet whenever coins are taken, blocks are broke, and so on. More powerfully, any user action and any game interaction is notified via a dedicated outlet, so that any musical process can be triggered from them, such as sound synthesis, score production, video generation, and so on.

As it is not infrequent for objects in each level to share the same properties (just like identical blocks, coins or enemies), prototypes can be created, in order to easily handle multiple instances of indistinguishable objects.

Some of the properties of an object can be sequences of instructions, wrapped in levels of parentheses, written in a dedicated scripting language, designed to modify the configuration of the object itself, or of other objects. Instruction sequences are provided whenever a character dies, a block is hit, or a portal is entered, and so on. Script commands allows a wide range of actions, including: breaking blocks, assigning points or victory points, generating new objects, adding or removing abilities to characters, changing the state of objects, notifying some action, changing level or position in the level, pausing the game, preventing the hero from dying, winning, losing ("game over").

As a simple example, the script

```
(add hero ability fly)
(goto level mynewlevel.txt at PipeRev
with (keephero 1)),
```

assigned to a given portal, provides the current hero with the ability to fly, and then loads the level contained in the file `mynewlevel.txt`, at the position of the portal named `PipeRev`, keeping the current hero state (including its properties, points and victory points).

Each game character has a script sequence for its death (the "death sequence"); as another example, among many others, if one needs to turn a character named 'Juan', whenever he eats a certain fruit, into a character named 'SuperJuan', who, in turns, when killed returns to be a simple 'Juan' (like for the Mario/SuperMario classic Nintendo duality), one might want to assign to the fruit a death sequence along these lines:

```
(add hero ability break)
(change hero (name SuperJuan)
(idlesprite superjuanidle)
(walksprite superjuanwalk)
(jumpsprite superjuanjump)
(flysprite superjuanswim)
(height 1.625)
(ext 0.35 0.35 0.825 0.825)
(deathseq (dontdie) (remove hero ability
die during 2000) (change hero (name
Juan) (idlesprite juanidle) (walksprite
juanwalk) (jumpsprite juanjump) (height
1) (ext 0.4 0.4 0.5 0.5) (deathseq))
(remove hero ability break))).
```

Specific information about keywords and syntax can be found in the *dada.platform*'s help file and reference sheet. I shall just underline, in particular, how the last example is based on the fact that the fruit's death sequence changes the hero's death sequence, which in turns contain an instruction to clear its own death sequence, when triggered.

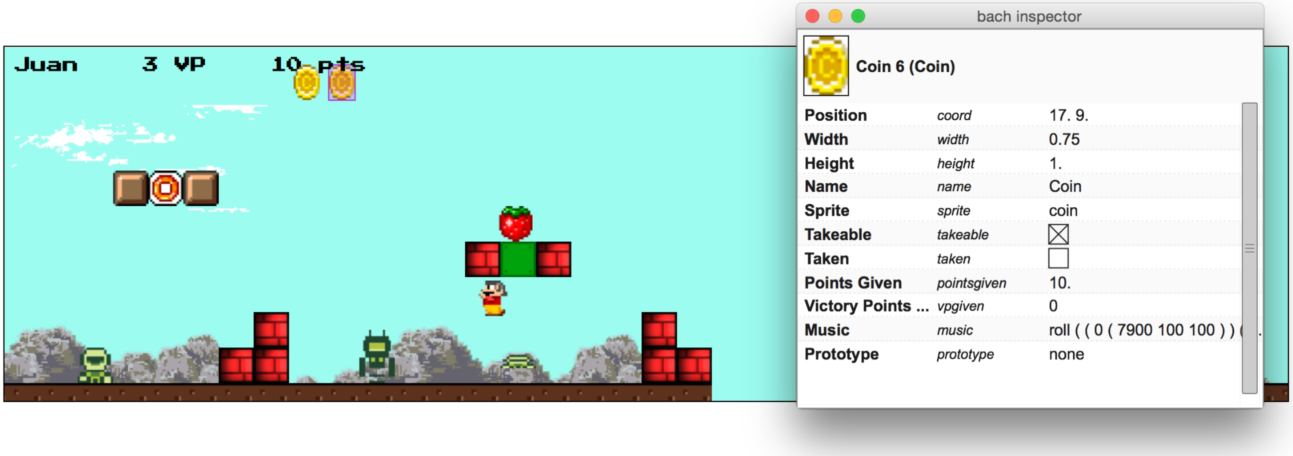


Figure 13. A screenshot of a *dada.platform* editor; the properties of the selected coin are displayed in the inspector.

6. COMPARISON WITH OTHER SOFTWARE

We have already emphasized the relationship of *dada* with project such CataRT, WAVE, Stuart James's objects, the Boids library, and Bill Vorn's Life Tools.

There is some correspondence between *dada*'s geometric approach and graphical sequencers such as Iannix [23] (as a matter of fact, a partial, two-dimensional porting of Iannix into *dada* might be a good addition to the library). On the other hand, the sequencing capabilities of Iannix largely outperform *dada*'s, whose purpose is not sequencing *per se*, but rather a seamless integration with the *bach* and Max environment, allowing, among many other things, live recording of scores.

The *dada* library shares with InScore [24] the interest in designing interactive non-standard symbolic representation. The idea of using games to interactively structure musical content resonates with Paul Turowski's researches and works, such as *Frontier* [18]. The *dada.life* module shares with Louis Bigo's HexaChord [25] the possibility of visualizing trajectories on musical lattices such as the Tonnetz—although the former focuses on the generation of cellular automata, while the latter is tailored for analysis purposes.

One should also remark the relationship of *dada* with music applications such as Björk's Biophilia, or Brian Eno's generative apps, or with interactive web tools such as some of the *Chrome Experiments* [10] or of the *A.I. Experiments* [11] (e.g., *The Infinite Drum Machine*); all these cases share with *dada* an interest for a tight, creative connection between visuals, gestures and music, and for exploring the grey area between interfaces and musical objects—however, if at least in Björk's case the musical apps are themselves art objects, *dada* modules are designed as simple instruments for composition [12].

[10] `https://www.chromeexperiments.com/`
[11] `https://aiexperiments.withgoogle.com/drum-machine`
[12] This has possibly one notable exception: the *dada.music~* module, included in the library, organizing and representing on a segment *all* music tracks, might be considered both as a conceptual work and as a piece of evidence for an exploratory approach to music.

7. FUTURE WORK

The *dada* library is still in its infancy, and a certain number of additions and improvements are needed to complete it and to make it more usable.

First of all, thorough testing and optimization are necessary to make the library more stable and the user experience more comfortable. Besides, a Windows porting is also needed (currently the library only works on MacOS).

One of the most important lines of development would be porting the interfaces on mobile operative systems (tablets, smartphones), where they might take advantage of multitouch support. The most convenient way would be to exploit the Miraweb package [13], developed by Cycling '74, which allows mirroring on web browsers specific interface elements contained in a patch; the possibility to add Miraweb support to third party externals should be explored.

As far as the documentation is concerned, comprehensive help files and complete reference sheets are already provided for each module. However, some video tutorials would be a valuable addition for users who need to get used to the *dada* environment.

The set of tools for corpus-based composition can be improved in a number of ways.

- The number of analysis modules should be increased, by attempting to bring into the symbolic domain important audio descriptors such as roughness, inharmonicity, temporal centroid, and so on.
- *dada.catart* and *dada.distances* should be provided with the capability to modify column values by dragging points on the display.

The tools for physical or geometrical modelling are probably the modules in *dada* whose development is most advanced; nonetheless the *dada.terrain~* module should be provided with anti-aliasing capabilities.

Finally, a certain number of improvements can affect the subset of tools dealing with rule-based systems and graphs:

- *dada.graph* is already capable of displaying graphs where the vertices are notes; it might also be pro-

[13] `https://cycling74.com/articles/content-you-need-miraweb`

vided with the possibility of displaying vertices as complex scores, which would open the way for potentially interesting applications.
- The *dada.graph* object should compute minimum spanning trees and shortest paths. It also should be provided with dedicated algorithms for special classes of graphs (such as trees or partially ordered sets). Automatic graph type detection, triggering the corresponding placement algorithm, might be a nice feature to have.
- The *dada.platform* object is currently little more than a "proof of concept". It would be interesting to issue something akin to a "call for scores" for pieces of interactive music based on it; this would probably also help detecting the bugs and the flaws of the system.

8. REFERENCES

[1] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015.

[2] M. Puckette, "Max at Seventeen," *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002.

[3] R. Cohn, "Introduction to neo-riemannian theory: a survey and a historical perspective," *Journal of Music Theory*, vol. 42, no. 2, pp. 167–180, 1998.

[4] D. Ghisi and A. Agostini, "Extending bach: A Family of Libraries for Real-time Computer-assisted Composition in Max," *Journal of New Music Research*, vol. 46, no. 1, pp. 34–53, 2017.

[5] A. Agostini, E. Daubresse, and D. Ghisi, "*cage*: a High-Level Library for Real-Time Computer-Aided Composition," in *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014.

[6] D. Ghisi, "Music Across Music: Towards a Corpus-Based, Interactive Computer-Aided Composition," Ph.D. dissertation, Université Pierre et Marie Curie / Sorbonne Université, IRCAM, France, 2017.

[7] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *Proceedings of the International Conference on Digital Audio Effects*, Montreal, Canada, 2006.

[8] D. Schwartz, "Interacting with a Corpus of Sounds," in *Symposium on Sound and Interactivity*, Singapore, 2013.

[9] S. G. James, "Developing a flexible and expressive realtime polyphonic wave terrain synthesis instrument based on a visual and multidimensional methodology," Ph.D. dissertation, Edith Cowan University, Australia, 2005.

[10] D. Ghisi and C. Agon, "Real-time corpus-based concatenative synthesis for symbolic notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, United Kingdom, 2016.

[11] A. Einbond, C. Trapani, A. Agostini, D. Ghisi, and D. Schwarz, "Fine-tuned Control of Concatenative Synthesis with CataRT Using the bach Library for Max," in *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014.

[12] Q. Wang and K. L. Boyer, "Feature learning by multidimensional scaling and its applications in object recognition," in *SIBGRAPI Conference on Graphics, Patterns and Images*, Arequipa, Peru, 2013.

[13] D. Gay, *Geometry by Discovery*. Wiley, 1997.

[14] C. Roads, *The Computer Music Tutorial*. MIT press, 1996.

[15] D. R. Hofstadter, *Gödel, Escher, Bach*. Basic Books, 1999.

[16] E. Andersen, "Origami and math," http://www.paperfolding.com/math/, 2012, accessed: 2015-13-12.

[17] M. Macauley, "Braids and juggling patterns," Harvey Mudd College thesis, USA, 2003.

[18] P. Turowski, "Digital game as musical notation," Ph.D. dissertation, University of Virginia, USA, 2016.

[19] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[20] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *Art and Agency: An Anthropological Theory*. Addison-Wesley Professional, 2001.

[21] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and Experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[22] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, no. 1, pp. 7–15, 1989.

[23] T. Coduys and G. Ferry, "Iannix-aesthetical/symbolic visualisations for hypermedia composition," in *Proceedings of the Sound and Music Computing Conference*, Paris, France, 2004.

[24] Y. Fober, Y. Orlarey, and S. Letz, "An Environment for the Design of Live Music Scores," in *Proceedings of the Linux Audio Conference*, CCRMA/Stanford University, USA, 2012.

[25] L. Bigo, D. Ghisi, S. Antoine, and A. Moreno, "Representation of musical structures and processes in simplicial chord spaces," *Computer Music Journal*, vol. 39, no. 3, pp. 11–27, 2015.

SPACE NOTATION IN ELECTROACOUSTIC MUSIC: FROM GESTURES TO SIGNS

Bertrand Merlier
Université Lumière Lyon 2
Département Musique & Musicologie
Lyon, France
bertrand.merlier@univ-lyon2.fr

ABSTRACT

This article is based on an analysis of the functionalities of many devices and software used for sound spatialization, an original research about space perception modes and finally an in-depth study about musical notation systems. Theses studies lead the author to propose a notation system for spatialization activities, simply based on the paradigm of our Western classical music notation. Various examples illustrate the merits and versatility of this proposal. The present notation is both descriptive and prescriptive. Thus, a practical implementation based on MIDI standard also makes possible instrumental space performances, implementation of algorithmic processes, space writing and structuring, but also offers access to all the existing software such as MIDI sequencer, MIDI computing and score writing. The *MIDISpat* plug-in – developed by the author – has been used for many years inside of Reaper digital audio sequencer.

1. INTRODUCTION

The present paper describes a space notation system based on the paradigm of our Western classical music score notation.

This surprising proposal rests on:

- 20 years of practice and research (see bibliography at the end of this paper);
- an in-depth analysis of various space practices, various working strategies, including an analysis of quite all software available on the market (see Section 2.1);
- listening tests that have highlighted new space perception criteria (confirming the lack of knowledge of space phenomena by developers of most digital audio software) (see Section 2.2);

First (Section 2: Former Observations), we will show that current spatialization tools – as powerful as they seem to be – work on a graphical continuous representation of reality (generally gestural reality), such raw data being unrelated to any concept of notation.

Second (Sections 3 and 4), our reflection will focus on a brief study of Western notation, allowing us to identify 9 elementary principles for a possible notation of spatialization.

Thirdly (Sections 5 to 6), we will propose a theoretical, graphic and practical implementation of this space notation. Several examples illustrate our purpose and attest the validity of our proposal. A simple but efficient MIDI implementation (Section 7) has been used by the author since the mid-2000s in many circumstances such as:

- spatial interpretation of acousmatic works,
- multiphonic studio composition (from 5.1 surround to 16 channels),
- live electronic or mixed music.

2. FORMER OBSERVATIONS

2.1 Inadequate Space Software

In a paper entitled “Reflections on electroacoustic music spatialization in digital audio software” [1], I presented a large panel of software allowing sound spatialization. It summarized various working strategies, various modes of coding and representing space information.

Each studied DAW (Digital Audio Workstation) software provides space information representations build with one or more curves networks. These “continuous streams of data” are physical representations of gesture reality, just like an oscilloscope screen shows electrical signal variations over time. **These representations of space information are remote from any notation system**: they are either imaginary trajectories, or hardware dependent technical curves, or gesture movements directly issued from spatialization gestures that have been practiced for several decades and dependent on gestural organs such as: mixing desk potentiometer, joystick and computer mouse (see examples at Figure 2).

The computer mouse – and consequently the joystick – is the most rustic and reductive organ that can be imagined (especially to control the spatialization!): only XY position detection, no velocity, no energy or speed detection, no polyphony...

This “curvy” mode of representation is similar to tablature notation (look at similarities between Figures 1, 2, and 3), dominated by technical aspects related to instrumental gesture and specifically designed for each "space instrument". It is not at all universal, nor endowed with the abstraction required for a real notation.

In 2005, we concluded that almost all DAW software were inadequate: lack of readability, lack of graphic or intellectual abstraction, difficulties in editing space curves, impossibility or difficulty of simultaneous display of both audio signal and spatialization signal on the same time scale, limitation of virtuosity, impossibility to manage space polyphony or to work on spatial masses because of XY driven sound trajectories, impossibility to work in 3 dimensions... Our paper ended with a set of suggestions for the future:

- liberate space from any hardware contingencies (i.e. related to "instrument" or hardware);
- liberate space from any causality contingencies[1] (i.e. related to gesture);
- build a description system of the produced effect.
- consider a functional approach[2];
- replace continuous curves[3] by abstract objects such as "space event" or "space phrase";
- adopt a common gateway to exchange informa-tion between all the existing software[4].

Figure 1. Byzantine religious notation.[5] In the 11th-12th centuries, the first Gregorian or neumatic[6] notations coded small melodic and rhythmic cells. That is to say the melodic (or rhythmic) movements. It seems that space notation is more or less at this stage[7].

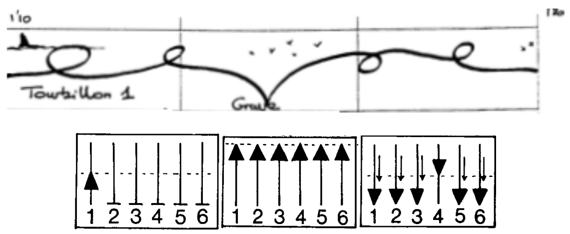


Figure 2. Two examples of spatialization representation. Top: representation of a space trajectory (*in* B. Merlier, *Nebuleuse M42* for cello and tape, 1993). Bottom: spatialization gestures at the mixing desk (*in* P. Boulez, *Dialogue de l'ombre double*, 1985).

2.2 Space Perception Modes

Our paper entitled "Space perception vocabulary in electroacoustic music composed or spatialised in pentaphony" – both presented in French at EMS'08 [1] and in English at SMC'08 [2] – aimed at clarifying or elaborating a vocabulary (a set of specialized words) likely to describe space perception in electroacoustic (multiphonic) music. A battery of tests have made it possible to highlight a collection of words describing spatial listening.

The results suggest five types of spatiality (see Table 1 on left column), 2 types of mobility, 4 or 5 families of adjectives to describe or characterize spatiality or mobility.

None of the studied commercial software (in Section 2.1) is able to seriously generate half of these 5 situations and criteria, proving one more time their inadequacy as regards space.

① sound bath	
② space image	
③ sound plan	localization geometry distance internal agitation movement
④ point	
⑤ demixing or counterpoint	

Table 1. Five space perception modes and associated criteria.

2.3 Conclusion

Only the sound actually produced by **one** loudspeaker is independent from gesture or information coding, and from hardware considerations (such as spatialization techniques and activities, as well as loudspeakers number and position).

If we want to progress into space control domain, if we want to be able to elaborate a real space discourse, to write it, to reread it, to understand it, we have to give up on this representation of reality by a network of curves, in favor of a simpler and more abstract representation.

[1] "Current spatialization representations are not efficient because they are linked to the description of the gesture that produces the effect, that is to say: causality. [...] The sound actually produced by a loudspeaker is independent from gesture or information coding, because the same perception can result from different causes. Even more, it is independent from hardware and – in particular – the number and position of loudspeakers."

[2] A functional approach makes it possible to envisage "compositional transformations on curves: symmetries, rotations, proportions modifications, homotheties, interpolations, smoothing, time offsets, time inversion, acceleration, control of trajectory speed..."

[3] "Instead of using continuous curves driving space without interruption from beginning to end of time, the notion of space object would make it possible to name, identify, record, memorize, duplicate, manipulate... space events."

[4] Only widely spread standard exchange vectors (such as MIDI standard, OSC protocol, OMF (Open Media Framework) files, SDIF (Sound Description Interchange Format) files [4]...) would allow communication between applications, between researchers, developers, composers... [5] As long as everyone remains in singular and idiosyncratic space practices, there will be no hope of having access to a somewhat universal notation; so no hope of seriously progressing.

[5] Courtesy of https://www.pinterest.fr/effiekondopoulo/byzantine-music/.

[6] *neume*: from the Greek νευμα that means gesture!

[7] It is interesting to reread music history ([6] or [7]) and to note numerous similarities between western notation apparition in the Middle Ages and current research on space: various experiments, quarrels of methods, misunderstandings between composers, performers and musicologists. So much so that one can easily imagine such a contemporary electroacoustic musician as the reincarnation of a twelfth century singer, another in the habit of a monk copyist and another in the role of a minstrel.

3. ABOUT WESTERN MUSIC NOTATION

Here are some brief historical and functional elements. Western music has pushed music notation sophistication far and wide. Even if in the twentieth or twenty-first centuries, many composers or aesthetic currents – including electroacoustic music – are cramped in these conventions when it comes to noting complexity, timbre or sound objects, even if diversions are frequent and necessary, musical notation remains today a fundamental tool, with its descriptive, prescriptive and memorial roles... (see for example [8], [9]or [1]). Table 2 presents and analyzes score key points.

Score graduations or discretization reduce musical complex reality to simple concepts (height, duration, intensity), that allow easy reading and writing (after learning the codes). This is one of the main reasons for score notation effectiveness (and success). This abstraction also allows building the artificial human complexity of our western music: polyphony, rhythms, sentences and finally all the subtle arrangements of melodies and harmonies... (see for example [6] or [7]).

paper support	fixation on a paper support constrains to find a two dimensions representation of the n musical parameters (height, duration, nuance, timbre, phrasing...). But, this constraint is also a guarantee of easy reading and reprography.
score offers:	- horizontally, a graduated time scale (tempo, measurements and pulsation); - vertically, a scale for graduated heights (tones and semitones).
note	this minimal musical event is a sign likely to graphically bear and express sound characteristics: height, duration.
	Other symbols – usually located around the note – indicate intensity or sound effects.

Table 2. Synthetic vision of Western music notation.

4. THEORETICAL BASIS FOR A SPACE NOTATION

Nine basic mandatory principles for the establishment of an efficient space notation are presented below. These proposals are based on an analysis of several notation systems in use, with the underlying idea that space may not be a bizarre or abnormal phenomenon and that it may not be necessary to invent a new scoring system.

a) Space should be written on paper like any other sound parameters (height, duration, intensity, timbre);

b) Its notation should be independent from any device (such as mouse, joystick, potentiometer, number and position of loudspeakers in space);

c) The proposed notation should be universal and adaptable to any of the following various circumstances, corresponding to the prescriptive, descriptive or memorial roles of score:
   - writing for non-real-time works in studio,
   - writing for real-time instrumental performances,
   - capture and notation of any "instrumental space performances",
   - retrospective reading for analysis purposes;

d) Like any other sound parameters, space needs to be noted as a discrete event[8], represented by a graphic symbol that can be drawn on paper and onto which characteristics can be assigned;

e) Apparent position of sound – as perceived by listeners – is due to a specific blend of sound level of *n* loudspeakers;

f) An elementary space event (ese) corresponds to the sound level on one loudspeaker at a given moment; This level can be zero, constant or variable;

g) An elementary space event (ese) has two main characteristics: intensity and duration, to which can be added certain effects such as attack, release, phrasing, distance, reverberation...

h) A space trajectory is a succession of space events (ese) arranged in time; A space trajectory can be thought of as a phrase or a space melody[9];

i) Simultaneous presence of the same sound on several loudspeakers can be considered and written as a space chord.

5. PRACTICAL AND GRAPHICAL IMPLEMENTATION

Taking the opposite way of all the practices in use, the author decided to rely on Western notation, i.e. get rid of any continuous curve and opt for discretization of space phenomena. We have previously justified our choice as being a trick intended to facilitate notation.

In concrete terms, our notation proposal is summarized in the following points:

- each line (or interline) of a staff corresponds to one loudspeaker[10];
- note faces are used as "space objects" describing each loudspeaker activity; thus, they own a duration, an intensity, several play modes or accentuations...;
- note and silence figures, tempo and measures indications have the same temporal meanings as in classical notation

[8] Which does not mean that space or spatialization are discrete phenomena! Discretization is only a simplification process, a view of mind.

[9] Melody: succession of musical sounds (*Dictionnaire des sciences de la musique*, Honneger, 1976, Bordas). The term "space melody" has already been used by various composers of electroacoustic music, i.e. Denis Dufour in the 1990s.

[10] A priori, at each user choice; this choice may well vary depending on hardware (number of loudspeakers, space layout...), depending on each work or each type of space writing.

- intensity or nuance symbols (attack modes, vibrato…) have the same meanings as in classical notation;
- phrase symbols (legato, staccato, trills…) generally have the same meanings as in classical notation; Link curves between events will generate continuous movements (thus thwarting graphical discretization);
- graphical abstraction gives access to structural notions such as sentences, chords... And consequently, to compositional transformations on a finally visible structure.

Details:

- A “speaker clef” can be added at the beginning of the score, in place of the traditional treble or bass clefs (see Figures 3 to 7).
- Sharps and flats are not used, as tonality or modality do not make sense. However, in a 3D situation, sharps and flat could very well be diverted from their traditional use to indicate top and bottom.
- In multiphonic music, several coupled staves will be used in order to note several simultaneous independent movements applied to several sound sources.
- For 16 channels, 3 linked staves can be used, depending on the loudspeakers arrangement and the desired readability. At the user's choice, staves may correspond to loudspeakers tessitura (bass, midrange, treble) or to their geographical or spatial layout.
- Example: in a surround configuration, the extra line of C bass can be used for the 5.1 bass channel.

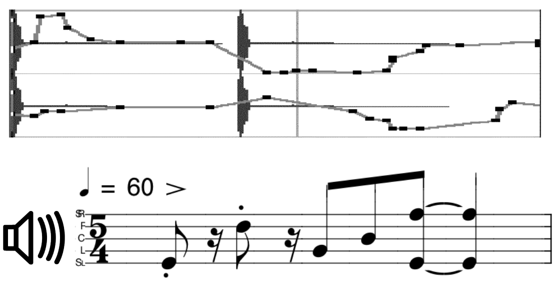


Figure 3. Question: What is the space figure displayed in these 2 examples? Top: Joystick movements representation are uneasy to read. Bottom: with practice, score space notation quickly becomes readable.

6. GRAPHICAL EXAMPLES

Some examples are given in Figures 3 to 7 and commented. Further details about technical implement-tation will be developed in the next section, which will present a practical implementation of this space notation proposal.

All examples correspond to 5.1 listening situations. Each staff line is associated with one speakers as shown to the left of each staff (SL = Surround Left, L = Left, C = Center, R = Right, SR = Surround Right).

Figures 3 or 7 – placing opposite a joystick spatialization representation with the same space notation example – should finish to convince the most recalcitrant on the readability question.

Figure 8 – later in this text, in the next section – displays another possibility of graphic representation or notation, i.e. as piano roll or barrel organ cartons. Practically speaking, this notation is more precise, but intellectually less readable insofar as it does not allow displaying accentuation or liaison criteria.

In practice, combining both notations (score and piano roll) is very powerful, easy to use and easy to read.

7. MIDI IMPLEMENTATION

Using this score system gives access to any musical notation software, as shown in Figures 3 to 7. It also allows access to MIDI encoding, so to take advantage of your favorite DAW infrastructure: effective simultaneous management of audio or MIDI events according to time, efficient visualization of these same parameters in various forms, automations... With a few minor diversions, it is quite possible to respect the double constraint set out in point (c) of Table 3 i.e. both “play what is written” or “write what is played”.

7.1 General Description

MIDI implementation principle is displayed at Figures 8 and 9, at Tables 4 and 5, and explained below:

- each **MIDI channel** corresponds to an input audio track;
- each **volume controller** (Ctrl 7) modulates the incoming audio signal intensity, either statically (balance between the channels), or dynamically (real time performance)[11];
- **MIDI note** codes apply to the outgoing signal, i.e. to the loudspeaker drive: appropriate MIDI height chooses 1 loudspeaker, and its MIDI velocity sets the loudspeaker amplitude. Velocity makes it possible to individually control the intensity of each loudspeaker statically;
- 2 envelope controllers (Ctrl 72 attack and Ctrl 74 release) allow switching from staccato or ping-pong mode to a “continuous” legato phrasing;
- other spatialization features can be modified by MIDI controllers, the use of which is described at Table 4.

Nothing is fixed, as in the MIDI standard; everyone can use its own conventions depending on habits or work to be done.

[11] Which is a brand normal situation used for audio track automation.

Figure 4. Three simple space movements.
a) Panning effect between rear left (SL) and front left (L) loudspeakers.
If the tempo is 60, this movement spreads over 2 seconds.
b) Ping-pong effect between the 2 same speakers. The sound lasts 0.5 seconds on each speaker.
c) Continuous intensity fluctuation on a single loudspeaker (under each note: velocity indications).

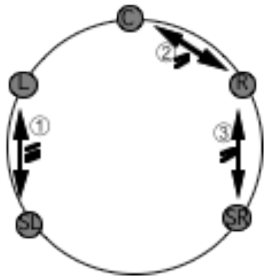
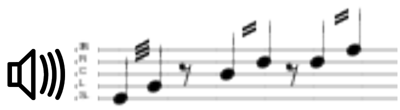
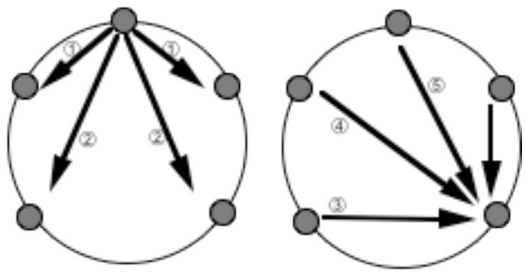


Figure 5. Three space trills . Left: graphical notation. Right: score notation. We note that any musical symbol (silences, trills...) applies to space without problem.

Note 1: if these 3 trills movements should take place simultaneously (instead of sequentially), the score notation would not pose any problem: neither writing nor reading. This simultaneity would be much more difficult to realize (and to read) with a joystick (because of the lack of polyphony of such a device).
Note 2: the reader will note that the left side figure does not allow apprehending temporality.
Note 3: space chords are impossible to realize with a joystick or XY curves.



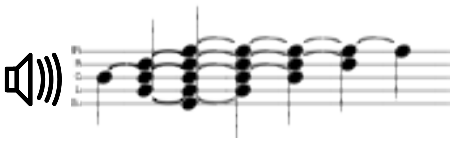


Figure 6. Space crescendo and decrescendo / realized by means of a mass change. Left: graphical notation. Right: score notation.

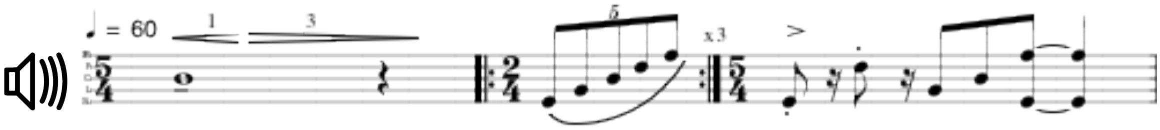
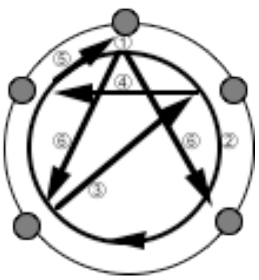
Note 1: here appears the notion of space polyphony or space mass (simultaneous use of several speakers). This effect is simply written by using notes chords.
Note 2: same remark on the representation of temporality.



Figure 7. More complex space figures: hold, rotation, then zigzag. Left below: graphical notation. Above: score notation. Description: sound apparition in 1 second on the central loudspeaker and disappearance in 3 s. // 1 second of silence. // 3 full rotations on all loudspeakers in 2 seconds, followed by a syncopate zigzag, then a chord on the 2 rear speakers.

Notation examples can be multiplied at will. Sophisticated space figures notation does not pose any problem (whereas XY representation – or any curve notation – becomes unreadable). Re-reading and comprehension are easy.

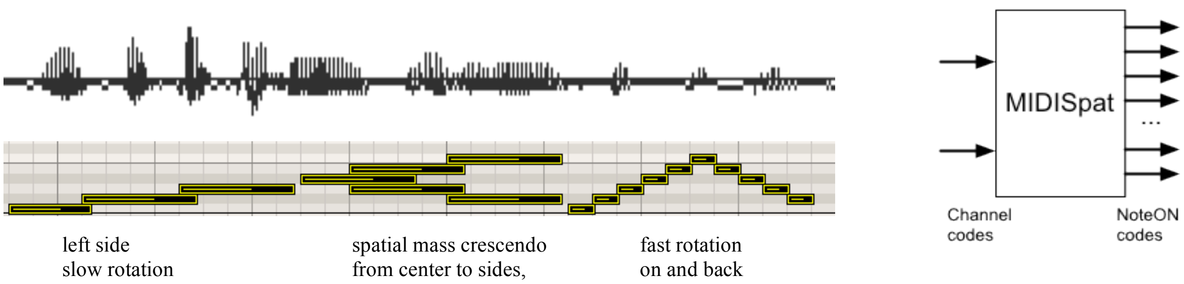


Figure 8. Driving and displaying spatialization via MIDI. The basic idea is to propose an easy edition and easy visualization of a sound space setting up, in synchronicity with the audio signal. Top: 1 mono or stereo audio track to be divided into 16 audio outputs (mono-stereo switching is automatic in Reaper). Bottom: 1 MIDI track to drive spatialization. Channel codes handle signal inputs. Note codes handle loudspeakers outputs.

MIDI controllers		parameters
1	modulation wheel	distance control (by means of filtering + reverberation).
…		
7	volume	input audio track level control, i.e. global nuances during a trajectory.
…		
64	sustain	allows holding notes (space positions)
…		
72		release [0-127] [0.1 – 16s]
73		attack [0-127] [0.1 – 16s]
…		
91	reverb	
…		
	all note off	switch off all the notes
	reset	reset all parameters

Table 3. Use of MIDI controllers codes for sound spatialization control in MidiSpat.

7.2 MidiSpat: a Simple MIDI Controlled Audio VCA

MidiSpat plug-in[12], developed in a snap thanks to the Reaper software JS language, follows many prototype versions written in Max / MSP. The total integration within Reaper (sounds, plug-ins, automations) greatly facilitates the composer's life, especially since Reaper is the most versatile software for routing audio tracks.Reaper also allows creating mixed tracks: MIDI + audio, thus offering a complete entity dedicated to audio signal spatialization. MidiSpat plug-in – used as a track insert – receives MIDI notes that will drive the audio signal to (up to) 16 audio outputs (see Figure 8).

Simultaneous spatialization of several audio tracks is not a problem; spatialization is done source by source, the audio result being automatically summed by the host software.

[12] MidiSpat plug in is available at: http://tc2.free.fr/espace/midispat.html.

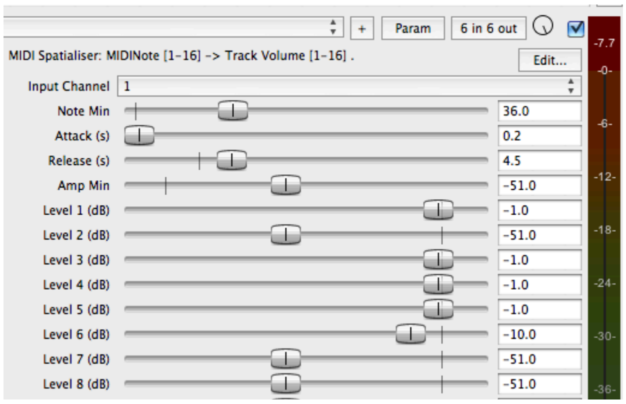


Figure 9. MidiSpat plug-in at use.

noteON [1-16][13]	selects the audio track output opens VCA according to velocity
vel [0-127]	determines output level MIDI velocity is graduated in dB vel 106 = +6 dB vel 100 = 0dB vel 0 = -100 dB
channel [1-16]	selects audio source input

Table 4. Use of MIDI note codes for sound spatialization control in MidiSpat.

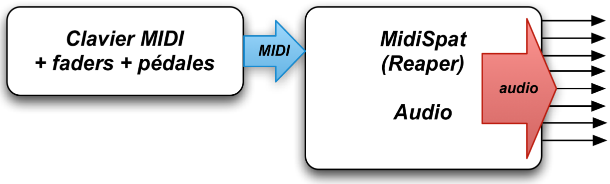


Figure 10. Live spatialization performance with a MIDI keyboard.

[13] Lowest noteON values were chosen, leaving the opportunity to use a synthesizer or a sampler on the same MIDI channel.

7.3 Space Instrumental Performance and Space Trajectory Memories

The previous presentation (Section 7.2) describes non-real time studio composition work. But the present device is equally usable in live performances or live electronics situations.

Reaper software offers unexampled audio routing, as well as a simple programming language (derived from C) allowing to write one's own plug-ins.

A MIDI keyboard with a modulation wheel (distance), a volume pedal (nuances), a sustain pedal (hold) and various faders (attack and release envelopes) makes it easy to spatialize any live audio signal (see Figure 10). The MidiSpat plug-in lets you turn any played MIDI event into volume curves controlling audio output levels of each Reaper track. A space-performer requiring more virtuosity can profitably use any MIDI sequencer to record space sequences step by step, correct mistakes and thus refine its performance. Using a MIDI sequencer allows memorizing an interpretation and visualizing it either in score mode, or in "piano roll" or "grid edit" mode. Possibilities of creating, reading, understanding, manipulating a spatialization performance are excellent.

This "instrumental keyboard spatialization performance" has been used by the author on several occasions in various public concerts. This surprising new practice is similar to the interpretation of acousmatic music using a mixing desk console. For a mid-level keyboard player, learning keyboard spatialization requires only a few days of practice, for a result that is otherwise rich and virtuoso than the one obtained using a mixing desk console (or a joystick).

It should be noted that virtuosity can be further enhanced by preparing MIDI spatialization sequences (trajectories or spatial mass changes) in advance, storing them in memories or presets and triggering them during performance. Ableton Live software is particularly well suited to this kind of work.

8. CONCLUSIONS

8.1 In summary

The term notation refers to a set of conventional signs by which sounds of music and how they should be played are written: letters, figures or graphic signs, representing musical phenomena, which are transcribed on paper in an universally admitted format. Notation by signs requires segmentation and discretization of musical phenomena; that is to say a simplification of reality. Only this "sacrifice" makes notation possible, but in return it offers access to the complexity of a language, to abstraction.

Regarding this model, the author proposes to discretize the space phenomena and to abandon the curves network representation. This choice is justified as a matter of course, if one accepts to look at musical notation adventures and history (ekphonetic notation, neumatic notation, interval notation...). Current spatial representations – when they exist – are strangely similar to early Middle Ages ones (see Figures 1 and 2).

This score notation is much more readable and understandable than representations by curves networks presently proposed in all digital audio software. The two essential concepts adopted are: discretization of space phenomena and creation of an elementary space event (ese) carrying 2 main characteristics: intensity (of a loudspeaker) and duration.

These simplifying choices make it possible to hook on the Western notation score paradigm, whose benefits are immediately apparent.

8.2 Advantages

This simplification of reality for scoring purposes has many advantages:

- readability and comprehension are far superior to the representations proposed in digital audio software;
- temporal organization is clear, thanks to time spread events on a horizontal axis;
- synchronization with musical events is obvious;
- gripping durations is easy, thanks to the usual symbols;
- space polyphony or work on space masses pose no problem of notation nor representation.
- and finally, the multi-secular habit of using score does not entail new learning.

This last point reinforces the idea that space can be considered as a fifth parameter of sound, in the same way as height, duration, intensity and timbre.

8.3 Validation

In a 1998 paper [10], the author accurately described – on about one page – the essential space notions, in form of a physical or phenomenological description.

In a paper dated from 2008 [1][2] (and following a former study [11]), the author highlights 5 modes of space perception, with various families of adjectives to describe or characterize spatiality or mobility (see Table 1).

The present notation proposal and its software implementation fully respects all this knowledge; and allows engaging without constraint all types of space activities, with any spatialization modes, real time or deferred time. In no case does this change perception.

By taking into account space events attack and release, discretization either becomes imperceptible because it is smoothed or becomes perceptible (which is a new situation impossible to realize with continuous curves); the proposed notation is perfectly compatible with any current spatialization practices and even allows considering instrumental performance of space.

It is thus easy to use the past experience and know-how, as well as the numerous existing notation software, with very few diversions.

The musicologist will also find his account by the existence of a written support giving access to space analysis, structure extraction of compositional thought, ideas formalization.

8.4 Future studies

The main problem with this proposal is essentially psychological or symbolic. Will composers issued from concrete music agree to use the fundamental tool of abstract music?

The first presentations of this notation in France suggest that the answer is NO!

Yet the step to a great progress is a tiny one, when one think that all the computer tools described here are at everyone's fingertips in all digital audio sequencers (Cubase, Logic Audio, ProTools or Reaper).

Acknowledgments

Thanks to Thélème Contemporain for his unwavering support. And J. Larran and F. Ferro for their friendly proofreading.

9. REFERENCES[14]

[1] B. Merlier, "Réflexions à propos de la mise en espace de la musique électroacoustique dans les logiciels audionumériques", in *Actes des Journées d'Informatique Musicale (JIM'05)*, Maison des Sciences de l'Homme Paris Nord, 2005.

[2] B. Merlier, "Vocabulaire de la perception de l'espace en musique electroacoustique composée ou spatialisée en pentaphonie", in *Electroacoustic Music Studies conference (EMS)*, Paris, 2008.

[3] B. Merlier, "Vocabulary of space perception in electroacoustic musics composed or spatialised in pentaphony", in *Proceedings of the Sound and Music Computing conference (SMC)*, Berlin, 2008.

[4] J. Bresson, M. Schumacher, "Representation and Interchange of Sound Spatialization Data for Compositional Applications", in *Proceedings of the International Computer Music Conference (ICMC)*, Huddersfield, 2011.

[5] N. Peters, T. Lossius, J. C. Schacher, "The Spatial Sound Description Interchange Format: Principles, Specification, and Examples", *Computer Music Journal*, vol. 37, no. 1, 2013.

[6] J. Chailley, *La Musique et le signe*, Les introuvables, Éd. d'aujourd'hui, Lausanne, 1967.

[7] W. Tappolet, *La Notation musicale et son influence sur la pratique musicale du Moyen-Âge à nos jours*, Editions de la Baconnière, Neuchâtel, 1947.

[8] C. Seeger, "Notation prescriptive et notation descriptive", in *Analyse Musicale*, no. 24, 1991. [In English:] "Prescriptive and Descriptive Music Writing", in *The Musical Quaterly*, vol. 44, no. 2, 1958.

[9] B. Merlier, *Vocabulaire de l'espace en musiques électroacoustiques*, Editions Delatour France, 2006.

[10] B. Merlier, "À la conquête de l'espace", in *Actes des Journées d'Informatique Musicale (JIM'98)*, Marseille, 1998.

[11] B. Merlier, "Vocabulaire de l'espace en musique électroacoustique : présentation, problèmes et taxinomie de l'espace", in *Electroacoustic Music Studies conference (EMS)*, Beijing, 2006.

[14] All the papers about space written by Bertrand Merlier are available on Thélème Contemporain website: http://tc2.free.fr/espace/.

SOME APPROACHES TO REPRESENTING SOUND WITH COLOUR AND SHAPE

Lindsay Vickery
Edith Cowan University
l.vickery@ecu.edu.au

ABSTRACT

In recent times much of the practice of musical notation and representation has begun a gradual migration away from the monochrome standard that existed since the emergence of printed Non-Western music in the 16th century, towards the full colour pallet afforded by modern printers and computer screens. This move has expanded the possibilities available for the representation of information in the musical score. Such an expansion is arguably necessitated by the growth of new musical techniques favouring musical phenomena that were previously poorly captured by traditional Western musical notation. As time-critical form of visualisation there is a strong imperative for the musical score to employ symbols that signify sonic events and the method of their execution with maximal efficiency. One important goal in such efficiency is "semantic soundness": the degree to which graphical representations makes inherent sense to the reader. This paper explores the implications of recent research into cross-modal colour-to-sound and shape-to sound mappings for the application of colour and shape in musical scores. The paper also revisits Simon Emmerson's Super-Score concept as a means to accommodate multiple synchronised forms of sonic representation (the spectrogram and spectral descriptors for example) together with alternative notational approaches (gestural, action-based and graphical for example) in a single digital document.

1. INTRODUCTION

Visual representation of the multi-parametrical nature of both sound and musical notation is an enduring "wicked problem". It is also a time critical problem, complicated by the differences between the human auditory and visual systems and even mental chronometry. Since the end of the common practice period the timbral pallet employed by composers and performers has greatly expanded, and particularly since the advent of digital computing, the range and detail of the spectral description of sound has exponentially increased.

Because of the technological limitations of typesetting Common Practice Notation (CPN) developed almost entirely in monochrome and with a vocabulary of fixed symbols. Since the advent of colour printing and colour screen-based scores there is no reason for these constraints to continue. This paper explores the value and potentials of employing colour and shape to accommodate the multiparametrical description of sound and notation, in particular through the utilisation of "Cross-Modal Correspondences" between auditory and visual perception as a means for developing semantically sound strategies and methods for representing sonic phenomena and notation.

In the context of the expanding range of forms of representation and notation, Emmerson's notion of the Super-Score, a digital document bundling media relevant to a soundwork/composition together, as a means for composers, performers and researchers to synchronously document and explore sonic works.

2. CROSS-MODAL CORRESPONDENCE

The perceptual phenomenon now generally referred to as Cross-Modal Correspondence (CMC), defined by Marks as "natural correspondences between experiences in different sense modalities" [1], provides some prospects for strategies that might improve the semantic soundness of music notation.

CMC is roughly analogous to the better known rare and idiosyncratic condition Synaesthesia, which causes individuals to experience sensory input cross-modally, the most common form being the simultaneous activation of the senses colour and sound. CMC along with synaethesia has been the subject of scientific enquiry for over two hundred years [2]. In the late 1960s Luria referred to CMC as the 'remnants' of synaesthesia "that many ordinary people have, which are of a very rudimentary sort (experiencing lower and higher tones as having different colorations)" [3]. Later research has tended to separate the two phenomena [4]: the relatively rare condition of synaesthesia (occurring in 0.5% of the population [5]) characterized as absolute, unidirectional, intransitive, rigid and CMCs, some of which are universal [4] as relative, bidirectional, transitive and malleable.

- CMCs are relative: they tend to be ordered in continua – i.e. low to high, soft to loud, dense to diffuse etc.;

- Synaesthesia is unidirectional: the sensory correspondences are not invertible – a pitch may elicit a particular colour but a that colour may not elicit the same pitch;
- CMCs are transitive: "the same core correspondences should emerge whichever sensory feature is used to probe them, confirming that the en bloc alignment of the dimensions is context invariant" [6];
- CMCs are malleable, they may be used as a component of a semantic languages that are able to be learned;

Spence & Deroy [7] have proposed that there are multiple forms of crossmodal mapping: statistical, structural, and semantic. They define statistical, the most hard-wired of the three, as occurring due to the similarities of the transformation of sensory information into perceptual information; structural, more learned and environmental resulting from functional regularities that can be commonly observed in the physical environment; and semantic correspondence, the most conscious and trained occurring when two objects are linked conceptually [8].

It is notable that these categorisations align with aspects of research in other fields for example Moody's *Physics of Notations Theory* [9] (i.e. cognitive fit, semiotic clarity, visual expressiveness, semantic transparency), MacEachren's *Expanded Graphic Vocabulary* [10] (i.e. location, size, resolution, transparency, colour, texture, orientation, arrangement, shape), Wierzbicka's *Semantic Primes* [11] (quantifiers, evaluators, descriptors, actions, events, movement, contact, time, space, intensifiers) and Patel et al.'s [12] discussion of *Augmentative and Alternative Communication Symbols* (gestalt, semantic attributes, cartoon conventions, compositional distinctions, line interpretation).

Interest in CMCs may have been initially sparked by the Köhler's bouba/kiki experiment [13]. In this experiment, "because of the sharp inflection of the visual shape, subjects tend to map the name kiki onto the (pointed, star-like) figure (…), while the rounded contours of the (other) figure make it more like the rounded auditory inflection of bouba" [14]. The cross-modal mapping tendency suggested that there are "natural constraints on the ways in which sounds are mapped on to objects" [4], for human perception in general and beyond the atypical perception of synaesthetes.

The social sciences appear to have contributed the notion that mental concepts could be arranged cross-modally in oppositional continua. In 1954 Guttman [15] proposed a circular psychometric structure called the circumplex for spatially and hierarchically situating emotions, it was applied to personality by Leary [16] and Block [17] and added the further cross-modal dimension of colour to the circumplex.

> *We can compare the two dimensions of the facial-expression surface to the blue-yellow and red-green axes of the color surface. This immediately suggests that there may be a third dimension, corresponding to visual brightness. The third dimension for facial expressions might well be the intensitive one we considered earlier, level of activation.* [18]

This correlation, and other similar associations, for example between shapes and sounds, facial expressions and colours [19] and colour and a range of musical phenomena including timbre, pitch, tempo, intervals, triads and musical genres in non-synaesthetes [20, 21], continue to be explored extensively and cross-culturally [22, 23, 24] providing insight into potentially more natural means to visually represent sonic phenomena.

3. COLOUR

There are a number of perceptually based restrictions upon the use of colour to represent sound. There are no clear direct perceptual analogies between human visual and auditory processes. The ear senses sound continuously with a resolution up to 15-20kHz, while in reading visual field is sensed in grabs of detailed data through focused fixations of about 4cm^2 for a minimum duration of approximately 5kHz or 200ms. The eye is much 'slower' than the ear. This is a crucial issue for the depiction of sound visually, as eyes are only capable of sensing visual detail of many orders lower than the ear senses sound - perhaps as much as 400-600 times lower.

The wide frequency sensitivity of the ear (in the order of ten octaves) also contrasts the single "visual octave" of the eye: colours in the visual spectrum do not repeat - ultraviolet and infrared are both invisible. In addition, although the eye can finely discriminate variations in colour, Green-Armytage suggests that, in terms of representing data with colour, a limit of 27 tones is "the largest number of different colours that can be used before colour coding breaks down" [25]. In contrast the ear can discriminate pitch differences as small as five cents [26].[1]

Although there have been numerous colour to sound mappings proposed over the centuries, the investigation of CMC between colour and sound appears to date from Schlosberg [18] and later to Plutchik who claimed that:

> *the primary emotions can be conceptualized in a fashion analogous to a color wheel-placing similar emotions close together and opposites 180 degrees apart, like complementary colors. Other emotions are mixtures of the primary emotions, just as some colors are primary and others made by mixing the primary colors.* [28]

Plutchik's mapping has been influential, underpinning a wide range of musical projects drawing on colour as a metaphor for musical expression [29, 30, 31, 32, 33, 34, 35, 36, 37]. Palmer et al's *Emotional Mediation Hypothesis* [38] expanded this research proposing, "that color and music are linked through shared emotional associations", showing systematic relationships between colour and a range of musical phenomena including timbre, pitch, tempo, intervals, triads and musical genres [20, 21].

[1] A number of the limitations of representing sound and notation are discussed in detail in [27].

Their investigation of instrument timbre interestingly showed that the average yellow-blue value was "correlated with timbre attack time whereas average red-green value is correlated spectral brightness [21]. Close inspection of their data shows that despite this correlation, the colours chosen by their participants were extremely varied: although the correlations to place the fitness of colours according to timbre attack and spectral brightness appeared to be strong, but did not point to *specific* colours as being more appropriate. This is crucial as it demonstrates that CMCs are relative, malleable and at least partially "explainable by exposure" [4] to environment and/or learning.

Application of cross-modal principals to colour is also problematic because of the difficulty of establishing a meaningful mapping of bright and dark colours. Whereas sound is mapped in a broadly linear fashion with the cochlea capturing frequencies continuously from high to low, the eye combines data from a range of different sensors – colour through three cone cells and luminosity through rod cells. The result is that vision is not mapped in a linear fashion: if it were, the light spectrum would appear as a bright to dark continuum from purple - the highest frequency colour to red the lowest frequency colour. The arrangement of rods and cones gives rise to anomalies such as the non-sequential perceptual "brightness" of colours such as yellow, cyan and magenta in the colour spectrum.

	H	S	L
Pale Goldenrod	59	30	94
Beau Blue	191	18	91
Sandstorm	53	64	90
Tea Green	108	11	90
Baby Pink	11	15	89
Thistle	291	9	83
Persian Orange	22	57	80
Ruddy Brown	21	74	78
Mulberry	328	53	74
Firebrick	2	76	71
Cool Grey	225	32	71
Dark Sea Green	155	30	70
Dollar Bill	79	59	69
Brass	54	60	62
UCLA Blue	222	53	62
Camouflage Green	115	40	61
Raw Umber	27	67	57
Dark Slate Blue	226	62	55
Cordovan	9	46	53
Viridian	176	41	52
Dark Slate Blue	274	61	48
Dim Grey	48	10	41
Dark Slate Grey	163	43	33
Caput Mortuum	341	68	32
Rifle Green	101	45	30
Smoky Black	0	0	4

Figure 1. Green-Armytage's 27 discriminable colour tones (adapted by the author in descending order of approximate perceptual brightness).

CIELAB colour space [39] attempts to mimic the non-linear response of the eye by modeling cone responses. Mapping the 27 colour tones suggested by Green-Armytage (Figure 1) to CIELAB colour space gives an approximate continuum of hues from brightest to darkest, together with a notional maximal number of discriminable hues. It is also possible to group the hues according to their proximity to spectral colours – reds, oranges, yellows etc. – in order to depict related sonic phenomena or notations: instrumental timbre variation, gesture or stick/mallet designation for example.

A further issue is the multi-parametrical nature of the representation of sound and notation. This prohibits the development of a standard method for applying colour in musical representation: it will always be necessary for the composer to make choices about the phenomena that is represented by any colour, colour continuum or colour attribute (for example mapping hue to brightness, noisiness to saturation and spectral skew to luminance [40]). What is clear is that the use of colour in representation and notation provides powerful tool for the formation of what Moody terms "Perceptual Discriminability" [9] maximising the distinctness of separate phenomena in the manner routinely employed in data visualisation [41], transport maps [25], and websites [42].

4. SHAPE

The simplest and perhaps least contested of these Cross-modal mappings is the vertical spatial depiction of frequency, in which higher frequencies are also spatially represented vertically higher on the page. This visual pitch metaphor that "while culturally diverse, may be based upon basic underlying mappings, stemming from bodily-based intermodal interactions with the physical environment" [22] and has been demonstrated pan-culturally [22] and in infants as young as 1 year old [43].

Vertically proportionality is one area in which CPN is in conflict with CMC: instruments higher on the score are not necessarily higher in pitch and a note may occupy the same vertical location if whether it is sharp, natural or flat. The same space between stave lines may represent a minor or a major third. This is a significant problem to overcome as musicians trained in this tradition can both read "music" and mentally sonify it (as opposed to visualise), but this process is only afforded by a significant range of implicit literacy skills.

The practice of music notation developed in the context of tonal/modal music many attempts have been made to "reform " this deficiency, to allow for efficient representation of the chromatic and smaller grained pitch grids/scales [44].

This issue is nowhere more apparent than in the representation of electronic sounds and field recordings. Robert Erickson's *Pacific Sirens* (1968), one of the first works to use a spectrogram-score as a means for performative engagement the timbral complexity of a field recording with instruments, is an example of the conflict. The work employed proportional notation, with an external time source (stopwatches), together with a spectrographic transcription of sound/frequency morphologies and directed the "improvising" performer's to "listen into" "the spectral complexes of the environmental noise and appropriately blend and protrude" [45]. The frequencies of the field recording are overlaid against traditional treble and bass staves and although spatially time is represented proportionally, pitch is not and therefore sliding pitches are symbolically, rather than accurately portrayed in the score (Figure 2).

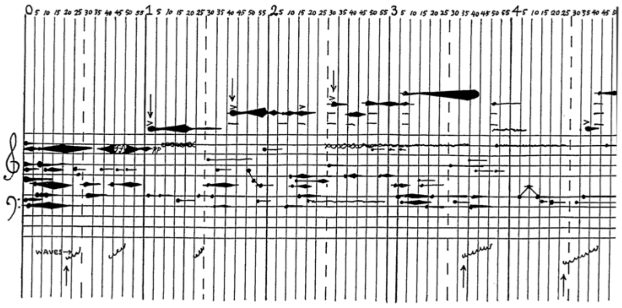


Figure 2. Excerpt of spectrographic score of Robert Erickson's *Pacific Sirens* (1968) © Smith Publications 1969.

A potential solution to this problem is a vertically proportional stave[2], however such novel systems are often unpopular with performers as indicated by the number of abandoned proposals littering the last two hundred years of music history [44].

CPN is of course not horizontally proportional either (spacing is principally determined by typographical compactness rather than the duration of note events). However horizontally "proportional notation" or "Time Notation" pioneered by Earle Brown in the 1950s [46] has been quite widely adopted by composers as diverse as Cage, Berio and Grisey. In propotional notation, the "spacing and length of the notes on the page, are put into a more or less direct relation to the timing and duration of the sounds [47].

The spatial/durational relationship in works by the New York School composers tended to be "simply observed" [48], as Cage noted, rather than executed in the context of a temporal grid (in contrast to Grisey's *Partiels* (1975) or scrolling/swiping digital notation [49] for example). This issue of precisely representing a sense of metricality, complex rhythmic structures and coordination of multiple performers in proportional notation is a significant issue that is only partially resolved by non-visual means such as a conductor, external click tracks [50] or animated notation [51].

Using the inherent semantic qualities of graphical shapes to denote sonic morphology also has valuable potential. The development of "a new graphical vocabulary based on spectromorphology" [52], the visualization of sonic phenomena, has been most fully explored in the field of spectromorphology in acousmatic music [53, 54, 55].

Blackburn refers to the cross-modal quality of acousmatic music listening: "it is frequently reported that, in concert, acousmatic music has the powerful effect of conjuring imagery, shapes, trajectories and spaces, which we as listeners proceed to describe verbally [52]. Blackburn's graphical vocabulary not only visualizes individual "sound units" but also shows how they can be "strung together to form longer phrase lengths" or "morphological strings" [54]. She emphasizes the use of perceptual metaphors, stating that words that are "more readily visualized ie. *spiral, flock, stream* and those with a clear associated physicality ie. *fly*, *drift*, *attack*, appear better suited for informing sound material creation" [55].

Spectromorphological representations, in two dimensions, share the same space as pitch/duration representations, and there are potential conflicts in the signification of other sonic parameters in this space, perhaps particularly dynamics/intensity, which are often depicted by increased size. As the size of a shape increases it also occupies the vertical continuum allotted to pitch, which is problematic when a sonic object is varying both in pitch and dynamic. Solutions to this problem might include three-dimensional representation or indicating pitch (or spectral centroid) with a line of consistent size and contrasting colour to other parameters.

The process of eye fixation (with a "gaze frame" of roughly $4cm^2$ for periods of time in the order of 200–400ms) is very slow in relation to both the auditory system and the mental chronometry that allows for the execution of physical actions. The author's previous research has suggested reading becomes difficult beyond a rate of approximately 3cm/s [36, 56]. Although musicians are capable of performing nuances at extremely minute durations, the eye is not capable of capturing data quickly enough.

Music is however not always performed in a sight-reading context, and perhaps the preponderance of Western Art Music presupposes rehearsal and practice. One solution available to for screen-based representation is multiscale representation analogous to digital maps, permitting magnification of the score/representation, while maintaining a constant graphical density. In cartography, for example, Bertin [57] suggests no more than 10 semantically meaningful units should be represented per cm^2. This is a feature of Digital Audio Workstations and Spectrographic software, but could also usefully accomodate notation that is either too fast or too detailed to read, providing a less detailed representation at lower resolutions.

5. HEURISTICS

Although CMC is malleable and relative it is possible, in conjunction with the perceptual restrictions discussed previously, to develop some heuristics or "rules of thumb" that might guide their implementation in sonic representation. Statistical, structural and semantic correspondences are somewhat fluid due to environment and training, however statistical CMC appears to be the most difficult to unlearn[3]. Tsiros' compilation of crossmodal sight-to-sound research [8], shows that the strongest correspondences (in order of strength) are spatial height to pitch association (although size to pitch is also significant), amplitude to light intensity, duration to horizontal length and texture granularity to timbre. (By extension shape is therefore

[2] Discussed in [36].

[3] For example cellists can "unlearn" the correspondence between rising pitch and lower spatial position on the fingerboard. Apparently jazz pianist Joe Zawinul, however used the technique of inverting the pitch of his keyboard in order to break his familiarity with its spatial layout [58].

associated with amplitude related sonic morphology). These correspondences should therefore be the most crucial to consider in the creation of a semantically sound notation.

More overtly semantic correspondences such as symbols, pictograms and text are more flexibly applicable for specific representation requirements. They are capable of being learned and indeed many are already embedded in CPN in varying degrees.

It is important to remember that although the acquisition rate of the eye for linear information is potentially as high as 20cm/s, practical examples indicate rates beyond 4cm/s become uncomfortable for the performer to read in an accurate and synchronous manner [59]. Therefore, detailed depictions of sonic events for performative reading or as a representation of audio presented in real-time, are restricted to approximately 4cm per second of sound.

Colour parameters such as hue, saturation and luminance may be mapped to spectral descriptors such as brightness, noisiness and roughness for example, to produce visual representations[4]. As mentioned colour discrimination for the purpose of identifying distinct phenomena is restricted to approximately 27 hues. Figure 3 shows the pallet of 23 hues used to depict separate instrumental parts for the chamber orchestra and fixed media work *bascule.* The colours are segmented into groups by instrument family (yellows for flutes, blues for strings for example) as well as subset variations of those hues to depict individual instruments within a family (Firebrick red for clarinet and Caput Mortuum for bass clarinet for example). This arrangement exploits the CMC between higher pitch and brighter hue.

piccolo	alto sax	e. guitar 3	violin 1
flute 1	tenor sax	e. guitar 2	violin 2
flute 2	baritone sax	e. guitar 3	viola
clarinet 1	trumpet	accordion	cello
clarinet 2	horn	cymbal	double bass
bass clarinet	trombone	bass drum	

Figure 3. Colour-to-Instrument coding employed in Vickery's *bascule* (2016).

Figure 4 shows an excerpt from the score *wellington forest* (2017) in which a spectrographic representation of an accompanying field recording is combined with temporally proportional traditional and rhythmic notation. The spectrogram was produced by "threshing" the field recording to remove all but the highest amplitude sounds (in this case frog croaks) and serves as a guide to the filed recording for the performers. Pitches are indicated via "cut out" staves or a five-line stave. Beams are used to indicate emergent phrase structures and stems are placed on the left side of each square notehead to aide coordination of the performers with the scrolling score and embedded soundfile. The score scrolls as a rate of 6.19mm/s.

In the excerpt from *[opi'lka* (2017) for septet and fixed media (Figure 5), a number of conventions are employed simultaneously to convey a variety of performance practices. Like *wellington forest* performers are provided with cut out pitch indications. The flute reads their material entirely from yellow colour-coded spectrographic representations, inferring the pitch content from noteheads on the cut out stave and emulating the morphology of the spectrographic shapes. The bass clarinet (colour-coded purple) reads from symbolic notation indicating whitenoise-like "breath jets" as well as spectrographic representations of those sounds. The electric guitar, playing with a slide, has a more gestural form of notation in which the shape indicates the contour of the slide movements. The score scrolls as a rate of 16mm/s.

Like *wellington forest* the 4 instruments in *kuroinami* (2016) are provided with a spectrographic representation of the fixed media part (Figure 6). Cut away staves are only used when the instruments are playing non-open strings. The double bass part primarily uses a semantically symbolic language to depict combined sul ponticello bowing and left hand pizzicato. The biwa, viola and cello parts mostly employ a gestural notation similar to that used in Lachenmann's *Pression* (1968) with specific instructions written in English. The score scrolls as a rate of 11.73mm/s.

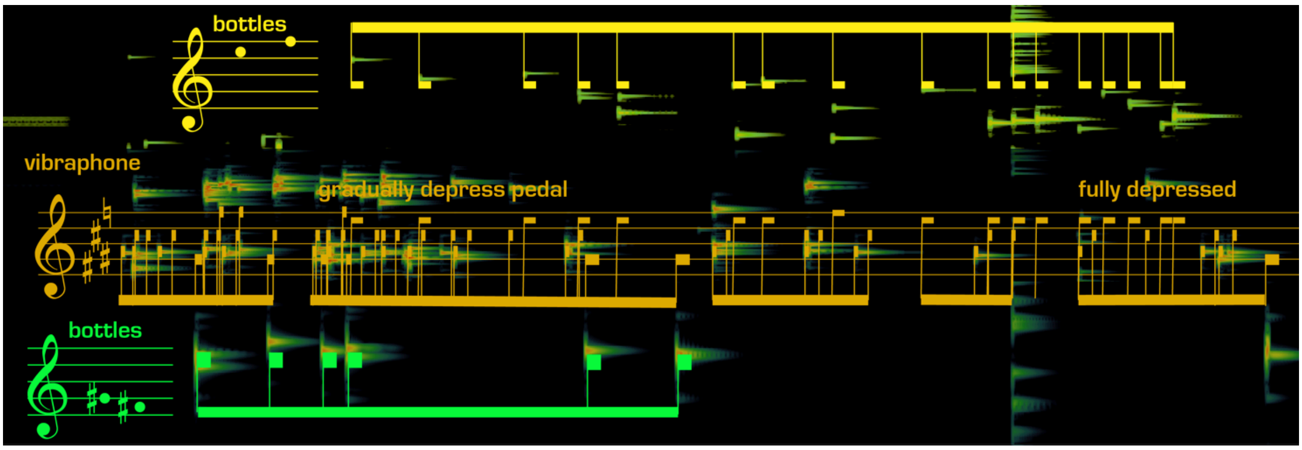


Figure 4. Excerpt from Vickery's *wellington forest* (2017) for percussion trio and field recording.

[4] A number of these issues are discussed in [40].

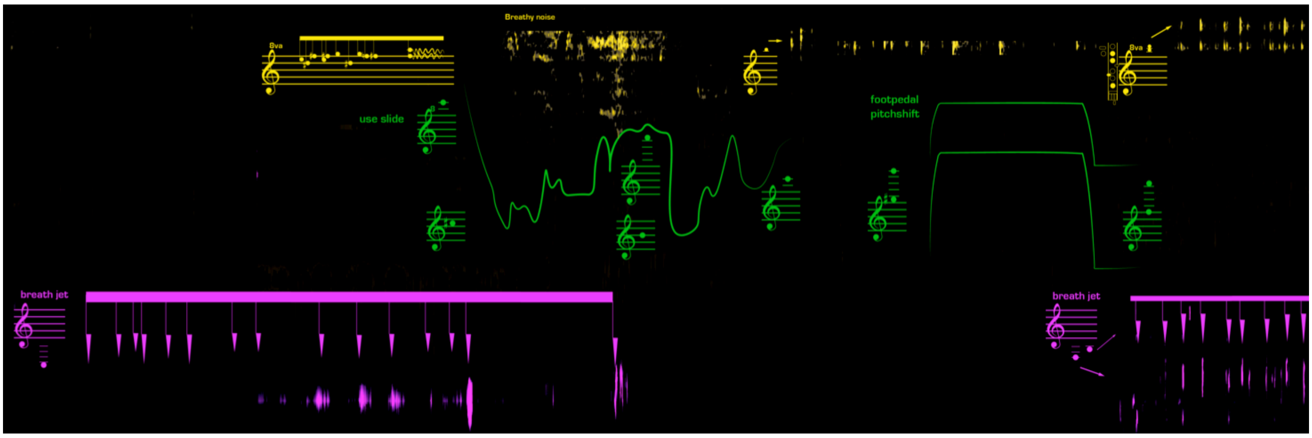


Figure 5. Detail showing excerpt of flute (yellow), bass clarinet (purple) and electric guitar (green) parts from Vickery's *[opi'lka* (2017) for flute, trumpet, soprano saxophone, alto saxophone, bass clarinet, electric guitar, prepared piano and fixed media.

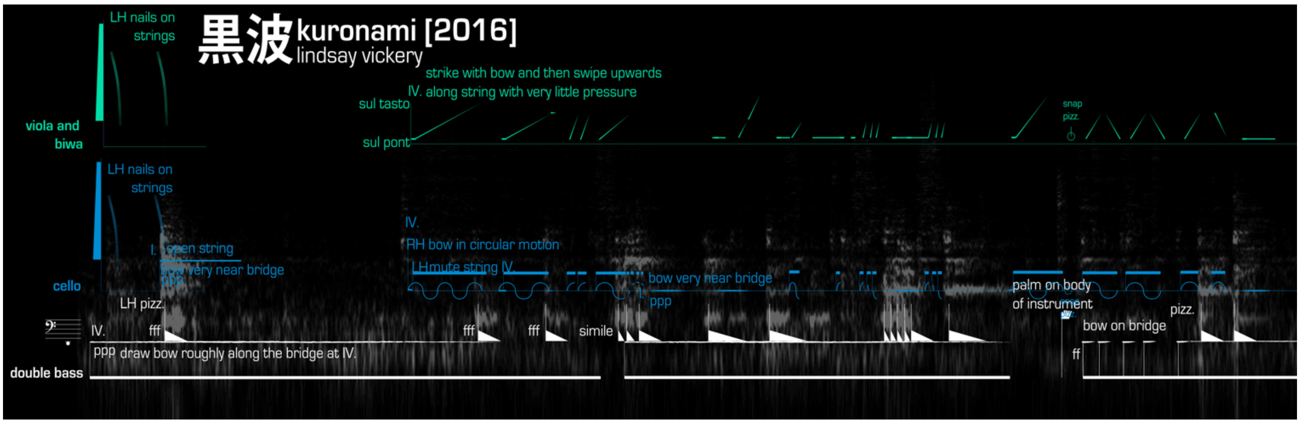


Figure 6. Excerpt from Vickery's *kuroinami* (2016) for biwa, viola, cello, double bass and fixed media.

These three scores combine spectrographic representations of sound with a different form of notation. The application of diverse notational forms highlights the fact that in each case a choice of notation most appropriate to a particular purpose was made to the exclusion of other forms. Each notational approach, CPN, proportional, tablature, gestural graphical and so on, favours different aspects of the performative requirements. Furthermore the prominence of the spectrographic representation is fixed and cannot be intensified or diminished. These developments point toward the possibility of a more multidimensional score, in which the performer may choose between and blend notational and representational approaches.

6. SIMON EMMERSON'S SUPER-SCORE REVISTITED

No single approach to musical representation can accommodate every existing notational and representational requirement. In addition to approaches that are well established in Europe such as CPN, tablature, graphic notation and the spectrogram, we might add those from Non-Western music [60, 61], Jazz [62], Popular Music [63] and European Early Music [64] and emerging approaches such a gestural [65] and "action-based" [66] notations. Related, but the other end of the resolution continuum is music analysis which often involves the schematization and compression of musical structures into meaningful components.

In 2000 Emmerson proposed that "the super-score of the future" could be a multimedia object bringing together all the necessary materials to define a sonic work [67]. His concept incorporated traditional notation, extended notation, audio, video, software and documentation. Emmerson's concept would allow for an all-encompassing digital document that would accommodate multiple synchronised forms of sonic representation, that could be viewed in multiple modes (in the way a digital map can be viewed in satellite or terrain mode) allowing the sonic phenomena and/or notations to be easy alternated.

Such a document might allow for:

- the synchronised alternation between and/or superimposition of, multiple forms of musical representation;
- linked supporting annotative media;
- multiscale representation [68] of image files (zooming);
- communication and synchronisation with digital audio and analysis tools.

- There are great advantages to "bundling" the performance or realisation materials into a single unit [67, 50].

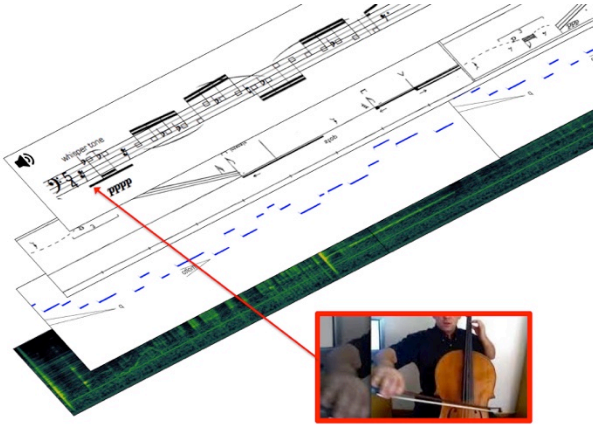


Figure 7. Depiction of the arrangement of multiple forms of notational visualization in an Integrated Score File Format.

The integrated score file format (Figure 7) would obviate the need to collect all required specifications within a single text and stream-lines inclusion of alternative forms of score and annotations by collecting them in an aligned format allowing the reader, performer, researcher to swipe between representations. The score could also be annotated with embedded text, audio and video and additional resources, such as necessary audiofiles, software, technical papers and so on.

7. CONCLUSION

The efforts to extend notation discussed here are part of an ongoing effort to better capture nuances of sound such as timbre, temperament and envelope morphology using shape and colour parameters in a manner that is concise and semantically sound. Although CMC does not provide a "magic bullet" solution, the current state of research does give helpful guidelines in regard to the appropriateness of deploying colour and shape in the service of sonic parameters.

Although the malleable quality of CMCs suggest that any system of associations can be learned it seems likely that the spatial metaphor of pitch and duration is particularly strong, and that the pre-existing (at least in English) cross-modal metaphor timbre/colour suggests the useful retention of that association. The human visual system's non-linear response to the light spectrum may by potential exploited in the service of representing multiple parameters. The contest for vertical space between pitch and dynamics is a persistent issue that will most likely elicit multiple idiosyncratic solutions.

The use of colour and shape to represent the mutliparametrical musical space embraces advances in printing and presentation technologies that will likely continue to improve.

In the context of the multiple means for representing sound and musical works, it is proposed that Emmerson's notion of the Super-Score, a digital format accommodating text, graphics, sound, video and algorithmic resources is reconsidered as a goal.

8. REFERENCES

[1] L. E. Marks, "On Perceptual Metaphors", *Metaphor and Symbolic Activity*, vol. 11, no. 1, pp. 39–66, 1996.

[2] C. van Campen, "Artistic and Psychological Experiments with Synesthesia", *Leonardo*, vol. 32, no. 1, pp. 9–14, 1999.

[3] A. R. Luria, *The Mind of a Mnemonist*. Harvard University Press, 1968.

[4] O. Deroy and C. Spence, "Why we are not all synesthetes (not even weakly so)", *Psychonomic Bulletin & Review*, vol. 20, no. 4, pp. 643-664, 2013.

[5] J. Harrison and S. Baron-Cohen, "Acquired and inherited forms of cross-modal correspondence", *Neurocase*, vol. 2, no. 3, pp. 245–249, 1996.

[6] P. Walker, "Cross-sensory correspondences and cross talk between dimensions of connotative meaning: Visual angularity is hard, high-pitched, and bright", *Attention Perception and Psychophysics*, vol. 74, no. 8, pp. 1792–1809, 2012.

[7] C. Spence and O. Deroy, "How automatic are crossmodal correspondences?", *Consciousness and cognition*, vol. 22, no. 1, pp. 245-260, 2013.

[8] A. Tsiros, "The parallels between the study of cross-modal correspondence and the design of cross-sensory mappings," in *Proceedings of the conference on Electronic Visualisation and the Arts (EVA'17)*, London, UK, pp. 175-182, 2017.

[9] D. L. Moody, "The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering", *IEEE Transactions on Software Engineering*, vol. 35, no. 6, pp. 756-779, 2009.

[10] A. MacEachren, "Map complexity: comparison and measurement", *American Cartographer*, vol. 9, pp. 31–46, 1982.

[11] A. Wierzbicka, *Semantics: Primes and Universals*. Oxford University Press, 1996.

[12] R. Patel, K. Schooley, and J. Wilner, "Visual Features that Convey Meaning in Graphic Symbols: A Comparison of PCS and Artists' Depictions", *Assistive Technology Outcomes and Benefits*, vol. 41, pp. 62–80, 2007.

[13] W. Köhler, *Gestalt Psychology*. Liveright, 1929.

[14] V. S. Ramachandran, and E. M. Hubbard, “Synaesthesia - A Window into Perception, Thought and Language”, *Journal of Consciousness Studies*, vol. 8, no. 12, 2001, pp. 3–34, 2001.

[15] L. A. Guttman, “A new approach to factor analysis. The radix”, in P. F. Lazarsfeld (Ed.), *Mathematical thinking in the social sciences*, Free Press, pp. 238–248, 1954.

[16] T. Leary, *Interpersonal diagnosis of personality: A functional theory and methodology for personality evaluation*, Ronald Press, 1957.

[17] J. Block, “Studies in the phenomenology of emotions”. *Journal of Abnormal and Social Psychology*, vol. 54, no.3, pp. 358–363, 1957.

[18] H. Schlosberg, “Three dimensions of emotion”, *Psychological Review*, vol. 61, no. 2, pp. 81–88, 1954.

[19] O. da Pos and P. Green-Armytage, “Facial Expressions, Colours and Basic Emotions”, *Colour: Design & Creativity*, vol. 1, no. 2, pp. 1–20, 2007.

[20] W. S. Griscom and S. E. Palmer, “The Color of Musical Sounds in Non-Synesthetes”, in *Annual Meeting of the Vision Science Society*, Naples, FL, USA, 2012.

[21] W. S. Griscom and S. E. Palmer, “Cross-modal Sound-to-Sight Associations with Musical Timbre in Non-Synesthetes”, in *Annual Meeting of the Vision Science Society*, Naples, FL, USA, 2013.

[22] Z. Eitan and R. Timmers, “Beethoven’s last piano sonata and those who follow crocodiles: Cross-domain mappings of auditory pitch in a musical context”, *Cognition*, vol. 114, no. 3, pp. 405–422, 2010.

[23] Y. Kazuhiko, Y. Natsumi, K. B. Schloss, L. R. Prado-León, and S. E. Palmer, “Cross-Cultural Studies of Color Preferences: US, Japan, and Mexico”. *Journal of Vision*, vol. 10, no. 7, 2010.

[24] L. R. Prado-Leon, K. B. Schloss, and S. E. Palmer, “Bach to blues, Color, music and emotion across cultures”, in W. Anderson, C. P. Biggam, C. Hough, and C. Kay (Eds.), *Colour Studies: A Broad Spectrum*, John Benjamins, pp. 366-379, 2014.

[25] P. Green-Armytage, “A Colour Alphabet and the Limits of Colour Coding”, *Journal of the International Colour Association*, vol. 5, 2010.

[26] L. K. Miller and H. Clausen, “Pitch identification in children and adults: Naming and discrimination”, *Psychology of Music*, vol. 25, no. 1, pp. 4-17, 1997.

[27] L. Vickery, “The Limitations of Representing Sound and Notation on Screen,” *Organised Sound*, vol. 19 no. 3, pp. 215-227, 2014.

[28] R. Plutchik, “The Nature of Emotions”, *American Scientist*, vol. 89, no. 4, pp. 344-350, 2001.

[29] I. Whalley, “Broadening Telematic Electroacoustic Music by Affective Rendering and Embodied Real-time Data Sonification,” *Proceedings of the International Computer Music Conference (ICMC)*, Athens, Greece, pp. 301-307, 2014.

[30] G. S. Vercoe, “Moodtrack: practical methods for assembling emotion-driven music,” PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.

[31] W. Jauk, “Interfaces–a Musical Situation”, *Musicological Annual*, vol. 43, no. 1, pp. 43-51, 2007.

[32] S. R. Livingstone, R. R. Muhlberger, A. F. Brown, and W. Thompson, “Changing musical emotion through score and performance with a computational rule system”, *Computer Music Journal*, vol. 34, no. 1, pp. 41-65, 2008.

[33] S. Baldan, A. Barate, and L. A. Ludovico, “Automatic Performance of Black and White n. 2: The Influence of Emotions Over Aleatoric Music”, in *Proceedings of international symposium on Computer Music Modeling and Retrieval (CMMR)*, London, pp. 437-448, 2012.

[34] I. R. O’Keeffe, “Soundtrack Localisation: Culturally Adaptive Music Content for Computer Games”, *The Journal of Internationalization and Localization*, vol. 2, no. 1, pp. 36-65, 2012.

[35] B. E. Carey, *Notation Sequence Generation and Sound Synthesis in Interactive Spectral Music*, Master's thesis, University of Sydney, Australia, 2013.

[36] L. Vickery, “Exploring a Visual/Sonic Representational Continuum,” in *Proceedings of the International Computer Music Conference (ICMC)*, Athens, Greece, pp. 177-184, 2014.

[37] S. Canazza, G. D. Poli and A. Rodà, “CaRo 2.0: An Interactive System for Expressive Music Rendering”, *Advances in Human-Computer Interaction*, 2015.

[38] S. E. Palmer, K. B. Schloss, Z. Xu, and L. R. Prado-León, “Music–color associations are mediated by emotion”, in *Proceedings of the National Academy of Sciences of the USA*, vol. 110, no. 22, pp. 8836-8841, 2013.

[39] G. Hoffman, CIELab Color Space. [Online]. Available: http://docs-hoffmann.de/cielab03022003.pdf

[40] L. Vickery, “Visualising the Sonic Environment”, in *Proceedings of the Australasian Computer Music Conference (ACMC’17)*, Adelaide, Australia, pp. 43-50, 2017.

[41] E. Tufte, *Envisioning Information*, Graphics Press, 1990.

[42] P. Stanicek, Color Scheme Designer, 2009. [Online.] Available: http://colorschemedesigner.com

[43] S. Wagner, E. Winner, D. Cicchetti, and H. Gardner, ""Metaphorical" mapping in human infants", *Child Development*, vol. 52, no. 2, pp. 728–731, 1981.

[44] G. Read, *Source Book of Proposed Music Notation Reforms*. Greenwood Publishing Group, 1987.

[45] J. Mackay, "On Community, Institutions and Politics in the Life and Work of Robert Erickson," in J. Wall (Ed.), *Music, Metamorphosis and Capitalism: Self, Poetics and Politics*, Cambridge Scholars Publishing, pp. 18-32, 2007.

[46] E. Brown, "The Notation and Performance of New Music", *The Musical Quarterly*, vol. 72, no. 2, pp. 180-201, 1986.

[47] C. Cardew, "Notation: Interpretation, etc", *Tempo*, vol. 58, pp. 21-33, 1961.

[48] P. D. Miller, "Indeterminacy and performance practice in Cage's Variations", *American Music*, vol. 27, no. 1, pp. 60-86, 2009.

[49] L. Vickery, "The Evolution of Notational Innovations from the Mobile Score to the Screen Score", *Organised Sound*, vol. 17 no. 2, pp. 128-136 2012.

[50] L. Vickery, "Mobile Scores and Click-Tracks: Teaching Old Dogs", in *Proceedings of the Australasian Computer Music Conference (ACMC'10)*, Canberra, Australia, pp. 63-70, 2010.

[51] R. R. Smith, "An atomic approach to animated music notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, pp. 39-47, 2015.

[52] M. Blackburn, "The Visual Sound-Shapes of Spectromorphology: An Illustrative Guide to Composition", *Organised Sound*, vol. 16, no. 1, pp. 5–13, 2011.

[53] A. Pasoulas, "Temporal Associations, Semantic Content and Source Bonding", *Organised Sound*, vol. 16, no. 1, pp. 63–68, 2011.

[54] D. Tanzi, "Extra-Musical Meanings and Spectromorphology", *Organised Sound*, vol. 16, no. 1, pp. 36–41, 2011.

[55] M. Blackburn, "Composing from Spectro-morphological Vocabulary: Proposed Application, Pedagogy and Metadata", in *Electroacoustic Music Studies Conference 09 (EMS09)*, Buenos Aires, Argentina, 2009.

[56] L., Vickery and T. Goh, "Music Screen-Reading: indicative results from two pilot studies", *Proceedings of the Australasian Computer Music Conference (ACMC)*, pp. 119-125, 2015.

[57] J. Bertin, *Semiologie graphique : Les diagrammes, les réseaux, les cartes*. Edition Gauthier-Villars, 1967.

[58] M. Jenkins, "Using and programming analog," in *Analog Synthesizers*, Focal Press pp. 142-207, 2009.

[59] L. Vickery, "Notational Semantics in Music Visualisation and Notation," in *Proceedings of the Australasian Computer Music Conference (ACMC'14)*, Melbourne, Australia, pp. 101-112, 2014.

[60] B. Nettl, *The Study of Ethnomusicology: Thirty-Three Discussions*. University of Illinois Press, 2015.

[61] H. Schippers, *Facing the Music: Shaping Music Education from a Global Perspective*. Oxford University Press, 2009.

[62] R. Rusch, K. Salley, and C. Stover, "Capturing the Ineffable: Three Transcriptions of a Jazz Solo by Sonny Rollins". *Music Theory Online*, vol. 22, no. 3, 2016.

[63] D. Temperley, "Syncopation in rock: a perceptual perspective", *Popular Music*, vol. 18, no. 1, pp. 19-40, 1999.

[64] R. Donington, *The Interpretation of Early Music*, Faber & Faber, 1963.

[65] P. Antoniadis and F. Bevilacqua, "Processing of symbolic music notation via multimodal performance data: Brian Ferneyhough 's Lemma-Icon-Epigram for solo piano, phase 1", in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, UK, pp. 127-136, 2016.

[66] J. Kojs, "Notating Action-Based Music", *Leonardo Music Journal*, vol. 21, pp. 65-72, 2011.

[67] S. Emmerson, "Crossing cultural boundaries through technology?", in *Music, Electronic Media and Culture*, pp. 115-137, 2000.

[68] G. W. Furnas and B. B. Bederson, "Space-Scale Diagrams: Understanding Multiscale Interfaces," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, Colorado, USA, pp. 234-241, 1995.

REMOVING THE IMAGINARY BOUNDARY BETWEEN SCORE AND WORK: INTERACTIVE GEOMETRICAL NOTATION

Jason Noble
McGill University
jason.noble@mcgill.ca

ABSTRACT

In many notational practices in late 20th- and early 21st-century music, the score has a visual artistry all its own. Nevertheless, even heavily graphical *Augenmusik* scores are often experienced only by the composer and performer, and are not part of the audience's visual experience of performance. Because elements from non-auditory modalities (especially visual) seem essential to many musical works, I argue for a multimodal understanding of such pieces, removing the imaginary boundary between score and work. I discuss a type of aleatoric, flowchart-like geometrical notation that I frequently use in my own compositions, using hybrid notation combining standard musical notation with geometrical forms. This kind of notation helps clarify the analogy between visual and auditory modalities. In my piece *simple geometries*, I integrate geometrical notation into performance with the projection of an interactive, animated score that uses movement and changes of zoom perspective to make the logic of the work's open form accessible to the audience.

1. SCORE AND WORK: *A FRONTIÈRE IMAGINAIRE*

The traditional model of production in Western art music keeps the composer at a mysterious distance: neither she herself nor the object that she directly produces—the score—is typically encountered by the audience during performance [1]. Although the composer is considered the "author" of the work, the most immediate fruit of her labour is taken to be curiously external to the work itself (except in the score's heuristic role of teaching the performer how to mediate the work to the audience). The composer is a kind of shadow-puppeteer, the contortions of whose hands are valued as a means to the end of the projected shadow-image but not as aesthetic ends in themselves. It matters what the score looks like, but only insofar as its appearance affects its clarity in instructing the performer, who in turn delivers the work to the audience through the medium of sound. That the audience does not see the score in performance is assumed not to impoverish their experience of the work, for a successful performance will have transmitted through sound everything essential about the work. The underlying principle is that music exists within the singular modality of the auditory domain, and the score—while necessary as a vehicle for the creation or transmission of the work—is fundamentally distinct from the work itself: a subservient, pragmatic entity that is aesthetically inessential. Gérard Grisey expressed such a conception by comparing the score to "*the map*" and musical sound as "*the lie of the land*" [2], as did Brian Ferneyhough in stating that the adequacy of musical notation—which occupies "a strange ontological position: a sign constellation referring directly to a further such constellation of a completely different perceptual order"—is determined by its efficacy as a method of specifying sounds [3]. Some authors have distinguished between "descriptive" notation, which conveys information about musical sound, and "prescriptive" notation, which conveys information about methods of sound production [4], but both notational concepts ultimately assume sound to be the essence and *telos* of the musical work, with the score serving a supporting, didactic role.

The putative ontological divide between score and work is inconsistent with the practices of some composers in the late 20th and early 21st centuries. The shapes and symbols in scores such as Cornelius Cardew's *Treatise* (1963-67) and George Crumb's *Makrokosmos* (1972-79) are clearly artistic elements in and of themselves, not aesthetically inert instructions for sound production. These practices find precedents in the works of much earlier composers such as Baude Cordier (1380-1440; Figures 1 & 2).

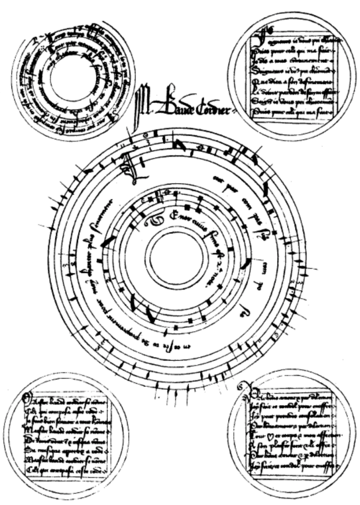


Figure 1. B. Cordier, *Tout par compas suy composes*.

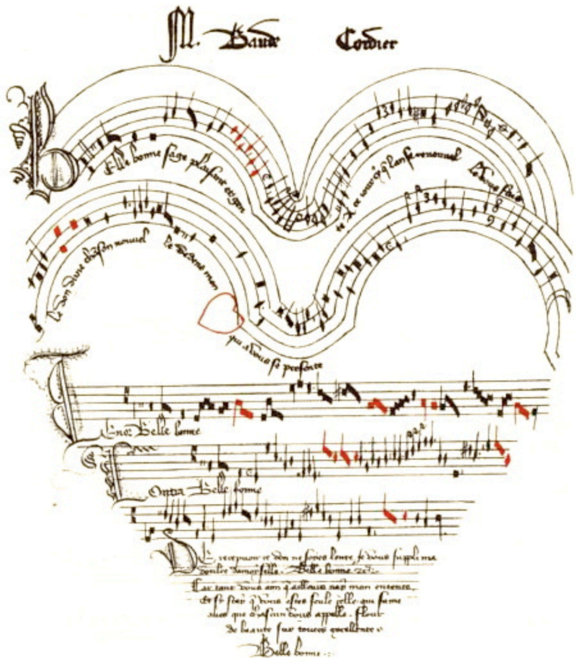



Figure 2. B. Cordier, *Belle, Bonne, Sage.*

Jason Freeman considers "concrete" scores such as these to "visually depict programmatic elements in the music through novel graphic design" [1]. Their visual appearance conveys information, ideas, and aesthetic effects difficult or impossible to infer from sound alone, and as such one could argue that an experience of these works that does not involve seeing the score is incomplete. They force us to either dismiss the visual elements as inessential to the musical work on the grounds that they do not reside in the auditory modality, or to adopt a multimodal concept of musical works that no longer assumes that everything essential is transmitted through sound. I would like to make a case for the latter position.

I contend that the score is not—or at least does not have to be—merely an elaborate sonic recipe, an externality in the service of the work proper. As Ferneyhough says, notation is "an explicit ideological vehicle (whether intended as such or not from the point of view of the composer)" [3] (pp. 2-3). The appearance of the score can be an essential artistic constituent of the work, an aesthetic deliverance in its own right, a symbiotic visual counterpart that can clarify, recontextualize, enrich, and reinforce the concepts presented through musical sound. A parallel situation is seen in concrete poetry, in which the visual layout of the words makes a distinct aesthetic contribution that complements the poem's linguistic meaning. A familiar example is "The Mouse's Tale," from Lewis Carroll's *Alice's Adventures in Wonderland* (Figure 3). In his piece *Adventures Underground* (1971-77), which sets this poem of Carroll's, David del Tredici creates a similar "*Augenmusik*" effect in his score [5].

In his Oxford Music Online entry, Thurston Dart describes *Augenmusik* ("Eye Music") as "[m]usical notation with a symbolic meaning that is apparent to the eye but not to the ear," and stipulates that "[s]ince its effects are derived from notation it is the concern of composers and performers rather than listeners" [6]. Dart goes on to

The Mouse's Tale

"FURY said to
a mouse, That
he met in the
house, 'Let
us both go
to law: *I*
will prose-
cute *you*.—
Come, I'll
take no de-
nial: We
must have
the trial;
For really
this morn-
ing I've
nothing
to do.'
Said the
mouse to
the cur,
'Such a
trial, dear
sir, With
no jury
or judge,
would
be wast-
ing our
breath.'
'I'll be
judge,
I'll be
jury,'
said
cun-
ning
old
Fury;

Figure 3. L. Carroll, "The Mouse's Tale" from *Alice's Adventures in Wonderland* (1865).

distinguish two simultaneous interpretations derived by performers of *Augenmusik*: one symbolic and the other "purely musical." Similarly, David Kim-Boyle acknowledges that "graphic scores often have a visual appeal that goes beyond a purely musical function" [7]. These authors address the multimodal nature of such works by distinguishing their musical functions from other functions. I want to offer another reading, interpreting the musical work as *inherently multimodal*. On this reading, works such as those listed above are not "purely" musical structures onto which inessential signs from the visual modality are appended, but rather are musical structure that are multimodal in their very conception.

When the score's visual appearance "becomes of primary formal importance" and is invested with "aesthetic and musical significance" [7], the question of what in the work is "purely" musical and what is not becomes academic. Also, as many have noted, the notion of musical "purity" is deeply suspect. Responding to Peter Kivy's concept of "music alone," Nicholas Cook states that "music never *is* 'alone' ... it is always received in a discursive context ... it is through the interaction of music and interpreter, text and context, that meaning is constructed" [8]. Cook describes how musical meaning arises through interpretive mappings between musical and extramusical domains. I want to suggest that such mappings may also operate *within* musical works (conceived multimodally). Similar to mappings between musical gestures and physical gestures, which invite what Arnie Cox describes as "mimetic motor imagery" and "mimetic motor action" as sources of embodied engagement in musical experience [9], visual structures in musical notation and auditory structures in musical sound may also invite homologous mappings that yield satisfying ways to engage with the work. For musically literate musicians accustomed to score-reading, this is a truism: even after hearing a work performed, musicians often feel their understanding of a piece is incomplete until they have seen its score, as the score's visual presentation of musical information vastly enriches their understanding of the musical work. Composers spend untold hours obsessing over the visual appearance of their scores—even scores that are not "graphical" in the conventional sense—and there is little doubt that the experience of reading scores is often an aesthetic visual experience for them (distinct from and complementary to aesthetic experience of hearing with the "mind's ear"). Nevertheless, visual charms of notation are conventionally assumed not to belong to the music proper, perhaps at most providing extramusical decorative addenda.

Works of *Augenmusik* make more explicit than standard musical notation the impetus to recognize the distinct and complementary contributions of different modalities as equally essential, to overcome the *frontière imaginaire* between score and work. My suggestion is that the modal divide between score and work is ideological and not ontological, and that there may be good reasons to reconceptualize the musical work to include multimodal components. By offering visual elements that require little or no special training to understand (unlike conventional musical scores, which require an idiosyncratic literacy), *Augenmusik* scores make illuminating and enriching visual experiences accessible to nonmusicians and musicians alike, "stimulated by a desire to realize broader social and political ideals of engagement" [7]. Frequently these visual experiences reveal important conceptual and aesthetic aspects of the work, as well as privileged insights into the work's structures and functions that would be lost on many listeners (including many musicians) in "monomodal" listening situations. Perhaps the satisfying act of recognizing relations between notation and sound may parallel the satisfaction of similar cross-domain mappings in concrete poetry, word painting, and incidental music for film, theatre, and ballet. Perhaps the contextualizing visual complement provided by graphical scores may provide an entry point to audiences unfamiliar with contemporary music, and may thereby broaden the reach of our art.

Incentives such as these may have become apparent to contemporary composers, as many have begun to explore innovative, integrative, multimodal practices that unite visual elements of notation with the deployment of sounds in time. Such works frequently incorporate "liveness," with score and performance co-evolve continuously in real-time [10], inviting a heightened sense of engagement in a responsive, real-time interaction [11]. In this kind of "live" context, dynamic relations between score and performance become a major source of aesthetic interest for composers, performers, and audiences alike: "[n]otation becomes a vehicle for expressing the uniqueness of each performance of a work rather than a document for capturing the commonalities of every performance of that work" [1]. In such scenarios it seems unintuitive to conceive of the score as an antecedent or externality to the work: musical sound is neither conceived nor encountered as an isolated entity, and the symbiotic, unfolding interaction between notation and sound is a an aesthetic end in itself, not a mere means to the realization of a "purely" musical (*qua* sonic) work. In this spirit, my own artistic work has gravitated towards multimodal integration and diminished boundaries between score and work, as discussed below.

2. GEOMETRICAL NOTATION

I make frequent use of geometric, flowchart-like aleatoric notation which provides an intuitive and visually pleasing vehicle for musical expression. The use of geometrical forms in musical composition and notation has many precedents. As noted above, George Crumb made use of circular and spiral forms in some of his scores. Iannis Xenakis used geometrical forms as the basis for both architectural design and musical composition, in some cases using the very same forms for both purposes: a famous example is his translation of the contour lines from the Philips Pavilion, which he designed with architect Le Corbusier, into glissando lines in *Metastaseis* (1953-54). Barry Truax's work *Riverrun* (1986), realized entirely with real-time granular synthesis, has no score in the traditional sense but deploys very brief sound events ("grains") according to complex geometrical distribution that is revealed by spectrographic analysis (Figure 4).

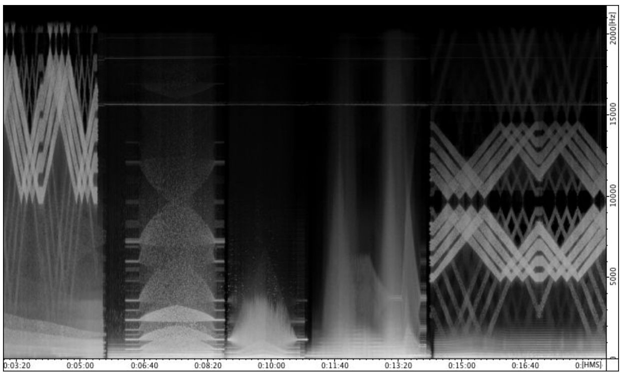


Figure 4. B. Truax, spectrograph of *Riverrun* (1986). Produced with permission.

Indeed, Truax's compositional process involved what he called "tendency masks," stochastic distributions of sonic parameters controlled with programmed geometrical shapes [12].

I find geometrical notation appealing for several reasons: it is elegant in its simplicity; it reveals the logic of some kinds of musical patterning in a straightforward, readily comprehensible way; and it supports mappings to a variety of cross-modal and extramusical domains, via the shared image-schemata of geometrical reasoning. My first experiments with geometrical notation followed from considerations of how to effectively notate the aleatoric deployment of defined sets of musical elements or values to performers. A standard solution is to do this with musical elements notated on a single staff with prescriptive textual instructions such as "play in any order" (e.g., Figure 5).

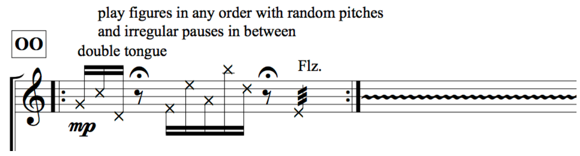


Figure 5. J. Noble, excerpt from *The Sphinx and the Garden Gnome* (2015).

But I found this kind of representation unsatisfactory because of the cognitive dissonance between the linear representation of elements on the staff and their non-linear deployment, and because of the reliance on textual instructions to convey ideas that should be diagrammatically communicable. Furthermore, for readers conventional Western music notation (as well as English and most Western languages), there is a strong learned tendency to read from left to right, and as a result performers tend to unintentionally favour left-to-right orderings between consecutive elements, skewing the distribution of the sounding result. Geometrical notation provides a more suitable visual representation of the desired distribution, positioning each element at a vertex on a geometrical figure and using unidirectional or bidirectional arrows to indicate the possible pathways (e.g., Figure 6).

Geometrical notation is versatile: the number and type(s) of elements in a given network may be chosen freely by the composer. In some cases the elements I have used are single notes or sound events, while in others they are longer sequences, such as melodies in the folksong pastiche in *One Foot in the Past* (2016; Figure 7).

Superposing multiple layers of carefully selected but indeterminately distributed elements creates a generative situation in which random coincidences of events produce emergent harmonic, rhythmic, and textural properties that come to temporary perceptual prominence and then dissolve. The characteristics of these emergent properties depend greatly on the constituent elements that make up the musical layers—whether they are timbrally homogeneous or heterogeneous, whether or not they are structured metrically or periodically, what potential intervallic relationships exist within their referential pitch structures (if any), etc. Different textural roles may be assigned to different

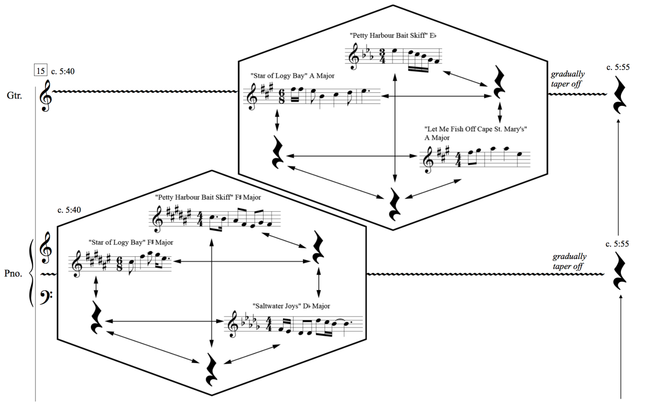


Figure 7. J. Noble, excerpt from *One Foot in the Past* (2016).

musical layers, with varying degrees of linearity (e.g. melodic content), periodicity, harmonic complexity, and so forth (e.g., Figure 8). It is also possible to alternate linear sections (using conventional notation or a close approximation thereof) with distributional sections (using geometrical notation), and/or to superpose linear and distributional sections in different orchestrational layers.

An attractive aspect of geometrical notation is that the formal organization of musical materials is not concealed beneath a linear realization (as is frequently the case in combinatorial music, for instance), but is rather laid bare on the surface of the score. Of course, any given performance takes a linear form as sound events are realized sequentially in time, and these could theoretically be notated more-or-less conventionally. However, geometrical notation makes clear that no particular realization is prioritized over any other: an indefinite number of potential combinations exists within the distributional networks, and a great deal of the fascination of the music comes from the coincidentally emergent properties of random samples of those combinations as they unfold indeterminately. Even viewers not equipped with the musical literacy to make sense of the content of the musical elements can still appreciate the multiplicity of possible pathways through the networks, as well as the visual beauty of simple geometrical forms.

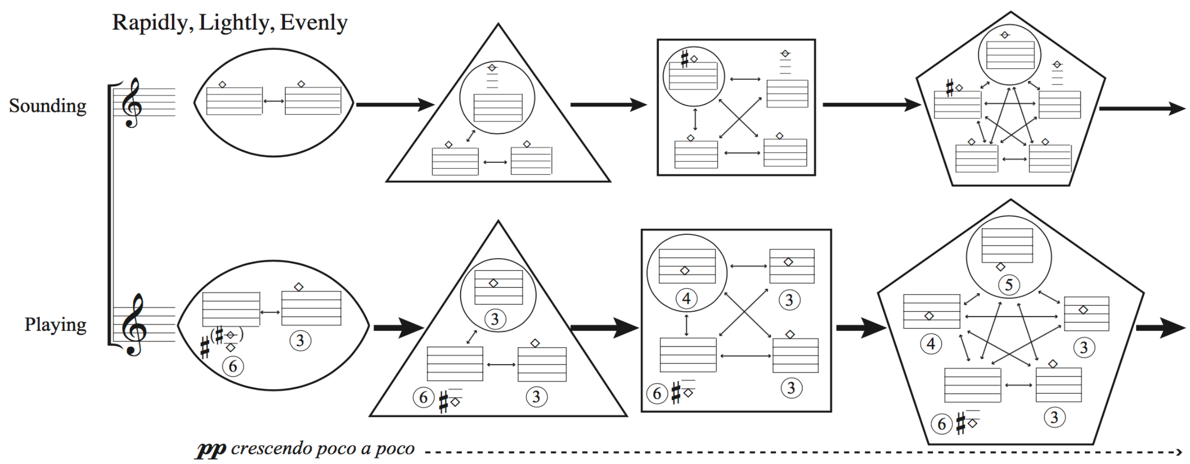


Figure 6. J. Noble, excerpt from *Shadow Prism* (2015).

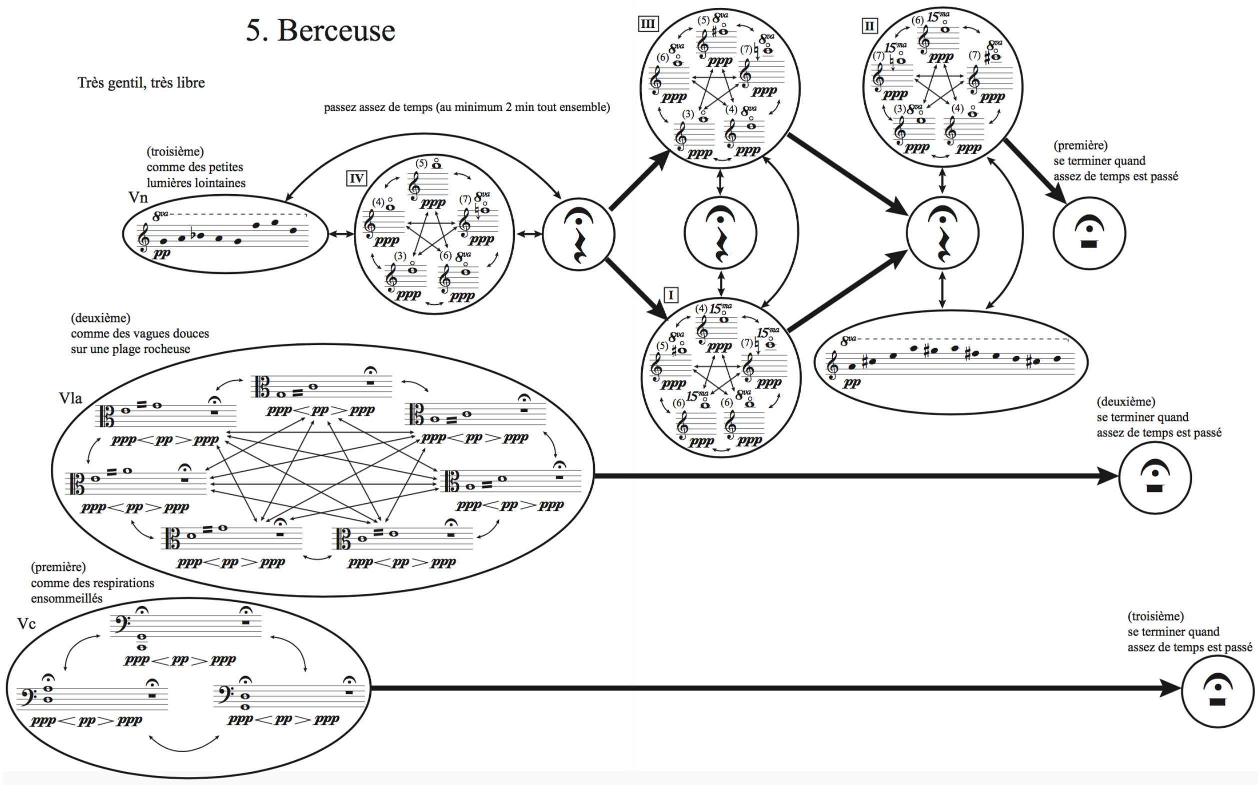


Figure 8. J. Noble, "5. Berceuse" from *Bathurst Suite* (2016).

Although the above-listed examples are multimodal in conception (and likely to be experienced as such by performers), they do not yet directly address the problem articulated in the first section of this paper: how can graphical notation be incorporated into performance, making the artistic visual qualities and their meaningful relations with musical sound available to audiences? My first serious attempt to answer this question was in *simple geometries* (2017) for cello, electronics, illuminated glass harp, and video projection.[1]

3. SIMPLE GEOMETRIES (2017)

simple geometries consists of seven musical layers (I – VII) organized approximately concentrically, through which the performer moves according to a bidirectional spiral pattern dictated by common elements between consecutive layers (indicated in the score with large two-headed arrows connecting the common elements; see Figure 9). The pitch content of the piece consists entirely of open strings and natural harmonics (which themselves follow a simple geometrical pattern dictated by simple numerical ratios), along with indeterminate pitches provided by idiophonic accessory instruments.

Layer I includes a singing bowl or very large crystal glass, layer II includes two large crystal glasses, layer III includes three medium crystal glasses, and layer IV includes four small crystal glasses. In layers I–IV, the specified pitches and accessory instruments for each layer may be played in any order; as such, there are no arrows within the dotted rings delineating these layers. However, transitions between layers must take place by way of shared elements indicated with dashed boxes and large bidirectional arrows (a kind of "common-tone modulation"). The idiophonic instruments are physically arranged on a table in front of the cellist in a spiral pattern, with the singing bowl (or very large crystal glass) in the centre. Each crystal glass is illuminated from below by LEDs activated by contact microphones when the glass vibrates.

Layers V–VII are executed only on the cello, and consist of cyclical ordered sequences containing 5 – 7 phrases, respectively, notated in ring patterns. The patterns are modelled on simple geometrical patterns: sinusoidal waves in V, sawtooth patterns in VI, and exponential expansion in VII. These same patterns provide models for suggested paths through the score, represented in the form diagram in the top left of the score. Durations and rhythmic patterns of elements are improvised within approximate ranges defined in the legend in the bottom left of the score, with the longest durations in layer I and the shortest in layer VII. Additionally, articulations, rest durations, bow placements, and contours are specified for the layers, and the performer may freely choose values within the given ranges for each of these parameters.

It is important to emphasize that the geometrical patterns in the score (spirals, concentric circles, polygons,

[1] A video recording of this piece may be viewed at: https://www.youtube.com/watch?v=rin-zdcgEjo

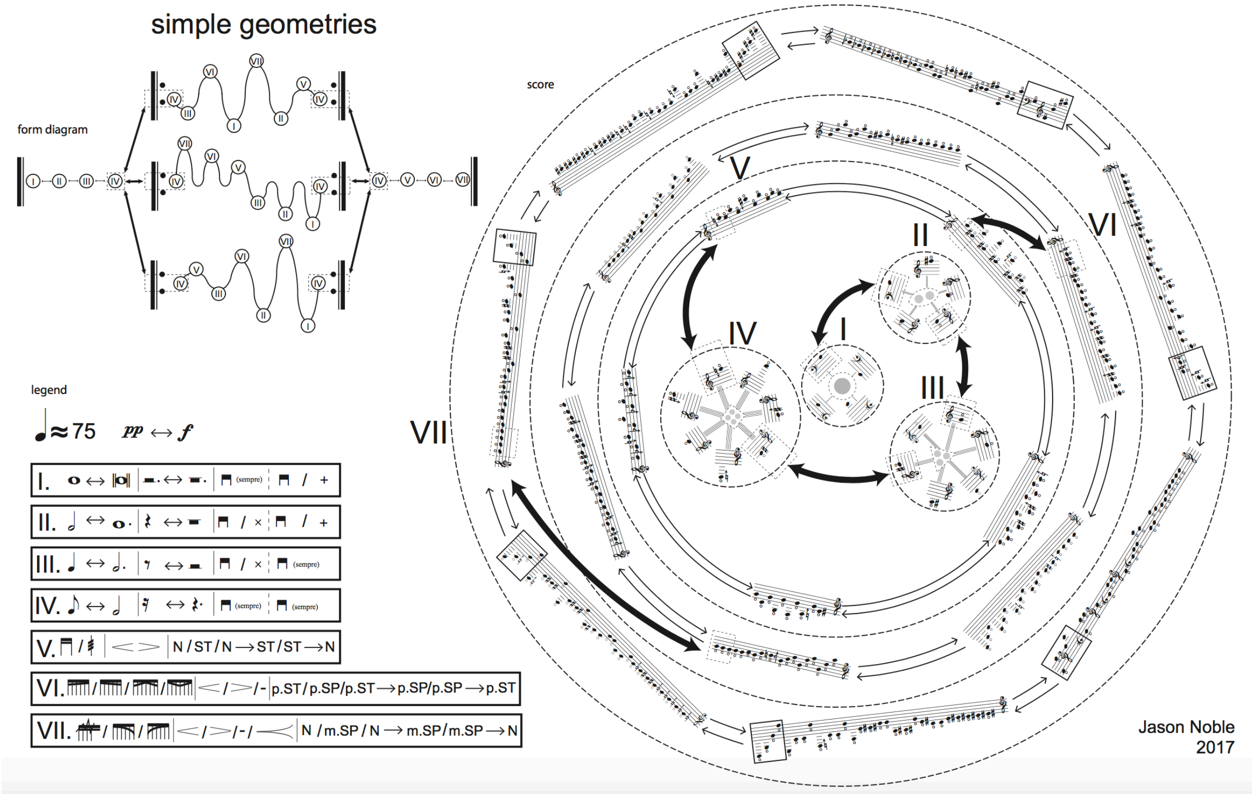


Figure 9. J. Noble, score of *simple geometries* (2017).

simple waveforms) directly reflect the piece's musical organization. Subsequent pages of the score provide sample realizations of each layer in standard, linear notation, but these are heuristic devices only and are far less adept at representing the work's musical logic. To make the multimodal conception of the piece explicit to the listener, an adapted version of the score (realized with MaxMSP/Jitter) is projected on a screen behind the performer (Figure 10).

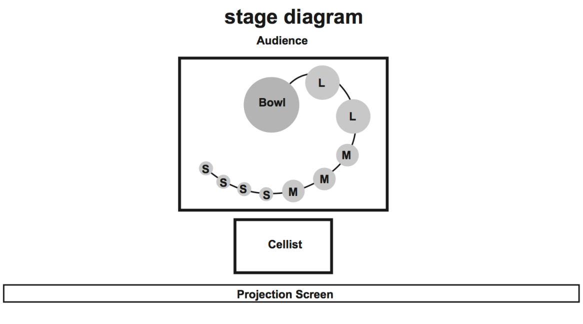


Figure 10. J. Noble, stage layout of *simple geometries* (2017).

At the beginning of the performance, all seven layers may be seen, three-dimensionally organized so that layer VII is closest along the z axis and layer I is farthest away (Figure 11).

When the performer plays a given layer, he uses a foot switch to zoom to that layer in the projection. A second switch may be used to trigger playback of a pre-recorded or live-captured sound files for that layer. When a given layer is sounding in the electronics, its associated notational layer moves in the projection: layers I – IV undulate irregularly, and layers V – VII rotate in the direction the performer chooses to play (following the ring sequence either clockwise or counterclockwise).

There are many possible paths through the layers of the score (including the suggested routes form diagram, as noted above). Within each layer, there may be considerable variation as per the free choices of the performer within the specified musical values. When layers are superimposed with the playback of sound files, the possibilities for variation are greatly multiplied as the chosen combinations, phase alignment, and (if live-captured recordings are used) the sonic content of the layers will also vary with each performance; the emergent properties of the music will vary commensurately. Although the specific

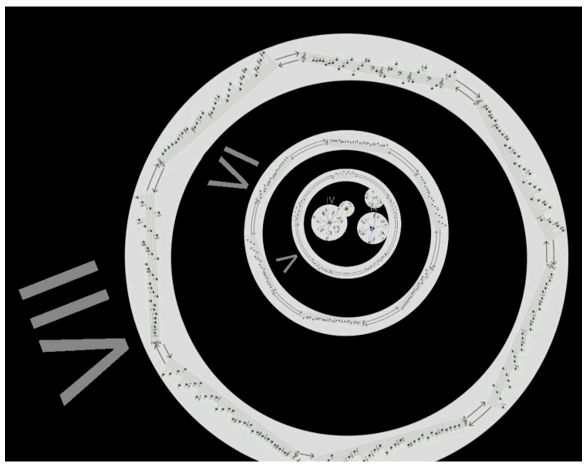


Figure 11. J. Noble, first projection image of *simple geometries* (2017).

configurations that emerge in each performance are subject to tremendous indeterminate variation, harmonic coherence is guaranteed by the derivation of all of the cello's pitch material from open strings and natural harmonics, and transitory pulses and metres are guaranteed to emerge from the periodic rhythmic organization of layers V – VII. It is likely that sound alone would fail to convey the relatively simple, layered organization of the piece, as superposing more than two or three layers at a time would likely overwhelm the listener's ability to perceive them as distinct strata (especially given the timbral and harmonic continuity between them). But the visual appearance of the score, especially when animated by motion corresponding to the activation of layers, makes the musical organization much clearer. The sounds and the dynamic projected score are equally important to the aesthetic of the piece, and it is only when both modalities, and the analogies between them, are perceived that the work is complete.

4. CONCLUSION

In this paper, I have argued for a multimodal conception of musical works that includes not only sound but also manifestations in other modalities (focusing here on the visual). This is consistent with practices such as *Augenmusik* and word painting, as well as theories such as embodied cognition and cognitive semiotics. It is inconsistent with the "music alone" ideology of formalism and "absolute" music. I believe that that ideology's manifestation in the Western concert tradition, in which musical sound is isolated as much as possible from other sign structures while audience members are expected to devote their undivided attention to the auditory modality, is a major contributing factor in the perennial alienation of popular audiences from contemporary music. Presenting audiences with multimodal experiences of works by integrating accessible features of notation into performance may help engage broader audiences in contemporary music. Graphical scores—already so visually and symbolically meaningful for composers and performers—represent an opportunity to reimagine performance practice in ways that overcome the imaginary boundary between score and work. My piece *simple geometries* attempts to do this by integrating the geometrical conception at the heart of the piece into the layout of its score, the pitch content and gestures of its musical materials, the physical setup of its performance forces, and the animated projection of its score.

5. REFERENCES

[1] J. Freeman, "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Computer Music Journal,* vol. 32, no. 3, 2008.

[2] G. Grisey, "Tempus ex Machina: A composer's reflections on musical time," *Contemporary Music Review*, vol. 2, no. 1, 1987.

[3] B. Ferneyhough, *Collected Writings.* Harwood Academic Publishers, 1995.

[4] M. Kanno, "Prescriptive Notation: Limits and Challenges," *Contemporary Music Review*, vol. 26, no. 2, 2007.

[5] F. J. Oteri, "The Alice Pieces of David del Tredici," 2018. [Online]. Available: http://www.daviddeltredici.com/alice/

[6] T. Dart, "Eye music (Ger. *Augenmusik*)," in *Grove Music Online*. [Online]. Available: http://www.oxfordmusiconline.com/grovemusic/.

[7] D. Kim-Boyle, "Real-time Score Genration for Extensible Open Forms," *Contemporary Music Review,* vol. 29, no. 1, 2010.

[8] N. Cook, "Theorizing Musical Meaning," *Music Theory Spectrum*, vol. 23, no. 2, 2001.

[9] A. Cox, *Music & Embodied Cognition: Listening, Moving, Feeling, & Thinking*. Indiana University Press, 2016.

[10] C. Nash and A. Blackwell, "Flow of Creative Interaction with Digital Music Notations," in K. Collins, B. Kapralos, and H. Tessler (Eds.): *The Oxford Handbook of Interactive Audio*. Oxford University Press, 2014.

[11] P. Auslander, "Digital Liveness: A Historico-Philosophical Perspective," *PAJ: A Journal of Performance and Art*, vol. 34, no. 3, 2012.

[12] B. Truax, "Composing with Real-Time Granular Sound," *Perspectives of New Music*, vol. 28, no. 2, 1990.

REFRAMING THE LISTENING EXPERIENCE THROUGH THE PROJECTED SCORE

David Kim-Boyle
The University of Sydney
david.kim-boyle@sydney.edu.au

ABSTRACT

Over the past ten years, performance scores have been radically foregrounded in a variety of performance practices. Whether such notations assume a prescriptive function, visually projected for musicians to interpret, or a descriptive one, unfolding as a documentation of a live coding performance, how might such a foregrounding reframe the listening process for an audience? Does a notational schema help promote a deeper, structural level understanding of a musical work? This paper will consider these various questions, exploring how principles of graphic design and the transparency of notation contribute to the listening experience. It will suggest that works featuring projected scores find aesthetic value in the juxtaposition of notation's traditionally mnemonic function and the unique temporal modalities that projected scores establish.

A full version of this paper appears in TEMPO, Vol. 72 – Issue 284, pp. 37-50 (ISSN: 0040-2982). Permission to abridge and include in these conference proceedings with kind permission of TEMPO Editor, Christopher Fox.

1. INTRODUCTION

From the perspective of the listener, the radical experiments with notational schemas in the 1950s and 1960s by composers such as Wolff, Brown, Stockhausen, Haubenstock-Ramati, Cardew, and Cage, typically remained in the background, only ever manifest in an aural space. In sharpa contrast, in a growing body of contemporary performance practices, the score has been radically foregrounded, displayed for an audience and offering not only an enriched aesthetic experience, but an opportunity for listeners to develop a deeper understanding of the processes and structures underlying a musical work or performance.

Like their traditional print-media counterparts, projected scores showcase a diverse range of approaches to the use of notation. They often feature information which is dynamically updated or transformed during a live performance, and many also integrate non-linear processes within these generative processes as in Nicolas Collins's *Roomtone Variations* (2013), for ensemble, or Jason Freeman's *Shadows* (2015) for piano and computer [1].

Projected scores need not adopt common practice notation,[1] nor do they necessarily need to be generated by computer. Jobina Tinnemans' *panoramic scores* [2], for example, feature hand drawn graphic notation presented on printed media spanning an entire performance space. In her *Imagiro Landmannalaugar* (2017) for small ensemble, for example, the score spans over twenty-metres in length, requiring performers to physically navigate through the performance space as they read the score, see Figure 1.

Figure 1. Still image from a performance of Jobina Tinnemans' *Imagiro Landmannalaugar* (2017). Image included by kind permission of Jobina Tinnemans.

Projected scores also need not be prescriptive in nature. Live coding performances, for example, often routinely display programming script edited by performers in real-time which outline the processes, albeit in highly coded form, that shape a musical structure. But even live coding performances need not feature programming script [3].

Irrespective of the type of notational schema projected or the motivation for projecting it,[2] the overt display of the score reframes the listening experience in distinctly unique ways. Does such a foregrounding necessarily promote a deeper structural understanding of a musical work or underlying performance processes? Might not the inherent decoding process inhibit such an understanding? How might the visual design or temporal modality of a dynamic score support this understanding? To better address these questions, a useful starting point is to consider how the visual design constraints of scores created on screens and intended for projection affect the ways in which composers articulate musical forms.

[1] Common practice notation is arguably used far less often than other forms of notation in this practice.

[2] While these may indeed include a desire to provide listeners with a deeper understanding of underlying musical processes, they may also be driven by a response to pragmatic challenges involved in presenting screen scores to small ensembles or simply an appeal to visual aesthetics.

2. DESIGN PRINCIPLES AND CONSTRAINTS

Projected scores are uniquely bound by several principles of visual design which frame the way in which performers and listeners engage with the work and musical processes they denote [4]. For those scores which are prescriptive in nature, these principles in turn facilitate certain modes of musical expression while inhibiting others.

Despite the obvious advantages of common practice notation, not least of which is its widespread familiarity, its informational density makes it not particularly well suited for visual projection, one of the reasons perhaps why graphic notation is often used in this practice. This problem is further exacerbated when multiple parts are embedded within a page. As a result, when common practice notation is projected for performers to interpret, it tends to operate within unique constraints – rhythmic complexity is avoided, pitch selections are often confined to smaller registral tessituras, and traditional expressive indications whether denoted by symbols or text are minimized.

While page turns are somewhat of an anachronism in projected scores, the constrained spatial area of a display has seen composers adopt a range of animation techniques in order to present performers with new musical information [5]. Cat Hope's screen scores, for example, often employ scrolling techniques to display new information to an ensemble, directly correlating the display methodology to the drone-based forms that underscore much of her work.[3]

New musical information can also be embedded within a single display through the animation of notational descriptors. In Bergrún Snæbjörnsdóttir's *Esoteric Mass* (2014) for sixteen wind instruments, for example, notes are denoted by circles of light which orbit along concentric rings projected onto the floor of the performance space around which the performers stand, see Figure 2.

Animated event descriptors can also be combined with traditional notation in a hybrid form. In Ryan Ross Smith's *Study No. 10* (2013) and Ingibjörg Fríðriksdóttir's *Right is Wrong* (2013), both for solo piano, only one grand stave is displayed, addressing the information density weakness of common practice notation, with discrete pitches scrolled across the display from right-to-left.

Irrespective of the type of animation adopted, the speed of dynamic change is constrained by the inability of the eye to accurately track rapid visual transitions, especially when that information is distributed over a large spatial area [6]. Visual information is rarely animated at a speed greater than that which it can be accurately tracked by the performers unless the failure of accurate tracking happens to be of aesthetic importance, as in the case of a work such as Lindsay Vickery's *Escadaria do Diablo* (2017) where the performer faces the challenge of reading a score in which notation randomly disappears.

[3] The types of animation techniques employed in a screen score often underscore a work's formal structure. Consider, for example, how performers might approach a performance of Hope's *Longing* should a "pages" methodology for displaying new information be used or how the event-driven textures of Ryan Ross Smith's various percussion works are related to temporal synchronicities and collisions between on-screen graphic primitives.

Figure 2. Still image taken by Henrik Beck/nyMusikk from a performance at nyMusikk's Only Connect festival of Bergrún Snæbjörnsdóttir's *Esoteric Mass* (2014). Image included by kind permission of Bergrún Snæbjörnsdóttir and Henrik Beck/nyMusikk.

Color assumes a more constructive role in scores generated by computer and projected in performance.[4] It can be used to help distinguish different parts within a work for ensemble, as seen in Cat Hope's *Longing*, or mark different dynamic levels of individual notes as in Ingibjörg Fríðriksdóttir's *Right is Wrong* for solo piano, or facilitate editing of live coding script. Alongside purely functional roles, color sometimes has an undeniable aesthetic importance in scores designed for projection. In Marina Rosenfeld's *WHITE LINES* (2003-ongoing), for example, a pair of parallel white lines are superimposed on a series of short color video projections. The lines vary in width and opacity with performers mapping those variations to musical parameters. While color certainly has a functional role in helping distinguish the white lines from the background image, it also has a fundamental aesthetic value in drawing attention to concepts of stasis and becoming.[5]

The musical processes denoted by the projected score are clearly conditioned by these and various other principles of visual design and organisation. And while the ability of the projected score to contribute to a deep structural understanding of a work may be open to conjecture [7, 8, 9], the foregrounded score nevertheless invites the listener to enter into a decoding process to support a better understanding of the musical and performance processes underlying the work itself.[6]

3. DECODING

> *...to listen is to adopt an attitude of decoding what is obscure, blurred, or mute, in order to make available to consciousness the "underside" of meaning...* [11]

In his influential 1986 essay "Listening" Roland Barthes identified three ways in which sound can provide mean-

[4] This is not to suggest that color has not been used in paper-based scores, refer for example to the use of color in the 14th century *Ars Subtilior* as a means of clarifying complex mensural division.
[5] Personal communication with the composer.
[6] In effect, a reversed type of *synchretic* listening where the image provides insight into an aural space, see [10].

ing – firstly through acting as an *indice* and thus providing a means of orientation, secondly through acting as a *sign* and functioning in a semiotic mode, and thirdly through functioning as a *shimmering of signifiers* that draws attention to what is unsaid. Barthes associates the third mode of listening with that of the experience of listening to the work of experimental composers such as John Cage where awareness is brought to the verticality of sound rather than its *syntagmatic extension*. While in many respects Barthes modes are woefully general,[7] they do provide a useful framework for helping to understand the experience of listening to musical works the scores of which are visible to the audience.

Through foregrounding the score, listeners are invited to engage in a deciphering process to help understand the musical processes to which they are attending. In Rosenfeld's *WHITE LINES*, this deciphering is even encouraged when the notation is exhibited in non-concert settings.[8] All this despite Barthes assertion that we do not listen to music in a deciphering sense.

Referential functions are made somewhat easier to decode through the use of animation techniques in certain generative scores to denote the onset of particular note events. In Bergrún Snæbjörnsdóttir's *Esoteric Mass* or many of Ryan Ross Smith's works, it is not difficult for the listener to perceive that the collision of graphic primitives or the intersection of moving circles with the spatial location of performers, denote the articulation of discrete note events. Similarly in scrolling scores which employ a *playhead* paradigm, the relationship of graphic shapes to relative pitch is easily decoded through observance of the vertical point of intersection of the shape with the playhead. In each of these modes, the referential functions of the notation employed are facilitated through the manner of their temporal unfolding.

Somewhat counterintuitively, perhaps, the referential function of notation can also be suggested through an *a priori* physical relationship between the performer and the visually presented score. This relationship is at the core of Snæbjörnsdóttir's *Esoteric Mass*, where the score is physically embodied within the performance space, but it is also explored in Jobina Tinnemans' *Imagiro Landmannalaugar* (2017), see Figure 1, where the decoding process is facilitated through the manner in which the performers choreograph their movement through the performance space in order to be able to read the twenty-four metre long score.

It does not necessarily follow that simply understanding a referential code [13] or syntactic structure of a notational schema allows a listener to more easily draw associative relationships across sensory modalities. This is particularly the case when various non-linear processes are embedded within a musical form or when notational schemas begin to assume a more poetic function [14]. Indeed, as notational schemas become more complex, their various referential functions become more ambiguous and difficult to decode. In Lindsay Vickery's *nature forms I* (2014) for three instruments and electronics, see Figure 3, for example, it is unlikely that the listener will be able to ascribe any referential function to the notation as these functions themselves are not semantically disjoint, with each of the three players interpreting the notation according to different rules. Clearly, in such a work, the poetic function of the notational schema assumes as much importance as any referential one. Nelson Goodman goes even further by claiming that a variable compliance relationship such as this fails to meet the semantic requirements of a notational schema, i.e. it is not semantically disjoint, and can therefore no longer be considered to be a notation at all [15].

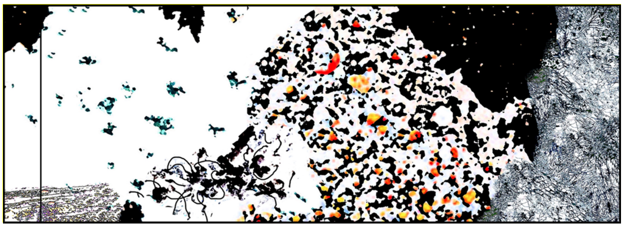


Figure 3. Excerpt from the score for Lindsay Vickery's *nature forms I* (2014). Image included by kind permission of Lindsay Vickery.

Despite the inherent difficulties inherent in the decoding process, it does not necessarily follow that the inability to unambiguously ascertain referential relationships between image and sound prevents the listener from developing a deeper understanding of a musical work just as it does not necessarily follow that someone who can fluently read common practice notation automatically has a deep understanding of traditionally notated works. Somewhat ironically perhaps, this supports Barthes original assertion that we do not listen to music by way of deciphering [11], despite the overt invitation to do so through the foregrounded score. While Barthes argues for a vertical signifying in his third mode of listening, which he contends is the manner of listening encouraged by the contemporary art music tradition of the early 1970s, he does not explore in great depth the temporality of the listening process. The author would argue that the temporal modality of scores foregrounded through projection present perhaps the most interesting insights on how composers working in this area of practice frame listener engagement with the work [4].

4. TEMPORAL MODALITY

The projected score encourages an engagement with procedural relationships as they temporally unfold in the score and are musically sounded in the performance space.[9] While this engagement is to a certain extent more easily recognised in those scores which employ various animation techniques, it is also strongly featured in those scores such as Tinnemans' *Imagiro Landmannalaugar* where sounds' becoming is underscored through the evolving physical relationship between the body of the performer and the materiality of the score. For those

[7] For a more detailed analysis of the shortcomings of Barthes modes of listening, the reader is referred to [12].
[8] Installation/Performance Notes provided courtesy of the composer.

[9] This, perhaps, as opposed to an idealized Adornian structural listening [7].

scores which do feature the animation of notational descriptors, the animation techniques employed ground the work in a particular temporal modality which fundamentally frame listener engagement.

In her critique of structural listening, Rose Subotnick argues that musical style "…defines the conditions for actual structural possibilities, and that structure is perceived as a function of style more than as its foundation." [8]. This observation is particularly manifest in the temporal modality of projected scores. In Hope's *Longing* or Tinnemans' *Imagiro Landmannalaugar*, for example, the drone-based flow of musical texture is strongly supported and musically reinforced by the scrolling model adopted in the display of musical information as well as the overt use of horizontal, graphic lines in the score. Similarly, in many of Ryan Ross Smith's works for percussion, the gradual acceleration and deceleration of sonic events which results in complex rhythmic textures is strongly supported by the manner in which sonic events are represented in the score through the collision of graphic primitives. It is hard to imagine the processes employed would be as transparent for the listener if sonic events were represented through a scroll-based score. In Marina Rosenfeld's *WHITE LINES*, the becoming of musical processes is strongly reinforced by the concurrent dissolution of the white lines in the score through variations in visual opacity. In all of these works, the temporal modality of the score underpins formal musical structure.

The mnemonic function of notation is extended in the projected score such that it serves as an aide-memoire not only for the performers but for the audience,[10] providing the visual support to relate current events to past but also to better anticipate how future events might unfold. In non-linear forms, open forms, or in visual scripts where denotative relationships cannot be unambiguously determined, this anticipatory function is fundamentally unique.[11] As a live coder edits the parameters of an iterative loop, for example, a listener reasonably cognisant of programming structure can anticipate sonic outcomes. Similarly, as a scrolling score unfolds, transitions from one sonic texture to another can be anticipated even though denotative relationships between graphic typographies and sounded results are not strictly unambiguous. Cat Hope has indicated that the ability of the listener to anticipate outcomes is one of the reasons she would rather an audience not see a score [17].

Through projection of the score, the audience is made aware of a field of structural possibilities that is typically closed with the navigation, decisions, and determinations that the performer/s make embedded as criteria for aesthetic reflection. This is in marked contrast to the experience of a seminal open-form work such as Stockhausen's *Klavierstücke XI* or Haubenstock-Ramati's *Liaisons* in which the virtual pathways through the score remain closed for the listener. The projected score thus concretizes the work's *protentive* possibilities [18].

As non-linear processes become more deeply embedded in a notated script, the ability of the listener to anticipate or protend sonic outcomes becomes more difficult. Nevertheless, the foregrounding of the score presents the audience with all of the work's latent and virtual possibilities [19], not just those that are actualized. In the author's *point studies no. 2*, the listener is presented with the entire field of possibilities that performers can take through the score although only one is sonically actualized. For the listener, the work becomes a field of potentiality ontologically defined as much by its latent possibilities as by those sounded.[12] These potentialities constantly shadow the work's actualization, overtly foregrounding the process of production and entelechy.

5. CONCLUSION

The visual presentation of the real-time score, whether that score be prescriptive or descriptive, invites listeners to engage in a decoding process to develop a deeper understanding of the musical processes underpinning a musical work. While this can rarely be unambiguously undertaken, this ambiguity nevertheless results in perhaps the most ontologically significant outcome in which the latent possibilities visually presented but not necessarily actualized come to establish a world, in a Heideggerian sense, playfully disclosed through sonic realization [21, 22]. The tension between the actualization of a world through sonic becoming, sound's haecceity [23], and the historically mnemonic function of notation forms, perhaps, the locus of aesthetic interest in the practice.

Are the creative possibilities afforded by a reframed listening experience and its subsequent ontological effects, somewhat tempered by a tendency to fetishize notational schemas? Might not the opportunity for an active, structural listening experience be diluted through presentation of notational schemas [24]? On the contrary, I would suggest that a notational schema affords an enriched engagement with a musical performance. Through a rich foregrounding of the score, with its typically inherent non-linearity and protentive suggestion of possibility, the listener is invited alongside the performer/s to playfully engage with a work's structural processes and in turn develop an intimate understanding of the world it explores.

Acknowledgements

I would like to thank Ingibjörg Fríðriksdóttir, Cat Hope, Marina Rosenfeld, Ryan Ross Smith, Jobina Tinnemans, and Lindsay Vickery who were kind enough to discuss work referenced in this paper. Thanks also to Christopher Fox for allowing me to abridge the full version of this paper which originally appeared in TEMPO, Vol. 72, No. 284, 2018.

[10] Adorno suggests that rather than developing as an aide-memoire enabling performances to be recreated, notation in fact served as a means of reifying musical practice most notably through techniques for indicating mensuration [16].

[11] The performance challenges involved in interpreting a generative notation are tangential to the focus of this paper. The reader is referred to [1] for more in-depth discussion.

[12] Agamben touches on this in his discussion of the poetics of the open-work, arguing for a negative presence [20].

6. REFERENCES

[1] J. Freeman, “Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance,” *Computer Music Journal*, vol. 32 no. 3, pp. 25-41, 2008.

[2] J. Tinnemans, <https://jobinatinnemans.com/portfolio/imagiro/>. 2017. Accessed September, 2017.

[3] T. Magnusson, “The Threnoscope – A Musical Work for Live Coding Performance,” in *First International Workshop on Live Programming in Conjunction with ICSE 2013*, San Francisco, 2013.

[4] J. Bezemer and G. Kress, *Multimodality, Learning and Communication: A Social Semiotic Frame*. London: Routledge, 2016.

[5] D. Kim-Boyle, “The Visual Design of Real-Time Scores,” *Organised Sound*, vol. 19, no. 3, pp. 286-294, 2014.

[6] L. Vickery, “Music Screen-Reading: Indicative Results from Two Pilot Studies,” in *Proceedings of the 2015 Conference of the Australasian Computer Music Conference*, Sydney: Australasian Computer Music Association, pp. 119-125, 2015.

[7] T. Adorno, “Types of Musical Conduct,” in T. Adorno, *Introduction to the Sociology of Music*. New York: The Seabury Press, Inc., 1976, pp. 1-20.

[8] R. R. Subotnik, “Towards a Deconstruction of Structural Listening: A Critique of Schoenberg, Adorno, and Stravinsky,” in R. R. Subotnick, *Deconstructive Variations – Music and Reason in Western Society*. Minneapolis: University of Minnesota Press, 1996.

[9] A. Dell’Antonio (Ed.), *Beyond Structural Listening? Postmodern Modes of Hearing*. Berkeley: University of California Press, 2004.

[10] M. Chion, *Audio-Vision: Sound on Screen*. New York: Columbia University Press, 1994.

[11] R. Barthes, “Listening,” in R. Barthes, *The Responsibility of Forms*. Richard Howard (trans.), Oxford: Basil Blackwell, 1986, pp. 245-260.

[12] A. W. Jing, “Affective Listening: China's Experimental Music and Sound Art Practice,” *Journal of Sonic Studies*, vol. 2, no. 1, 2012. [Online]. Available: http://journal.sonicstudies.org/vol02/nr01/a11

[13] R. Jakobson, “Closing Statement: Linguistics and Poetics,” in T. A. Seobok (Ed.), *Style in Language*. Cambridge, MA: MIT Press, 1960, pp. 350-377.

[14] L. R. Waugh, “The Poetic Function in the Theory of Roman Jakobson,” *Poetics Today*, 2:1a, pp. 57-82, 1980.

[15] N. Goodman, *Languages of Art: An Approach to a Theory of Symbols*. Indianapolis: Bobs-Merrill, 1968.

[16] T. Adorno, *Towards a Theory of Musical Reproduction: Notes, a Draft and Two Schemata*. Weiland Hoban (trans.), Cambridge: Polity Press, 2006.

[17] C. Hope, Personal communication, 2016.

[18] E. Husserl, *On the Phenomenology of the Consciousness of Internal Time (1893-1917)*. J. B. Brough (trans.), Boston: Kluwer Academic Publishers, 1991.

[19] G. Deleuze, *Bergsonism*. Hugh Tomlinson & Barbara Habberjam (trans.), New York: Zone Books, 1991.

[20] G. Agamben, *The Man Without Content*. Georgia Albert (trans.), Stanford, CA: Stanford University Press, 1999.

[21] M. Heidegger, “The Origin of the Work of Art,” in *Poetry, Language, Thought*, A. Hofstadter (trans.), New York: Harper & Row, 1971, pp. 15-86.

[22] R. Raj Singh, “Heidegger and the World in an Artwork,” *The Journal of Aesthetics and Art Criticism*, vol. 48, no. 3, pp. 215-222, 1990.

[23] G. Deleuze and F. Guattari, *A Thousand Plateaus*. Brian Massumi (trans.), Minneapolis, MN: University of Minnesota Press, 1987.

[24] T. Adorno, “On the Fetish Character in Music and the Regression of Listening,” in T. Adorno, *Essays on Music*. Richard Leppert (ed.), Berkeley and Los Angeles: University of California Press, pp. 288-317, 2002.

A COMBINATION OF GRAPHIC NOTATION AND MICROTONAL PITCH NOTATION IN THE VIDEO SCORE OF THE OPERA "THE CROSS OF THE ENGAGED"

Christian Klinkenberg
Vrije Universiteit Brussel and
Koninklijk Conservatorium Brussel
christian.klinkenberg@gmx.net

ABSTRACT

This paper describes a procedure for isolating pitch notation in the environment of a flexible real-time graphic video notation. It aims to combine precise microtonal pitch and the flexible interpretation of other parameters such as rhythm, volume, attack and decay. The procedure was developed and tested in the opera *The Cross of the Engaged*. The pitch is notated exclusively by note heads and accidentals on the lower third of the screen, sometimes supplemented by short written explanations or pictograms. These note heads are linked to a corresponding graphic element on the upper part of the screen by using the same color. Each musician, conductor, director, singer and technician was given a custom-made video file, playable on his or her private laptop or tablet, regardless of operating system or video player app used.

1. INTRODUCTION

1.1 The Opera

In January 2015, the preparations for the microtonal opera *The Cross of the Engaged*[1] began. As the financial means were limited, the rehearsal time with the orchestra had to be reduced to the minimum.

After the development phase, the individual scenes were combined through video editing to a single piece with duration of 1h45.

In the process, some scenes' lengths had to be shortened in collaboration with the director. Similarly, silence was added to other scenes in order to create space for musical development. A score for the orchestra, singers and lighting technicians had to be created before the premiere on the 4th of February 2017, based on the transcription of this video.

After some experiments with different notation systems, real-time graphical notation, playable on tablets and notebooks, had been chosen. This made microtonal notation and complex rhythm readable and more intuitive for the musicians. Every musician, conductor, director, singers and even the lighting technician received individual scores.

1.2 The Libretto

Nicole Erbe, writer and theatre director at the *Landestheater Neuss* (Germany), was commissioned to write the libretto and later to direct the opera. The topic was given – the libretto is based on a true story:

Marie Solheid and François Reiff are probably the most famous victims of the High Fens, a huge moor area in the Ardennes. In the summer of 1870, during the fair in the village of Jalhay, Belgium, two young people have met. Soon, they were yearning to be wed. In January 1871, Marie and Francois made their way to Xhoffraix, Marie's birth-place, in order to get her marriage papers, but were surprised by a severe snowstorm. Weeks later, on March 22, 1871, the 24-year-old Maria Solheid's body was found.

2. THE COMPOSER

2.1 Previous compositions

The author's first microtonal compositions[2] included a complex rhythmical structure. When rehearsing, musicians were sometimes overloaded by the sheer amount of information – the double complexity: on one hand the complexity of the rhythm notation, on the other hand the inclusion of the additional accidentals. One possible solution was to search for a possibility to separate rhythm and pitch.

Looking at the compositions of the *"classic"* pioneers [1] in the field of graphic notation, such as Earle Brown, Sylvano Bussotti, John Cage, Morton Feldman and Roman Haubenstock-Ramati, one notices how often the pitch is notated relatively imprecisely. Often the musician is offered an approximate range. Specifying a precise pitch, of course, was usually not a concern of these composers. The origin is likely that these compositions were mostly intended for the 12-equal temperament. The musicians knew all the sorts of intervals and combinations in this temperament due to lifelong experience.

[1] *Das Kreuz der Verlobten – Eine Oper*. http://www.daskreuzderverlobten.com

[2] http://christianklinkenberg.com/

Symbol	Inflection	Cents
	- 1/4 tone	-50
or	- 1/8 tone	-25
	0	0
	+ 1/8 tone	+25
	+ 1/4 tone	+50

Figure 1. Examples for microtonal accidentals.

However, when starting to work with quarter-tones or even eighth-tones, most musicians can no longer rely on this wealth of experience. In this case, the composer must have a clear way of notating pitch while other parameters might remain in the spirit of *"classical"* graphic notation.

The notation for the accidentals (Figure 1) used in the opera is inspired by the suggestions for quarter tones from Kurt Stone's book *"Music Notation in the Twentieth Century"* [2], and was combined with up or down arrows for eighth tone deviations.

2.2 Microtones[3]

Specially designed acoustic instruments based on alternative scales have been constructed in recent years in the recent years. Examples are Stephen Fox's[4] Bohlen-Pierce clarinet or Stephen Altoft's[5] 19edo trumpet. Another tendency is the production of instruments that extend the 12-equal temperament with additional flaps and valves[6] (for wind instruments) or additional frets (for example, for guitars) to quarter, eighth or even twelfth tones. The reason to limit the composition for this project to eighth tones (in consultation with the artistic director of the ensemble) was the collaboration with the *Ensemble 88*[7], which specializes in contemporary music with standard instruments.

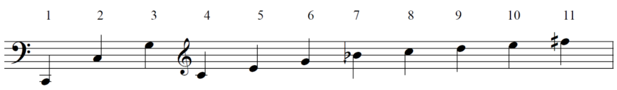



Figure 2. Harmonic series.

The *seventh harmonic* (Figure 3) is lowered by 1 eighth-tone[8] compared to the twelve-equal temperament and the *eleventh harmonic* (Figure 4) by a quarter-tone [3]. The 2, 3, 5, 9 harmonics (Figure 2) and their multiples are not corrected in comparison to the standard notation in twelve equal temperament, because the difference is closer to the traditional notation than the eighth-tone (25 ¢[9]) or quarter tone deviation (50 ¢). The possibility of the notation of quarter-tones and eighth-tones allows a relatively acceptable approach to the 7th and 11th partials.

Eighth-tones can be produced by means of special microtonal techniques on most traditional instruments.

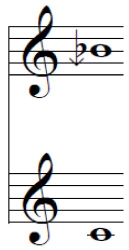


Figure 3. 7th Harmonic ≈ 31 ¢ flatter than a minor 7th in 12edo[10]. Ratio 7/4.

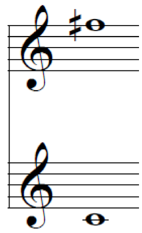



Figure 4. 11th harmonic ≈ 49¢ sharper than a forth in 12edo. Ratio 11/4.

Thus it becomes possible to reproduce approximate intervals from the harmonic series that do not occur in the twelve equal temperament. Some of intervals/ratios (Figure 2) used in the opera are 7/3, 7/4, 7/5, 9/5, 11/7, 11/9.

3. THE PAINTER

3.1 Introduction

Since 2014, the composer and author of this paper Christian Klinkenberg have been working together with the painter Marc Kirschvink[11] on the project called *"Partitur",* intended for a jazz quartet. The musical output was pre-eminently based on intuitive, temporally variable musical interpretations of different elements of a painting. That is, the length of interpretation of individual graphical elements was variable, as was the length of the entire performance. At that time, we were working with individual paintings rather than a series of pictures. Also, the pitch was never specified.

A year later and 4 concerts richer in experience, this collaboration evolved into a part of an opera production. The aspect of improvisation had to be retained, but the temporal aspects and pitches had to be defined.

Real-time video notation with individual versions per instrument was ultimately the result of various experiments with many notation forms. The staves for the pitches ended up being positioned below the common graphic score, which is a series of paintings.

For this opera project the painter Marc Kirschvink was also asked to give final form to the composer's sketches. These sketches were combined in a film score and parts.

[3] A microtone is an interval smaller than a semitone.
[4] *Stephen Fox Clarinets*, http://www.sfoxclarinets.com
[5] An example of the microtonal technique for a standard 3-valve trumpet playing eighth-tones can be found on the blog by Donald Bousted and Stephen Altoft, https://microtonalprojects.com
[6] http://www.21stcenturyoboe.com/
[7] Ensemble 88 is a Limburg (NL)-based musical ensemble for contemporary music, http://www.ensemble88.nl
[8] Rounding up – strictly a 1/6 tone.
[9] ¢ = cent, the standard semitone can be divided in 100 ¢.
[10] edo = equal division of the octave.
[11] *Marc Kirschvink's Website*, http://www.mkirschvink.com

3.2 Graphical elements

Under the painting, the lyrics and microtonal pitches in eighth-tone notation were notated in 4 different, easily distinguishable colours: red, green, blue and black.

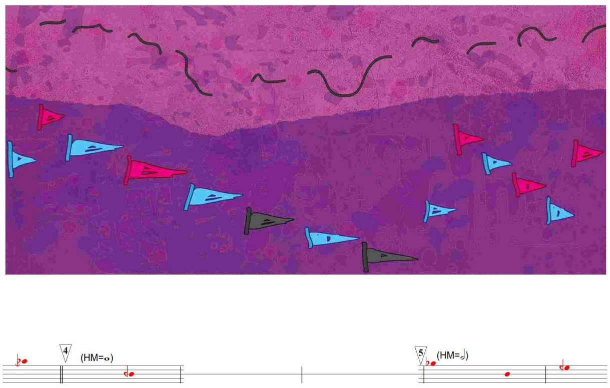


Figure 5. Flute, page 99.

First example: The flute-player (Figure 5) has to play the red graphical symbol. A possible interpretation would be a strong attack followed by a long decrescendo. Other musicians play the blue and black shapes.

Second example: Every musician, in this case the viola (Figure 6) plays his/her individual film score on a tablet or a notebook. Underneath the painting that is common to all, one can see the separate pitch notation that has to be combined with the graphical symbols in the upper painting with the corresponding colour. The viola player has to play the green clouds. As the violin is using this colour as well, both pitches are included for an easier intonation.

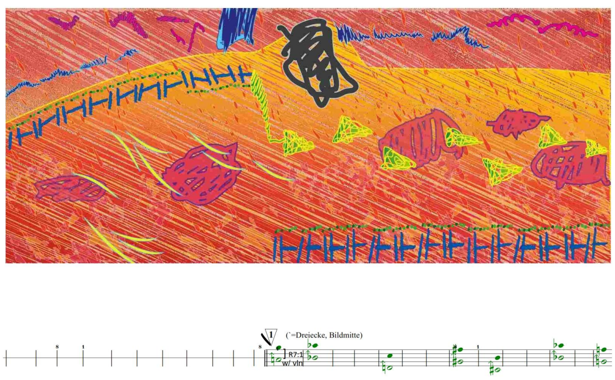


Figure 6. Viola, page 28.

3.3 Tempo and grids

The scrolling in this graphical score runs from right to left. The score length on the sketch paper was 40 cm per page. It takes 40 seconds to perform one page. This is not apparent to the performer, since the individual sheets are connected to each other.

The rhythm can be precisely transferred to a transparency (Figure 7) with the different underlying grids.

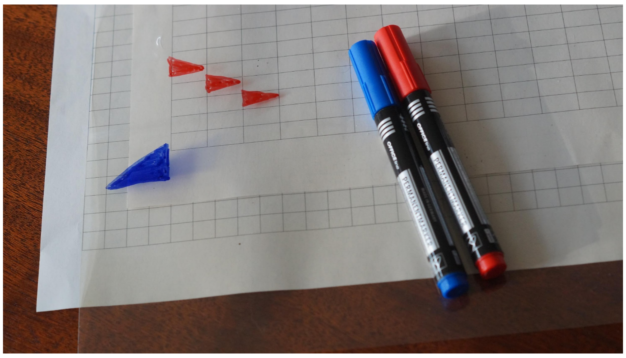


Figure 7. Transparent foil, permanent markers in different colors, grids for 54 and 32 BPM.

3.4 Duties and liberties of the painter

The painter receives the sketches of the composer. The sketch is scanned into the PDF[12] format and the painter can save this sketch as a layer[13] in his drawing software (in this case a combination of Adobe Acrobat [4], InDesign [5] and Illustrator [6]). After this, he can paint over this sketch in a new layer. The size, the basic colour, the vertical and the horizontal position, the background colour, the basic form and its outline are specified by the composer.

3.4.1 Size

The size must be kept, as it affects the volume parameter.

3.4.2 Color

The basic foreground color (black, red, blue, green) must be adhered to, as the form must be linked with the note on the bottom part of the screen. The basic emotions (anger, disgust, fear, happiness, sadness and surprise) [7], researched by the American psychologist Paul Ekman [8], are used as a basis for the background. These have then been combined with the psychology of colors. In Figure 5 the background is colored in violet. *"Violet"* stands for a depressed situation. In Figure 6 the background is colored in red. *"Red"* stands for anger and passion.

However, the painter has the opportunity to adapt the color tone and the graphic design to his pictorial composition and to the background.

3.4.3 Positioning

The vertical and horizontal position must be adhered to, since the temporal aspect of the note is important in terms of composition.

[12] Portable Document Format (PDF) is a file format used to present and exchange documents reliably, independent of software, hardware, or operating system.

[13] https://helpx.adobe.com/illustrator/using/layers.html

3.4.4 Form

The basic form (triangle, square, circle, cloud, funnel ...) must be respected by the painter, but the design is left to him. For example, within the size, the painter can creatively form the shape of the cloud.

3.4.5 Outline

The composer gives the outlines. For example, wavy, jagged, round, straight. However, the painter also has creative freedom to a certain extent: Shape of the waves, regularity, irregularity or thickness of the outline.

3.4.6 Filling

The painter can choose the filling of the form. For example, the default may be *"restless"* or *"fluent"* (added to the sketch in writing). The painter can freely choose the design of these fillings.

3.4.7 Background

The composer gives the background colour, as the background can change the character of the music. This way, through the background colour selection, the musical scene becomes *"agitated"* or *"graceful"* – without the composer's notions such as *"agitato"* or *"grazioso"*. In addition to the colour selection, the background design chosen by the painter (large dots, lines, blurred shapes...) can influence the musical character as well.

3.5 Mutual enrichment

The influence of the painter on the musical output is signicficant. The mood is completely changed with only small graphical variations. There are possibilities of influencing the tone colour, shape variation, outlines and filling. An example: the strings change the style of their playing according to the intensity of the form in the graphic. So the painter can to a certain degree influence whether the strings ultimately play *"arco", "pizzicato", "tremolo", "sul tasto"* or *"sul ponticello"*. It would be an interesting experiment to explore the musical output from the work of another painter using the same composer's sketches. Apart from that, the overall visual impression is much more appealing, compared to composer's rough sketches.

4. SIMILAR APPROACHES

Only a few months after the premiere of the opera, the composer learned about a software called *Decibel ScorePlayer*[14]. It is interesting to see how many similar solutions have been found: screen scrolling from left to right, vertical line indicating position, possibility of relative pitch by vertical arrangement, etc. [9]

In the opera, the vertical line was intended only for the conductor, not for the musicians. In this way, the conductor was able to detach himself from this line as needed. This was necessary, especially because the singers/actors had to play freely and therefore did not have the screen with the video score while singing.

A big advantage of the *Decibel ScorePlayer* is the synchronization of the scores over a network connection. This means that all musicians always see exactly the same time window. In the opera, the conductor counted at the beginning and all musicians had to press *"play"* simultaneously. An exact synchronization can of course not be guaranteed in this way. However, it did not cause problems in this case, as the conductor conducted with numbered hand signals.

Knowing about *Decibel ScorePlayer* would probably have significantly influenced the development of this project. Individual solutions, such as the isolation of the pitch might not have been found.

A disadvantage of *Decibel ScorePlayer* is the binding to Apple. This would have already been a deal-breaker for this project because of budgetary restrictions. The musicians are forced to use a single vendor's hardware exclusively. Also, the painter could not create the entire composition on canvas as was the case with the software of his choice.

5. THE PERSPECTIVE OF THE MUSICIANS

After initial difficulties, the musicians (Figure 8) quickly got used to this alternative notation system. Of course, some musicians are alienated by the world of computers. Another negative point was the unreliability of computers as opposed to notes on paper. At the concerts there were several crashes of individual computers. For this reason, a replacement computer was provided (Figure 9). It included all scores. It was observed that the sections in the composition, which had a stronger rhythmic impetus, were difficult to coordinate. Intricate rhythmic writing is probably not best suited for this style of notation.

Figure 8. The premiere, beginning of the second part.

However, there are a lot of advantages: Different tempos can be performed concurrently and the music is understandable even to non-musicians: the lighting technician got his own score, which did not require any knowledge of music notation, but it still allowed a comprehension of the scenic and musical events.

[14]http://www.decibelnewmusic.com/decibel-scoreplayer.html

Figure 9. One rescue-notebook is placed on the scene for safety. With so much electronics involved there is a higher probability of failure.

A very positive aspect was the possibility of improvisation. The musicians interpreted some graphic symbols in very different ways during the 5 performances. On the other side, the lack of possibility to make notes (e.g. conductor's remarks) on the score was a common complaint.

6. THE CONDUCTOR

Apart from the conductor, none of the musicians and singers received a fixed positional reference on the screen. The conductor received a vertical black line (Figure 10) on his score, which would not change its position during the play. Thus, the conductor could act freely within the score in a range of 40 seconds. There is always a possibility that the singers in certain places *"wander"* too far in advance, or they may lag behind in other places. Similar to Lutosławski in his *3rd Symphony*, 5 numbers were used in connection with the finger marks. These are hand signals from numbers 1-5, in which the respective number of fingers is shown. The numbers are notated in the middle of a downward pointing triangle. Kurt Stone also proposes this solution, which is quite similar to the subject of video notation, in the context of *Proportional Notation* in his book. This way the director is able to tell or show the musicians where on the screen he wants them to be at this moment. This is especially important in free passages.

7. THE SOFTWARE

Finale[15] was used as a notation software for the lower part of the score. The individual line for every musician involved was placed under the painting.

The paintings were transformed into high resolution JPG[16] files, so they could later be incorporated in a slideshow (Figure 11) with *Adobe Premiere Pro*[17].

The painting was in 16:7.5 format and the score in 16:2.5 format. Afterwards the movies could then be rendered in 720p resolution.

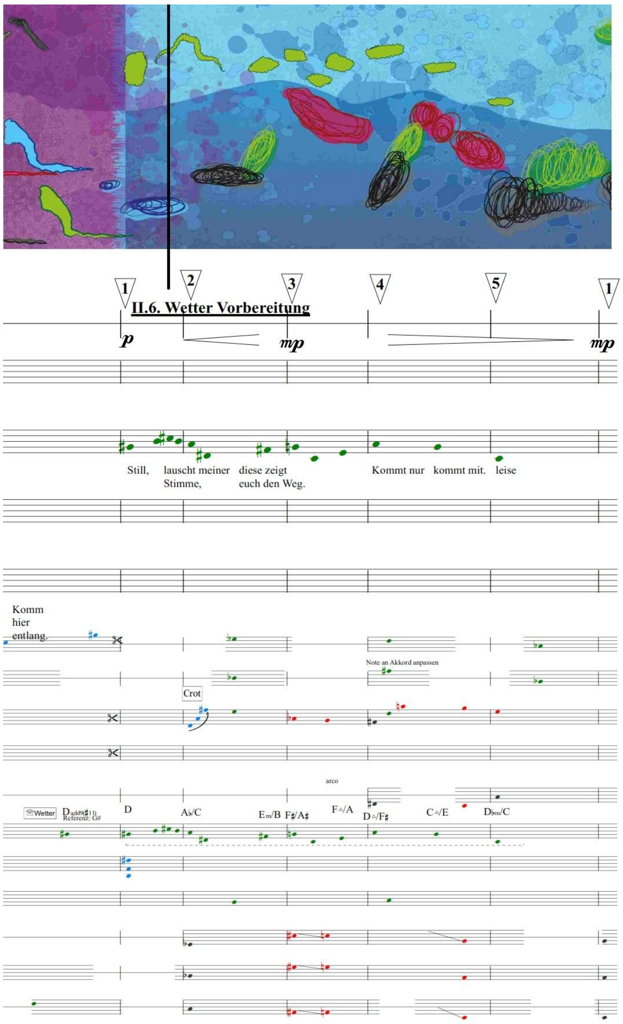


Figure 10. Page 52 of the score: The conductor has a vertical black line on his score and finger-numbering marks are indicated.

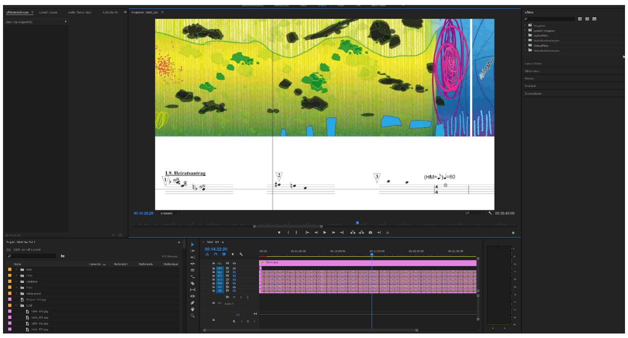


Figure 11. In *Adobe Premiere Pro* all JPG files had been imported: on top is the painting that is common to all the parts, on the bottom are the individual pages from *Finale* (exported to PDF and converted to JPG).

[15] https://www.finalemusic.com/.
[16] JPEG is a common format for lossy compression of digital images.
[17] Video editing software [10].

8. CONCLUSION

8.1 The score

The main reason for the graphic notation was the isolation of pitch from the other parameters. This has succeeded in any case. The pitch is easily readable despite many additional accidentals. Unfortunately, structured fast rhythms are easier to read using traditional notation. This implies that the composer must adapt his music to the notation method.

8.2 Outlook

A composition in the alternative tuning system Bohlen-Pierce[18] is planned for the 27th of April 2018. Unlike the 12edo, the repeating frame is not formed by an octave, but by a tritave[19]. This tritave is divided into 13 steps. The result is an alternative scale that opens new possibilities to music. For this reason, the musicians have specially constructed instruments (clarinet, keyboard, chimes). The string instruments (e-guitar, bass, and violin) also need to retune their strings and have to increase the spacing of the fingerings or frets by about half.

The use of a characteristic alternative temperament such as Bohlen-Pierce opens the doors to a interpretation of the intervals that has more freedom compared to *"classical"* graphic compositions, as mentioned at the beginning of this paper. It is quite similar to a guided improvisation. The lower note head would not always be necessary. The use of eighth-tones, as in the opera is an extension of 12edo. Compared to improvising in systems like Bohlen-Pierce, the combinatory possibilities of the intervals are almost too many.

Since this graphic notation requires no prior musical knowledge, it is possible to integrate the audience into the performance via a projection. This is made possible by a web app for smartphones or tablets (Figure 13). The spectators will interact proactively with the orchestra by reading the projected symbols.

A second opera is planned for November 2019, combining various systems like 19edo, 48edo and Bohlen-Pierce [10] (Figure 12).

8.3 Considerations for improvement

8.3.1 Synchronization

In the open-source software *syncplay*[20] there is the possibility of cross-system synchronization of video players such as VLC[21] or MPV[22] over network.

8.3.2 Alternative to the connection through colors

Unfortunately, colorblind people are disempowered by the combination of the music heads and graphic elements. One approach would be to highlight the linked graphical elements and to blur the remaining elements.

Figure 12. For the next opera project (2019), three microtonal keyboards have been built. From top to bottom: A Bohlen-Pierce keyboard in Lambda mode; a 19edo-keyboard (yellow: C, black C#, silver Db, white D, black D#, silver Eb, white E, ...); and The Bohlen-Pierce Keyboard in Dur II-Mode.

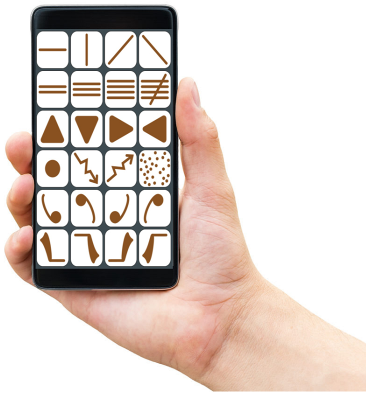


Figure 13. Web app for smartphones. Spectators can press on different symbols to send commands to live electronics during performances.

8.3.3 Clearer indications of rests

As a pictogram to indicate that the colour connection (e.g. black) is over, the scissor-symbol was chosen. Sometimes, however, the musician overlooked this symbol. This means that the musicians combined the previous sound with the coming symbols of the same colour that were not intended for them. An alternative would be to hide the staves in the parts where the musicians should not play.

Acknowledgments

I would like to thank Peter Swinnen and Bart Bouckaert for fruitful discussions. The painter Marc Kirschvink for making valuable suggestions for improvement. I would also like to thank all those idealistic people involved in the opera. Be it the actors/singers, the director/librettist Nicole Erbe, set designer, the musicians, the technicians and the co-producers. The German-speaking Community of Belgium for the financial support. Last but not least, the lectors Vedran Mutić and Donald Bousted for their great suggestions for improvement.

[18] *The Bohlen-Pierce Site*. http://www.huygens-fokker.org/bpsite.
[19] perfect twelfth.
[20] https://syncplay.pl
[21] www.videolan.org
[22] www.mpv.io

9. REFERENCES

[1] E. Karkoschka, *Das Schriftbild der Neuen Musik*. Moeck, 1991.

[2] K. Stone, *Music Notation in the Twentieth Century: A Practical Guidebook*. WW Norton & Co, 1980.

[3] S. Safari and M. Stahnke, *1001 Mikrotöne.* von Bockel Verlag, 2015.

[4] L. Fridsma and B. Gyncild, *Adobe Acrobat Classroom in a Book.* Adobe Press, 2014.

[5] K. Kordes Anton and T. DeJarld, *Adobe InDesign CC Classroom in a Book.* Adobe Press, 2017.

[6] B. Wood, *Adobe Illustrator Classroom in a Book.* Adobe Press, 2017.

[7] R. J. Davidson, *Nature of emotions*. Series in Affective Science, 1995.

[8] P. Ekman, *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life*. Times Books, New York, 2003.

[9] C. Hope and L. Vickery, "Visualising the Score: Screening Scores in Real-time Performance." in *Interactive Media E-Journal,* 7, 2011.

[10] M. Jago, *Adobe Premiere Pro CC Classroom in a Book*. Adobe Press, 2017.

SCORING AN ANIMATED NOTATION OPERA – THE DECIBEL SCORE PLAYER AND THE ROLE OF THE DIGITAL COPYIST IN 'SPEECHLESS'

Cat Hope
Sir Zelman Cowen School of Music
Monash University,
Melbourne, Australia.
cat.hope@monash.edu

Aaron Wyatt
Sir Zelman Cowen School of Music
Monash University,
Melbourne, Australia.
aaron.wyatt@monash.edu

Dan Thorpe
Electronic Music Unit
University of Adelaide,
Adelaide, Australia.
thorpe.daniel@gmail.com

ABSTRACT

This paper outlines the developments made to the Decibel ScorePlayer, the software that enables the delivery of the digital, graphic animated score of an hour long new opera by composer Cat Hope, entitled Speechless. The Decibel ScorePlayer is an iPad application that delivers the coordinated reading of graphic notation that was first created in 2012 and has been updated regularly ever since it was made available on the iTunes store the following year. Engaging the software to deliver the score for an hour long opera featuring a thirty piece orchestra, a thirty voice choir and four soloists saw considerable improvements made, contributing to a smooth running workshop period for the work. The workshop saw the score for the opera being updated, revised, added to and shared with different sections of musicians, vocalists, technicians and stage managers daily. This paper summarises the major additions to the score player software that came about as a result of this workshopping period and discusses procedures, developments and contributions engaged to facilitate the score updating process in a digital score for a large-scale work.

1. THE SCORE FOR SPEECHLESS

Composer Cat Hope has been using digital graphic scores to notate her music since 2008. While drafted with coloured pens and paper, the final graphic scores are created in the design program Illustrator, and exported as PDF for hardcopy or PNG files for the softcopy 'screen score' [1]. The development of Hope's composition practice has occurred alongside the development of the Decibel ScorePlayer application[1], with creative ideas for new works feeding the development of the player, and vice versa.

The Score Player application connects multiple devices playing the same score and keeps them accurately coordinated. It enables the viewing of independent parts and has functions such as pause, change of speed and the ability to scroll through the score to find particular points in the score.

The majority of Hope's scores use a 'scrolling screen score' format [1] where the graphic passes by a playhead, a vertical line that signals the point at which performers perform the score. While there are other models for the score player software, this is the one used for Speechless. To facilitate the workshopping period of the work, the programmer who wrote and continues to update the Decibel ScorePlayer (Wyatt) was engaged as the music director for the project, and a 'digital copyist' (Thorpe) worked with the director and composer on revisions and updates. The first of the two weeks saw a series of smaller workshops with soloists, the choir, and section leaders of the orchestra. The second week saw larger scale rehearsal with full orchestra and all performers.

The opera involves two community choirs (around 30 performers) from the local area, the Australian Bass Orchestra (a scratch orchestra of around 30 performers, all playing instruments that can perform notes below middle C [1]) led by four 'section leaders', and four vocal soloists. The work is made of three acts, an overture, and a short interlude section. Each instrumental group has its own colour in the part, as do the vocal soloists. The choir parts have similar colours to the vocal soloists, but in different opacities. The instruments sometimes have divisi parts, and some instructions presented on the score as text in english. The notation is proportional - pitch denoted by placement on the vertical axis, dynamics, and density by the thickness of the line.

The workshop period concluded with two performance showings of the work at the conclusion of around ten working days. The work was conducted by Wyatt, and featured some minimal staging, lighting, and sound reinforcement.

[1] https://itunes.apple.com/au/app/decibel-scoreplayer/id622591851

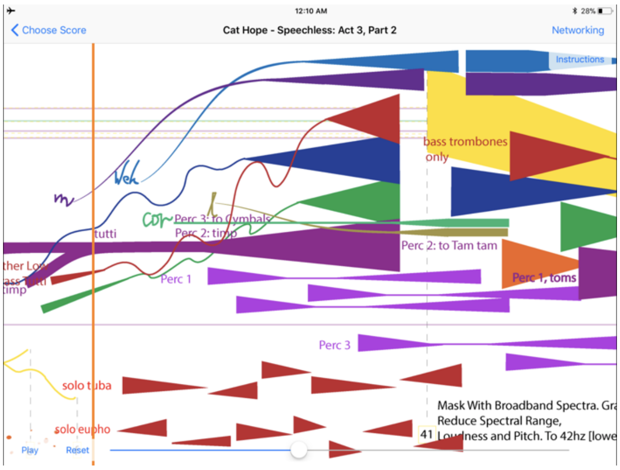


Figure 1. An excerpt from a section of the Speechless master score showing the red (brass), purple (percussion – in different shades of purple according to instrument), yellow (electronics), Orange (harp, piano and bass guitars), green (wind), blue (cello/double bass) and four vocal parts (violet, navy, brown and lime) and text instructions.

2. FACILITATING MULTIPLE UPDATES

To date, the largest number of iPads engaged simultaneously while using the Decibel ScorePlayer app had been 18, serving around 30 performers, in a performance of Hope's The Moment of Disappearance, at the performance venue Carriageworks in Sydney in 2014. The software worked smoothly in performance, although there were networking issues that arose in rehearsal that have since been resolved, mostly due to the heavily automated manner in which the ScorePlayer was making use of the Bonjour protocol at the time [2]. Loading the score onto that many iPads was time consuming and cumbersome, largely due to the proprietary, walled off nature of the iPad software. Each individual device had to be connected in turn via USB to a computer, with the score file transferred using the file sharing feature of iTunes. This was very time consuming for a one-off performance, but for a workshop environment where new versions of the score need to be uploaded to around 30 devices on at least a daily basis, this has the potential to become a major obstacle.

To help mitigate this problem in preparation for the Speechless workshops, a new option to check for score updates from a web server was added to the ScorePlayer software, with further refinements made over the course of rehearsals. It was also decided to bundle together the individual sections (acts, interlude, and overture) that made up the entirety of the opera, each appearing as a separate score in the player, within the one overarching score file to limit the number of downloads required, and to ensure that there was no chance that version discrepancies could occur between the different acts. While the first version of this file had to be transferred via iTunes as usual, subsequent versions of the score could be downloaded over the network from within the ScorePlayer itself. To achieve this capability, the opus.xml [3] file within the score file first needed two additional elements: one that defines the current version and one that points to a header containing update information.

```
<version>0.54</version>
<updateurl>
http://finn.psiborg.org/scores/Speechless
</updateurl>
```

The header itself, hosted on the update server, is a simple text file containing two comma separated values. The web address of the most recent revision of the score file and its version number:

```
http://finn.psiborg.org/scores/Speechless.dsz,0.55
```

When checking for updates, the ScorePlayer compares the current version of the score with the version given in the update header and if the remote score has a higher version number it gives the user the option to download the new score file. The code that it uses to extract and install the zipped score file from the server is the same code that is used to install a file installed locally via iTunes, so no additional checks are performed to verify that the server is supplying what it claims to be. While this level of trust in the server is acceptable when using a tightly controlled local network, as was the case in these workshops, some level of security checks will need to be added in the future if the feature becomes more widely adopted, especially if score updates are hosted online.

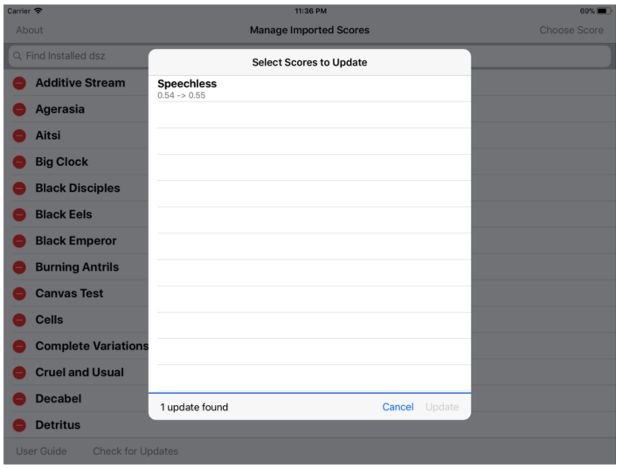


Figure 2. The update window in the ScorePlayer.

Further refinements to the update code included fixing some minor cosmetic issues (cleaning up the way some messages were displayed in the status bar) as well as fixing some more serious bugs. The code uses Apple's NSURLSession API [4] to download both the header files and the updated score files, and by default any HTTP requests are cached. This resulted in some of the iPads failing to see that updates were available if they had a recently cached version of an outdated header file. While this could be worked around by using Cache-Control directives [5] on the web server end, the more permanent fix was to change the caching policy on the iPad for our header download NSURLSession, as demonstrated in the code below:

```
//Create our initial URL session.
NSURLSessionConfiguration *headerSessionConfig =
    [NSURLSessionConfiguration defaultSessionConfiguration];
headerSessionConfig.timeoutIntervalForRequest = 2;
//Make sure we always download our headers and don't use any
    cached responses.
headerSessionConfig.requestCachePolicy =
    NSURLRequestReloadIgnoringCacheData;
headerSession = [NSURLSession
    sessionWithConfiguration:headerSessionConfig];
```

We also had issues at times with the response received by the iPad containing unexpected trailing characters which affected the version number. While it was unclear whether the issue was within the Apple API itself, our adoption of the API, or our web server, we were able to correct the problem by coding the ScorePlayer to ignore any header data beyond the first newline character, and to trim the received data to the length expected from the HTTP headers.

3. FURTHER SCOREPLAYER IMPROVEMENTS

As well as the code additions that allow for the updating of scores within the ScorePlayer, a number of other changes were put into place. These were principally to ensure the performance ran as smoothly as possible with our increased number of iPads on the network. As such, most of these changes were to the networking code, with the aim of increasing both stability and ease of use. These included simple changes such as a new alphabetical sorting of the list of connected iPads in the network connection screen, that made it much easier to see whether all of the devices were connected as expected and to quickly identify any that weren't. Naming and labelling the iPads by instrument also greatly helped with this process, as users could call out the name of the iPad, or the conductor could identify the iPad by instrument section, rather than having to go into the settings to see the assigned name of their own iPad.

Before the Speechless project, changing between scores required having to disconnect from and reconnect to the network. It was soon apparent that this would be a problem in a single long form work made up of different score sections that are sometimes required to flow easily in to one another. The change meant that the conductor was able to move the ensemble seamlessly from one section of the opera to the next without the need for any of the performers to interact with the iPads. The only way to achieve anything like this previously would have been to create one massive, single scrolling score that contained the entire opera. While doable, it would have made it difficult to quickly and easily change the duration and tempo of individual sections – an important facility in a workshop session. While files could be stretched or compressed in Illustrator (where the scores were created), this would have made the source image more difficult and unwieldy to work with, and consume considerable time.

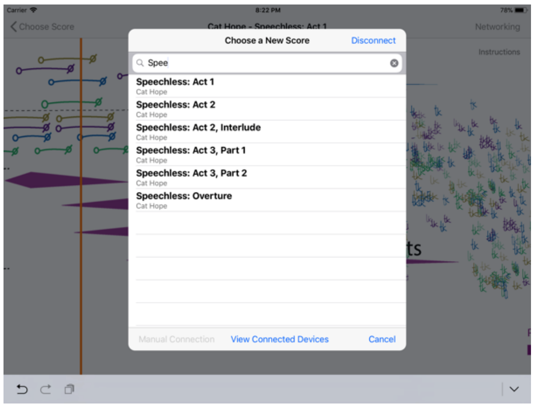


Figure 3. The score change window in the player.

To achieve this capability, a few new network commands needed to be added to the ScorePlayer, breaking compatibility with the previous version of protocol. The new network protocol (Decibel Networking Protocol v14) has been used in all versions of the ScorePlayer now since version 1.8.0, first released to the Apple App store on the 28th of June, 2017. Any previous versions of the ScorePlayer will no longer connect to these newer versions and must be updated. As with many of the existing ScorePlayer commands [6], these new commands can also be sent via Open Sound Control (OSC) from external devices, such as a laptop running MaxMSP, allowing further control and automation options if desired. These commands are outlined in Appendix A.

4. THE SCORE SERVER

The device used to host the update server was a first generation Raspberry Pi Model B [7], named 'finn', running the Raspbian Jessie distribution. As well as running Apache2 to provide our HTTP server, finn ran DHCP and DNS servers to provide network configuration and name resolution services for the iPads. The advantage of this approach is that it provided a single, cheap, and highly portable device that, once set up, can be easily deployed in conjunction with a wireless access point in any venue. While we used the ISC DHCP server and Bind for DNS during the workshop, subsequent smaller projects and performances have used a more lightweight Dnsmasq utility[2].

As well as providing a staging area for updated scores, finn provided us with the opportunity to automate aspects of the ScorePlayer score creation process. Each score has a master and several parts that are extracted from it (in this case strings, percussion, wind, brass, electronics, choir, soloists, bass guitars/piano/harp). In the design of the original Illustrator masters, the parts are ascribed to separate layers, that are exported separately to create parts in the ScorePlayer 'score file'. To make a score file for the Decibel ScorePlayer, the score png image/s must be combined with other data and turned into a specific file type with the filename extension 'dsz'. By connecting to the network and navigating to http://finn.psi-borg.org/scores/, the composer or copyist was able to

[2] http://www.thekelleys.org.uk/dnsmasq/doc.html

upload new versions of the png score images to the server. By setting large enough values in the php.ini [8] configuration file for **upload_max_filesize**, **max_file_uploads**, and **post_max_size**, the images for every part and every section of the score could be uploaded in a single, convenient transfer.

The PHP script that handled the upload only accepted files that were named as expected so that we didn't end up with a lot of unrelated materials in the upload directory, and so that the uploaded files worked properly with the score creation script. Since the ScorePlayer checked this by comparing the names of the incoming files to the names of the files already in the server's images directory, empty files initially needed to be created with the Linux *touch* command[3] before the first set of images could be uploaded. This check also meant that the person uploading the images could be quickly alerted to any error that might have occurred in the image naming process.

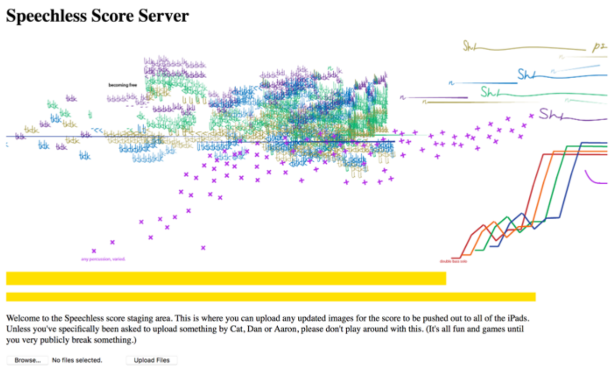


Figure 4. The score update web page hosted on finn.

Once the updated score image files had been uploaded, the score creation Python script, MakeDsz.py, could be run manually to create the new score file and header file. The only user interaction required with the script was the entry of the new version number, although in future even this could be automated if desired using a version number based on a timestamp. As well as editing the version number in the opus.xml file and writing the new header file, the script also split images that were too big for display in the ScorePlayer into equally sized tiles using the Wand[4] ImageMagick binding for Python. It then zipped the opus.xml file and all of the image resources together to create our finished score file.

Earlier versions of the script attempted to call the ImageMagick convert[5] command rather than using the Wand library. While this worked in tests on an Apple laptop, performance on the Raspberry Pi was very poor, with memory shortages proving a major issue. Using Wand fixed these issues, and allowed the script to run faster and more reliably. Manipulation of such large images is still relatively slow on the original generation Pi, however recent testing on a Raspberry Pi 3 Model B shows that the newer devices fare much better. Measured using the Linux time command, the score creation script took 2'35" to run on the newer device compared to 30'14" on the original – a marked improvement.

5. SCRIPTING ILLUSTRATOR

When it was discovered that Illustrator could be scripted using Javascript [9], we saw an opportunity to automate the exporting of png files from the original score image Illustrator files. As well as eliminating the potential for human error in the process (that occasionally resulted in files of the wrong dimensions), it also assisted in the creation of score images for individual parts. Performers found it useful to see the whole score, but often had issues seeing their own part within the gamut of instruments. In addition, their part was sometimes obscured by other parts. To assist the performers, parts were made for each section, where their part was shown at full opacity, while the rest of the orchestra was at 30% opacity. While this would have been possible to do manually, it would have been a time consuming and tedious process.

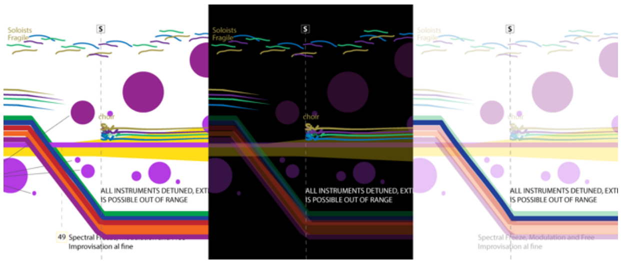


Figure 5. A side by side comparison of an excerpt from Speechless showing the full score to the left, the projected vocal part in the middle, and the blue cello/double bass part highlighted, with the rest of the parts at 30% opacity on the right.

It also provided the opportunity to fix another issue that arose from the performer workshops. The singers (both choir and soloists) were reading their parts from a projection behind the audience, and the score's white background was producing a lot of undesirable light bleed from the back of the hall into the main performance space. As well as lighting the space undesirably, it drew the audience attention to the score, rather than obscuring it, which was the original intention of projecting it there. The team also wanted to experiment with the idea that it may be easier to distinguish the singers individual parts from a distance when placed against a darker background, as the choir is broken into four parts at times, and there are four soloist parts. To add the black background manually in Illustrator would have required the selective colour inversion of all of the common elements of the score, such as the text used to signal rehearsal marks and general performance instructions, which were all in black, one element at a time. Automating the task was definitely the preferred option for accomplishing this.

The main body of the script loops through and sets the opacity for each of the layers as required, before exporting a png at the correct resolution for each part. In addition to the layers for each part, there is a common guide laye showing rehearsal marks to all players. These and

[3] http://man7.org/linux/man-pages/man1/touch.1.html
[4] http://docs.wand-py.org/en/0.4.4/
[5] https://www.imagemagick.org/script/convert.php

the layer for parts are set to 100% opacity, while the other layers are set to 30%. Conditional branches within the loop can be used to accommodate special cases. For example, the percussion and electronics parts contain three different colours in a single layer that need to be simultaneously set to 100% opacity. A subroutine is run for the vocal parts, that creates temporary duplicates of the guide and vocal layers with any colour that is almost black inverted, along with a black background layer fitted to the size of the Illustrator artboard. These are created and destroyed every time the script is run so that they precisely duplicate any changes that have occurred in the original working copies of the layers. For the purposes of the script, almost black was any object with an RGB value of less than 15,15,15 or a grayscale value of less than 30%.

Because of the nature of the object model employed by Illustrator's scripting engine, where layers and groups of objects can contain sublayers and subgroups, the inversion function had to act recursively. Each time it is called it iterates through the collections of path items, text frames, and compound path items that belong to the supplied group or layer, before being called again on any subgroups. While other types of objects can exist in a layer or group, the score for Speechless did not make use of them and so we did not include them in our script. The script should suffice, however, for any score that only makes use of vector graphics and text. And if colour inversion is not required, and only the layer opacity changes are needed, then it would work with little modification for a score that included these additional types of objects.

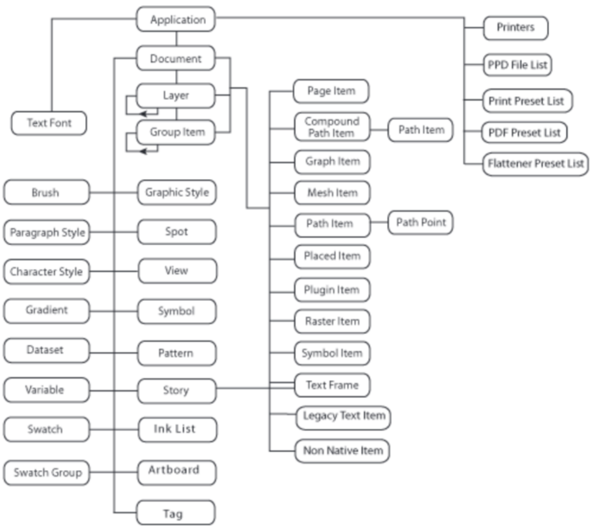


Figure 6. The object model used for scripting Illustrator[6] to generate the parts in the score. Layers and groups can contain sublayers and subgroups, as well as any number of objects in the adjacent column.

Since the script finds each layer based on the name assigned to it in Illustrator, alternative names can easily be substituted into the script. Additionally, a composer could simply choose to stick to a common naming convention for the Illustrator layers of their scores to make the script reusable. An array holds the list of layers that are iterated through in the main loop, while another variable holds an index to a special "GUIDES" layer that appears in every part. Since the script checks whether layers, including the guide layer, exist before performing any operations on them, it would be possible to fill the array with any possible layer name that you might want to use, and the script will export only parts corresponding to those layers that actually exist. It would also be possible to create special branches within the loop for specific layer names. For example, any layer whose names starts with "INVERT" could have its colours inverted in the way previously described.

The code excerpt below shows the start of the loop and the snippet of code that precedes it, where the guideIndex variable is set and where our array of layer names is set up. By checking if the result of getLayerIndexByName is equal to -1 we can find out whether the layer exists or not and can then either ignore it or operate on it as appropriate.

```
var guideIndex = this.getLayerIndexByName("GUIDES");

//Reshow our layers one at a time and export a png.
var parts = ["BLUE", "GREEN", "OLIVE", "ORANGE", "RED",
"YELLOW", "STAGING"];
for (var i = 0; i < parts.length; i++) {
   var layerIndex =
      this.getLayerIndexByName(parts[i]);
   if (layerIndex != -1) {

      ...
```

6. THE ROLE OF DIGITAL COPYIST

Outside of new software developments facilitating the ScorePlayer, a work of Speechless' scale and type also required a fresh approach to the scoring and rehearsal workflow. This was expedited by a unique relationship between the composer and copyist in the rehearsal cycle. Prima facie, the role of copyist in the digital, graphic, and animated score context [10] is not dissimilar to that in traditional notation: it requires an informed approach to refining and organising the composer's ideas, a knowledge of orchestration, and the ability to identify the needs of sections/performers to create adequate parts. In the case of Speechless, however, the copyist also needed to understand the composers' unique notational techniques, their approach to using the software (in this case, Illustrator) and the conventions of the ScorePlayer software. These aspects when combined with traditional approaches were informed and ultimately reshaped by the decidedly non-traditional delivery medium of the ScorePlayer itself.

Hope's notational practice and the ScorePlayer application allowed for several practical innovations that would otherwise be arduous in a more traditional context. In the context of Hope's notational practice, the clear advantage is the ability for players to read from a score while still having a clear and defined part through the opacity and colour coding discussed above. The ScorePlayer's ability to deliver essentially unlimited variations for groups of performers and was incredibly useful, and the copyist worked very closely with the programmer to facilitate the different requirements of individual sections of instruments. The use of vertical dashed lines to coordinate

[6] https://www.adobe.com/devnet/illustrator/scripting.html

player entries, and rehearsal marks as points of synchronisation were very useful in this automated environment, with the assistance of the conductor.

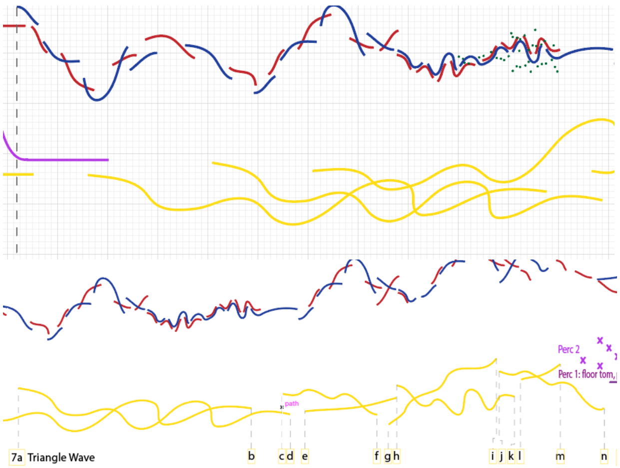


Figure 7. A screenshot of an early-stage, un-annotated section of Speechless, followed by a later draft of a similar gesture with additional cues and instrumentation marked in. Some cues – such as those on the yellow (electronics) part – came from the performers needs. Others, as deemed necessary by the copyist and composer.

The key challenges of the copyist role were the organisation required for layer-by-layer export, and the establishment of orchestration conventions in a decidedly unconventional context. A clear example of this was present in the foreground vs. background placement of musical materials. Hope's work uses proportional notation on both a horizontal and vertical axis, but also has a depth through the use of layers. In the context of a colour coded score, for example, this could mean an orange gesture appearing over a red gesture in one section, and then appearing below in another section. There needed to be a way of notating the depth, or complexity of multiple parts in the master, while enabling clear parts for the performers. Another example could be where there are five parts all playing at the same place, same dynamic on the vertical and horizontal axes - how can part be created that describes all the parts, but is accurate for each performer? Because instruments were sorted into layers on the master score, all of that instruments' gestures for that movement had a fixed, global depth in the part relative to other instruments. The copyist spent three days at the beginning of the workshop checking, re-organising and imposing a somewhat artificial hierarchy on months of work by the composer to simplify and ensure the parts were consistent and some level of depth was enabled on the master score.

Orchestration conventions were similarly challenging. A convention of indicating text instructions on a per-part and score-wide basis was required in a way that would not be interpreted as further graphic notation, and within the layer export hierarchy of the Illustrator script. A convention of using black, dashed lines and clear, bold typesetting within white boxes with black borders was established for score-wide text instructions. On a per-part basis (for example: divisi and tutti), colour coded text was used and black lines continued to indicate start times to contrast with the colour coded graphic notation. Score-wide instructions were the top of the layer hierarchy, and part instructions were in the same order as their respective part but always above the part to which they applied. In the case of the electronics parts, cue numbers were indicated in yellow-bordered boxes (similar to rehearsal marks), with dashed lines indicating start times, as seen in Figure 5.

An issue that was discovered in the performance workshops was the description of parts in the player. When the performer opens the score in the ScorePlayer on the iPad, they see the full score. Using an upward swipe motion, they move between parts. It didn't take long to realise that if the instrument didn't have any music in the first few minutes of the piece, they would be unable to know which was their part, unless they remembered how many swipes it took, which is a rather unreliable method. The copyist devised a method where a coloured triangle, that matches the colour of the part, shows in the lower corner of the score. This means that the master has a 'white' corner, concealing the other colours in the master, but enabling them to be seen in the parts. Figure 8 shows the red part, indicated by the red triangle in the lower left-hand corner. The opaque parts of voice 1 and percussionist 2 can be seen, yet the red part is yet to come into view. The coloured corner enables the instrumentalist to recognise their part. Also noted on Figure 6 is the removal of the horizontal swipe bar at the bottom of page, and the score information along the top and the orange playhead that can all be seen in Figure 10. These are other new functionalities of the ScorePlayer, developed for when scores are projected in a way where the audience does not need to see the timer, timeline, playhead or other iPad information. Also, a black background has been added at the start and end of each piece, to limit the screen luminosity at the beginning and end of movements.

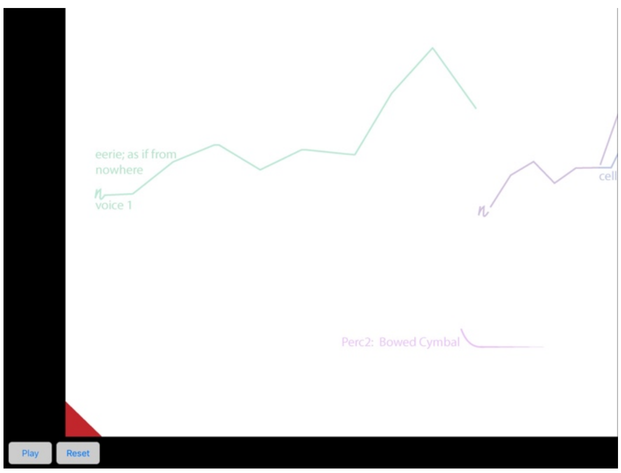


Figure 8. The start of Act 1 of Speechless, showing the red part, with the triangle in the corner indicating the correct part, as well as the removed playhead, timeline and other information from the top.

The workflow developed between the composer and copyist during the Speechless workshop period was unique for the workshopping of a significant, large scale, graphically notated, animated score. Once an agreed upon hierarchy had been established in the Illustrator versions of the score image files, an effective and efficient workflow between composer and copyist - and occasionally programmer - began to develop. Eventually, a very fast turnaround on corrections and changes was established. The composer would sit close to the conductor, following the score on her own, networked iPad and giving verbal notes to an assistant who would write on a Google Doc open on both the assistant and the copyist's computers. Using screenshots, notes and time codes, changes would be noted on the Google Doc, and seen by the copyist in real time. This meant that all changes to the score were happening as the piece unfolded, meaning that with the new Illustrator scripts automating part extraction, new parts could be ready and sent out as a score update to the iPads for the next rehearsal after a tea break. These kind of rapid changes would be impossible with typical paper or graphic scores.

SPEECHLESS SCORE UPDATES

DATE	NAME/PART	Score VS 1 Page	Score VS 2 Page	CHANGE
20-Jul	Cat Hope	overture	overture	Overturne additions - theremin. CH to do. Dan to check layers etc
		19	19	Add "as if from no-where" as marked
		20	20	Make wider space in green part, as marked
		21	21	Add blue vocal part as marked
		21	21	Blue wider, as marked
		21-22	21-22	All joins as marked
		22	22	New brown part as marked
		23	23	Joins as marked
		23	23	Wider gap in green part, as marked
		25	25	Return cyan to red
		27	27	Add text, "very low" as marked
		27	27	Add G as marked
		27-28	27-28	Add text to blue part
		27-28	27-28	Shorten vocals as marked

Figure 9. A screenshot of the Google Doc communicating changes from composer to copyist while rehearsal was taking place. A highlighting system was used to prioritise urgency.

Another important development from the project was the integration of stage management cues. A separate part was created for the stage manager to call lighting and sound cues during the showing performances. While these are currently quite simple, as seen in Figure 8, they will become quite complex, and even more useful in the full production of the work. They take advantage of the tight synchronisation of the Decibel ScorePlayer, and ensure that cues align exactly to the music at all times.

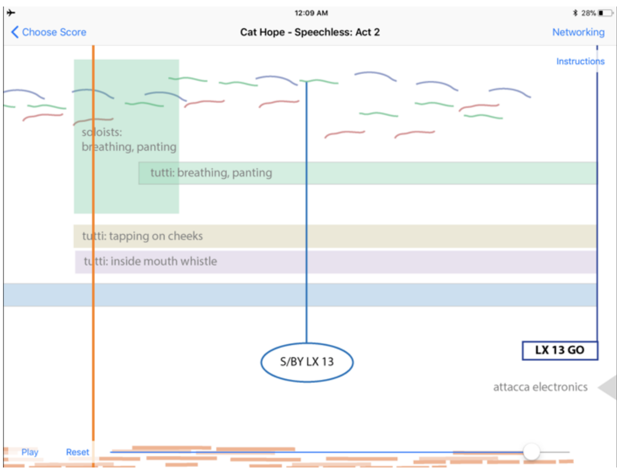


Figure 10. Screenshot of the iPad with the stage management part selected, showing lighting/sound standby and effect cues. Note how other parts are opaque.

The process of having a digital copyist for the Speechless workshop period has informed several potential future directions for the ScorePlayer in regards to the integration of notational media. The export algorithm could be revised to export parts by hex colour code, rather than layer. Assuming the composer retains the trend of using a set hex colour or range (for example 0x00FF69B4 to 0xFFFF69B4), this would allow for the layering of material in a more nuanced and compelling manner while retaining the convenience of batch export of parts; although the ability to quickly hide and show parts may be complicated by this. Overall, it is clear that this role within this project will continue to offer opportunities to develop practical digital graphic scoring methodologies, and methodologies for creating complex works with Decibel ScorePlayer as a delivery medium.

7. CONCLUSION

This paper has examined the workshop period of a new, animated notation opera Speechless and the new developments to the Decibel ScorePlayer that have come about as a result. These included new scripts to enable the extraction of parts from the Illustrator program into the ScorePlayer file format, new bulk update functions on the network, the use of a web server for score dissemination, new naming and labelling protocols for hardware and software, and new projection mode capacity.

The role of the 'digital copyist' demonstrated the ability for the copyist to bridge communication between the composer and programmer, and facilitate creative ideas into technical insights. This role contributed to the development of master score hierarchies, part identifiers and protocols. It also allowed for stage management instructions to be incorporated as a part in the score, and parts with theatre presentation taken into consideration were developed – a first for Decibel Score Player.

Speechless is the first opera written with an animated notation score, but at the time of writing, it is yet to be fully staged. The workshop period was a vital part of the works' development in the way it enabled further technical developments of the Decibel ScorePlayer in addition to the Speechless score, including its delivery to over 75 musicians and technicians in a series of rehearsals and showings.

8. REFERENCES

[1] Vickery, L "Screening the Score" in C. Hope (Ed.) *Audible Designs*, PICA Press, 2011.

[2] C. Hope, A. Wyatt, and L. Vickery, "The Decibel ScorePlayer: New developments and improved functionality," in *Proceedings of the International Computer Music Conference (ICMC'15)*, Denton, TX, USA, pp. 314–317, 2015.

[3] A. Wyatt and C. Hope, “Animated music notation on the iPad (or: Music stands just weren't designed to support laptops),” in *Proceedings of International Computer Music Conference (ICMC’13)*, Perth, Australia, pp. 201–207, 2013.

[4] NSURLSession - Foundation | Apple Developer Documentation. [Online]. Available: https://developer.apple.com/documentation/foundation/nsurlsession?language=objc

[5] Cache-Control - HTTP | MDN. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control.

[6] S. James, C. Hope, L. Vickery, A. Wyatt, B. Carey, X. Fu, and G. Hajdu, “Establishing connectivity between the existing networked music notation packages Quintet.net, Decibel ScorePlayer and MaxScore,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, A Coruña, Spain, 2017.

[7] eLinux.org, *RPi Hardware*. [Online]. Available: https://elinux.org/RPi_Hardware

[8] PHP, *List of php.ini directives.* [Online]. Available: http://php.net/manual/en/ini.list.php

[9] Adobe Systems Inc., *Adobe Illustrator CC 2017 Scripting Guide.* [Online]. Available: https://wwwimages2.adobe.com/content/dam/acom/en/devnet/illustrator/pdf/AI_ScriptGd_2017.pdf

[10] C. Hope, “New Scores for Electronic Music: The Possibilities of Animated Notation,” *Computer Music Journal*, vol. 41, no. 3, pp. 21–35, 2017.

9. APPENDIX: NEW NETWORK COMMANDS

/Server/RegisterScores (*scoreName composerName scoreType version*)...

Sent to the server from the client iPads when they first connect. The client sends four arguments for each score installed on the device: the score’s name, composer, type (for example “ScrollScore” for a scrolling score), and version number. (This is “0” if unspecified by the score file.) This is used by the server to keep a list of scores that are common to all of the connected devices. Only these scores are available for loading via a network command, so it is crucial that all devices have the same version of any score that you wish to load in this manner.

/Server/ScoreList (*scoreName composerName scoreType version*)...

Sent to the client iPads from the server whenever the list of common scores changes. (When one of the iPads connects or disconnects.) This ensures that the user interface avoids showing any scores that are not common to all of the devices, and so cannot be loaded via network command.

/Server/LoadRequest *scoreName composerName scoreType version*

Sent to the server from an iPad client or an external device to request a change of score. This is the command that is invoked by selecting a new score in the score change window of the ScorePlayer.

/Server/RequestRejected

/Server/RequestOK

Sent from the server to an external or iPad client in response to a LoadRequest message. The most likely reason for a negative response for an iPad is that a new device without the selected score just joined the network. (There is only a very small window in which this can happen before the list of common scores is updated network wide.) The iPad client simply shows a dialogue box announcing this. For an external, the most likely reason is that one of the parameters didn’t exactly match the expected parameters for the score.

/Score/Load *scoreName composerName scoreType version*

Once a successful LoadRequest has been made, the server sends this command to all of the iPad clients. This prompts the new score to be loaded on all of the devices, and blocks any control messages (play, seek, reset) from being sent until all iPads have finished loading the new score, or until a five second timeout has been reached. It also prevents new clients from connecting during this period.

/Server/LoadOK

Sent by the iPad clients to the server to let it know that they have finished loading the new score.

/Server/GetScoreList

Can be used by externals to retrieve a list of available scores. This prompts the server to respond with a /Server/ScoreList message which is usually only broadcast to the iPad clients.

SCORING FOR CONVERSATION

Aaron Finbloom
Concordia University
aaron.finbloom@gmail.com

ABSTRACT

This paper discusses the prospects of using verbal notation to score live conversation. It defines a practice of conversation scoring that lies in-between two poles of structured conversations 1) where the content is entirely scripted, and 2) in which a conversation is structured primarily based on an initial set of static conditions (ex. location, time, roles, etc). By working in this middle-ground, conversation scores push conversation to new pedagogical, formal, and methodological limits, while retaining critical elements of conversation such as: spontaneous interruptibility, investment in a subject matter, and a non-linear yet quasi-coherent thought pathway or topic. This paper will discuss notable examples of event-scores both as a means of distinguishing this practice from other verbal notational practices, and for the purposes of elucidating key notational methods which have influenced this practice. The bulk of the paper will then go on to discuss various types of conversational semantic (and para-semantic) directives and end by discussing mechanisms for sequencing these directives. It is my hope that by expanding scoring into a live conversational field, that the practice of conversation itself can be expanded by adopting notational methodologies and aesthetic components that allows us to conceive of conversation as not entirely bound by its content, but defined by its dynamic movements and performative parameters.

1. INTRODUCTION

In the Fall of 2015, I began to re-shape my current artistic practice in the direction of scoring for conversations. This development took place alongside my adviser Sandeep Bhagwati at Concordia University as we undertook an independent study called "Scoring Conversation" aimed at "translating" contemporary avant-garde music scores into conversation pieces. The description of this independent study reads:

> "A score is traditionally understood as a visual method of transcribing music; however, in the past 80 years artists have begun to explore alternative methods of scoring that complicate dominant paradigms in western musical notation. These alternative scorings re-think and re-map the relationship between what is played and notations that direct what is played. This course will look at these contemporary scoring techniques and theoretical disruptions to traditional scoring and begin to experiment with how to apply these techniques to conversation (i.e. semantic dialogue). The following are some questions that will be explored: Can one score a conversation that is both structured and spontaneous? How can the practice of conversation be expanded and diversified through scoring? Is some fundamental quality of conversation (authenticity, spontaneity, depth) lost when a conversation is scored?"[1]

The practice quickly began to envelop a variety pedagogical fields and performative methodologies as I began experimenting with conversation scoring at residency programs, social occasions, workshops and university courses. Soon, it soon became clear that unique notation methodology was beginning to develop that responded to the nature of conversation – emergent, spontaneous, situative, non-linear, non-predetermined. It is the aim of this paper to describe these notational practices as they correspond to the emergent quality of conversation.

2. CONVERSATION

It is precisely these conversational qualities articulated above that presents the greatest creative challenges to the practice of scoring for conversation. In order to more precisely explicate conversation's emergent qualities, I draw on the Russian linguist Lev Yakubinsky's account of interruptibility:

> "One might say that to a certain extent mutual interruption is characteristic of dialogue in general. Our participation in dialogue is determined by our expectation of being interrupted, by our awareness that an interlocutor is preparing to respond, by our fear that we might not be able to say all that we want to say." [1]

The possibility of being interrupted and the awareness of a rejoinder being formed simultaneous to one's

[1] The documentation of this independent study is not published but is available upon request.

utterance, implies that a conversation cannot be entirely predetermined or planned, and suggests that conversation is improvisational. The practice of scoring for conversation, as I define it, must carve a delicate balance between elements that are overly scripted, which close down a conversation and do not allow conversational content to spontaneously emerge, and an opposing problem of scores which create external conditions for conversation to occur but do not directly prescribe conversational content in the moment it is occurring. It is in the creative play of this middle-ground that conversation scoring lies – this middle ground which shares territory with the concept of "structured improvisation" and Bhagwati's term "comprovisation", defined as, "musical creation predicated on an aesthetically relevant interlocking of context-independent and contingent performance elements" [2]. Outside of the practice of conversation scoring, the vast manifold of conversational aspects – content, mood, tone, rate and duration of interruption, etc – are mostly contingent upon the given performance and various unplanned contextual conditions; however in conversation scores, some of these elements are aesthetically, or "consciously" [2], set-in-place, thereby creating a field of deterministic and indeterministic elements specific to each conversation score.

3. AIMS

As one expands scoring practices to include conversational elements, the question of intention or aim frequently arises. While I cannot speak towards the aim of all artists who score for conversation, I can speak towards some aims that I associate with this practice:

1) To develop a bilateral relationship between conversational content and context. For example, a given instruction that shapes the context of discourse (i.e. only speak in questions) will uniquely structure the content of what occurs. Likewise a conversation about ethics might call for a theatrical situation to help inform its content. Both content and context can condition each other and each score creates a structure that allows for a unique opening of these dual directional causal pathways.

2) To allow for marginalized methodological elements (ex. materiality, embodiment, nonsense) to begin to inform conversational practices. One example of this is within the culture of philosophical or intellectual conversation which assumes behaviors such as: mellow intonations (not too loud), somewhat passive bodily movement (sitting in chairs, standing at lecterns), non-excessive emotional intimacy, sentences which make sense, etc. By bringing in these margins not only are more people able to engage in a given practice (persons who might feel isolated from the conversation practice's center), and not only is the conversational content itself expanded through a methodological widening, but additionally, the aims and consequences of a particular practice can be deepened, re-framed, or revitalized.

3) To create an aesthetic container for a conversation to form which has a particular style, character and feeling. In each score there is a unique combination of the conversational content and the assortment of methodological cues (gestures, instructions for speech, movements, etc). At some moments of a conversation the mood may be serious and somber while people discuss morality, at other moments it can become playful and lively while discussing politics. The entirety of a given score aims for an integration or cohesion of both aesthetic content and conversational content (where this can include not only the conversation topic, but emotional, gestural or embodied factors). An ideal conversation score will create conversational pathways and methodological turns that form a cohesive aesthetic afforded by the activation of the piece's structure and rule-set in conjunction with the immersive decisions formed by the players.

4. IN-BETWEEN SCRIPTS AND SCORES

There is a rich history of using word-based notation to score events which touch upon the notation methodologies I am exploring; however, some of these notation methods fail to score *within* a conversation and merely create the conditions for a conversation's emergence. One can look at some classic Fluxus pieces such as George Brecht's *Drip Music (1962)*, or Alison Knowles's *Proposition #4 Child Art Piece (1962)*, wherein notation is used to define parameters for an emergent event. Let's take Friedman's *Restaurant Event (1964)* as an example.

> "Dress as badly as possible. Wear surplus clothes, tattered shoes and an old hat. Go to an elegant restaurant. Behave with dignity and exquisite decorum. Request a fine table. Tip the maitre d' well, and take a seat. Order a glass of water. Tip the waiters, the busboy and staff lavishly, then leave." [3]

This score facilitates sets of actions, many of which implicate conversation (one example of which could be a discussion that occurs between the poorly dressed individual and the restaurant staff). The score creates conditions that surround these conversations, frames them, and supplies them with possible content; however, the score's notational content, does not direct the particular moments of conversation. In the practice of scoring for conversation, I am invested in departing from this tradition of event scores, by creating more specific parameters for conversational content that works on conversation while it is happening.

There are also traditions of utilizing scripts to facilitate scripted or scored conversation which have the opposite problem of organizing semantic content which becomes too tightly bound to its instructions. A traditional theatrical script will indicate which words must be spoken and in what order. Each script differs in the para-semantic content that is organized around the speech, i.e. a given sentence can be spoken with various tones, moods, settings, and bodily and gestural variations. In fact, even a field such a Conversation Analysis, which has created transcription methodologies to account for these para-semantic cues [4], there is still room for some improvisation; there is always some degree of contingency if these transcriptions were to be performed. However, in all these examples, participants performing these scripts are not free to determine the conversational content. As defined above, for a conversation to be a conversation, for it be interruptive and therefore somewhat spontaneous, one must not know *what* one is going to say, not merely not know *how* one is going to say it. In this sense, deterministic scores that prescribe the precise content of what to say, and when it must be said, foreclose the potential for a conversation to emerge.

Some event-scores do in-fact utilize notation that more directly speaks to the emergent content developed in particular moments of the score's performance, and these scores have been quite influential to the practice of scoring conversation. Some of these works structure their pieces with more detailed instructions alongside more specified sequencing, as seen in works such as Cornelius Cardew's *The Great Learning (1968-71)* or George Maciunas's *In Memoriam to Adriano Olivetti (1962), and* Robert Ashley's *The Entrance* (1965-6) [5]. Seth Kim-Cohen's *How to Write A Text About How to Write a Text Score (And Why) (2009)* [5] is a clear example of a semantic score, and although it is written for monologue, is perhaps the best example of a neo-conversation score that I have found. The most influential event-scores for my practice of scoring conversation have been John White's *Newspaper-Reading Machine (1971*), for its exegetical and textual components, and Douglass Barrett's *A Few Silences (2008)* for its innovative use of participant scoring within the event-score. Both of these pieces were originally written for groups of performers, making them more conducive to conversation, and were "translated" into Conversation Scores by Sandeep Bhagwati and myself in the Fall of 2015 [6].

5. SEMANTIC DIRECTIVES

The practice of scoring conversation utilizes a vast range of verbal instructions which will be discussed at length below. These instructions will hereby be called "semantic directives," which I define as the prescriptive use of language aimed at instructing participants in the meaningful use of words.

1: Read aloud a passage from Plato's Phaedrus
2: Take turns: Person A says sentences beginning with "If I were Socrates I would _____" while improvising the endings. Person B says sentences beginning with "If I were Phaedrus I would ______" while improvising the endings.
3: Present contrasting opinions argumentatively
4: Only ask questions
5: Discuss

Figure 1. Conversation Score Sample #1.

Figure 1 represents a sample conversation score which progresses from highly scripted to minimally scripted elements. Either ends of the score display the limits of conversation scoring discussed above. Round 1 utilizes pre-set conversation content, lines read verbatim similar to a traditional theatre script. In Round 5 the instructives merely indicates that conversation should occur, and the content is conditioned by the implicit setting and the prior rounds leading up to this one. The rounds in-between present three possible midpoints between these poles. Round 4 is a section that leaves open the content and style and gives only a single directive that asks for an interrogative mood via a grammatical directive. Round 3 presents 1) a slightly more prescriptive directive that instructs mood through a direct indication to change mood ("be argumentative") 2) asks for a particular topic to be discussed (politics), and also 3) includes a more structural directive (present contrasting opinions). Round 2 directs the participant to use sentence stems which provides a partially scripted sentence that the participant utters and then fills in with their own improvised content.

These specific directives in the order that they occur in this score help to facilitate a coherent movement of conversational content, mood and form. David Kennedy describes this coherence as a coordination and holding together of multiplicitous perspectives through which meaning comes to be shared alongside a growing complexity and entanglement of the very perspectives that supply this meaning [7, p.210]. This coherence is made possible by conversational investment, by a collective feeling that "something is at stake" in the conversation. The proper placement of semantic directives in the right time, can create responsiveness and help to transfer the content (thematic or emotional) from one round to the next and establish greater coherence and investment. For example, imagine if Rounds 1 and 5 were replaced. This would create an entire conversational thematic buildup that would then be abruptly altered by a passage of scripted text. Rounds 2-3 are attempts to dive into the rising investment by prodding issues that may be at the heart of the interlocutors involved. Quasi-open rounds like #4 are essential as they allow issues which may have strayed from the interests of those involved (via the in-depth directives and the specific direction of the conversation) to be brought back into the discussion.

6. DIRECTIVE GRAMMARS

John Lely in *Word Events*, devotes much attention to the varied grammars that event-scores can utilize calling attention to context, register, process, tense, mode, mood, voice and circumstance, stressing the importance of this work because, "grammatical choices can create very different perspectives on the world; for instance, through a change in one element of grammar, a description of activity can be transformed into a command" [5, p.3]. As I discuss various conversation scoring elements I will touch on some of these distinctions brought forward by Lely focusing on an analysis of key grammatical functions particular to conversation score usage.[2]

[2] Lely devotes a small section titled "verbal processes" which perhaps comes to the closest to my usage of the term semantic directives for conversation; however very little is said towards this practice in this section [7, p. 21].

Conversational practices outside of the scope of scored conversation utilize directives both implicitly and explicitly and create categorical distinctions separating one type of directive from another. Matthew Lippman, one of the founding practitioners of Philosophy For Children (P4C), a practice of structured facilitation to create philosophical dialogue with children, utilizes mental directives such as "reflect" and "imagine" to bring a meta-awareness to the process of thinking and thereby aid the practice of philosophical dialogue [8]. A drama therapy practice might utilize more active, ludic and emotionally-oriented directives which aid patient expression. Conversational practices that foster authentic connection such as Circling [9] use semantic directives that embrace focusing on inner-feelings and what is felt in the moment rather than more topical or information-based conversation topics.

1: Only use sentences beginning with "I feel"
2: Only ask questions
3: Pick a question and discuss
4: Uncover underlying assumptions

Figure 2. Conversation Score Sample #2.

Figure 2 presents a sample conversation score arranged to reflect some varied conversational practices each of which shapes the conversational direction giving it a certain focus, mood, and structure. Round 1 receives inspiration from practices of non-violent communication, therapeutic and authentic relating practices, but also has roots that lie within the linguistic device of personal pronouns. Unlike most words, "I" is a deictic term; its meaning is contextually grounded, as each time someone speaks "I" it denotes a different entity. Emile Benveniste points out that it is this conversational exchange of "I"s that grounds dialogue itself within an interlocking reciprocity of identity markers [10]. As the "I" switches from each interlocutor the unique emotional realms of each also begins to transfer as well, providing fodder for emotional connectivity and an excellent beginning of a conversation score if the intention is to form connection.

The practice of only asking questions derives from a few different cultural sources. The Question Game is featured in Tom Stoppard's *Rosencrantz and Guildenstern are Dead (1990)*; however my usage of questioning in this manner is not an agonic back-and-forth aimed at determining a winner and loser. Constant questioning opens up a field of curiosity which does not close-down in an answer, but rather is re-opened and re-engaged again and again through further questions. Careful listening allows for these questions to build upon prior questions. By scoring the question round after the exchanges of I-statements, this allows for an initial intimacy to provide the fodder for an expansive inquisitive exploration of relational content that emerges from this opening round.

Round 3 utilizes a directive taken from a wide variety of pedagogical practices (from salons, to debate to teams, to classrooms) and functions predominantly as an open directive whose content is determined by the rounds immediately preceding and following. It is important to note how vital open moments such as this can be in the construction of conversation scores as they allow for breath, reflection and spacing. Quite often in the practice of scoring for conversation, too many directives can leave performers awash in a sea of instructions without the ability to speak "freely". Open directives create an opportunity to explore the terrain opened by more rigorous structures of the preceding rounds.

Finally Round 4 asks for a deeper investigation, using techniques borrowed from P4C. Other examples of P4C techniques include, "pointing out the necessary implications of a statement," "identifying a contradiction," and "restating a point as a logical proposition" [7, p.148]. Since P4C is largely a rationally focused practice and helps develop subtle depths within an already-established topic, it helps to utilize these techniques once a conversation already has gathering ground and contains a central topic of discussion.

7. PARA-SEMANTIC DIRECTIVES

In addition to directives that instruct conversationalists in semantic content, various para-semantic directives can greatly enhance the potential range and depth of a conversation score. Many of these para-semantic directives play with structural factors regarding time, spacing, frequency and number of players.

The gap between utterances is one of the most sensitive aspects of a conversation to score via para-semantic directives as it contains the mechanism by which conversationalists listen and respond. Dmitri Nikulin, in *Dialectic and Dialogue* speaks of this interruptive gap as, "a pause taken by the speaker in order to allow the other to act and react against the original and provocative action, thought, or utterance" [11, p. 98]. The conversation gap between utterances is far from empty, but rather it both signals and gives time for the conversational responses which build a conversation via a back-and-forth procedure. By adjusting the length of this pause, the frequency of pauses, their affective quality, mood or tone, one can begin to design the degree of responsiveness within a conversation.

1: Allow long gaps of silence between utterances
2: Two persons discuss a topic brought up from the prior round while all other participants interrupt with one-sentence questions or clarification
3: One person give a monologue
4: Two persons give simultaneous monologues
5: Write

Figure 3. Conversation Score Sample #3.

Figure 3 provides a sample conversation score that displays some common para-semantic directives that I use in designing conversation scores. Round 1 utilizes a technique borrowed from Quaker Meetings of providing long gaps of silence between speech to allow for a greater reflective period of inward analysis. By providing this in the opening of the score, players begin with an emotional attunement and quiet contemplative togetherness.

Round 2 divides participants, by giving only two players a chance for unrestricted conversation, while the rest of the players can give only brief interruptions in the form of either questions or clarification. Given that conversation is predominantly a monophonic action (i.e. conversation is a series of back-and-forth monophonic rejoinders, and only rarely and briefly does conversation erupt into polyphonic, simultaneous utterances), conversations with many participants can easily leave someone out, or else take a long time to allow everyone the opportunity for expression. In which case, breaking-up conversation into smaller groupings can be a vital directive in conversation scores, as this technique allows conversations to move more deeply with greater rapidity. The particular advantages afforded by the directives of this round is that all given participants *can* speak (it is not an abrupt shift of entirely active to passive); rather there are two main speakers (mostly uninhibited in their speech) while the rest of the participants play at giving quick rejoinders.

Although monologue has elements that are antithetical to conversation in that, “it does not expect an answer and thus does not presuppose the other to respond and ask questions” [11, p.82], it can still be a useful antipode to conversational interruptibility if used strategically. Monologue is the defining conversational attitude of academic lectures, conferences and speeches. By polarizing the role of speaker and listener, monologue creates the capacity for vast hierarchical displacement, but also for uninterrupted utterances and a kind of calm that is provided by knowing one is safe from the somewhat anxiety-producing conversational fact of interruption. Monologue provides a temporary, yet perhaps necessary, facade in the face of interruptibility’s reminder that we are never stable solid entities, that the dialogic intersubjective state is actually the ontological grounds of our very subjectivity [7, p. 81-6], [11, p.103-5].

An opposing extreme of singular monologues is the difficult-to-attain, polyphonic simultaneity of dual monologues. This practice of continuously speaking *while* another is speaking is found in brief moments in both Linda Griffith’s *Age of Arousal (2004)* and Glenn Gould’s *Solitude Trilogy (1967-77)*, but departs from these examples as my utilization of this practice asks for simultaneous speaking and listening which eradicates the temporal divide that separates these two activities. A hard-to-achieve radical togetherness is formed in this activity; however resistance usually occurs and much skill is required to work out the nuanced tempos and dynamics of voice that make this achievable. By placing this polyphonic round after a singular monologue, it allows for another player to seamlessly come into this round, by adding to the threads of the preceding monologue.

Writing, in Round 5 of Fig. 3, creates another kind of conversational polyphony, as each participant can express thoughts simultaneously but without significantly influencing one another. This silent quality of writing has long been considered one of writing’s greatest assets and makes writing’s distributability radically different from that of speech [12, 13]. While writing falls on the outskirts of a conversational practice as it is predominantly non-verbal and non-interruptive, it nonetheless can be strategically inserted into conversation practices to provide dynamic gaps in audible expressive content, to pause the conversational competition for attention and voice, and to force conversation into a period of isolated individuated expression around a given topic, which can then later be integrated into the verbal conversation.

8. GAME MECHANISMS

A vast array of event-scores and avant-garde music compositions utilize gaming mechanisms (such as timings, cards, turn order, etc) to sequence rounds and actions. From Cage’s chance encounters with the *I Ching*, to George Macianus’s *In Memoriam to Adriano Olivetti (1962)* which utilizes found tapes from adding machines to determine the ordering of actions for a series of rounds, to Michael Parson’s *Walk (1969)* which uses randomly assigned numbers to determine walkers speed and frequency of pauses [5], these mechanisms can create a greater degree of interactivity in scores by resisting linearity and making the sequencing techniques necessitate player interaction. This is particularly important for conversation scores, as conversation’s emergent quality necessitates nonlinearity (in “organic” conversation one doesn’t know beforehand in which order semantic content will be uttered and arranged). Prior to this section, I have discussed how particular semantic directives help to achieve this nonlinear quality in conversation by creating openness and spontaneity *within* a particular round or moment of conversing; however, this nonlinearity can also occur in the structuring of the rounds themselves, the way in which one directive is chosen, and the method by which the score moves from one directive to another.

Figure 4. Example of an individual card hand from *Oscillations of One-to-Many* (2017) by Hannah Kaya and Aaron Finbloom.

What does determines the transition from one semantic directive to the next? One option is that rounds can be timed, and timers can be used to indicate when switching should occur. In some instances this can aid a conversation by forcing it to advance to the next stage even when one doesn't feel ready to advance, thereby moving a conversation away from its felt necessity and uncovering challenging, uncomfortable and unanticipated moments. Another option, which has proven to be quite fruitful, is to explore inherent mechanisms within a round that could be utilized for switching. In my piece *Deictic Dialectics (2016)* each rounds implicates different players in different roles. The responsibility to switch rounds is either felt out by one of the players as they consider when the round needs to advance (perhaps when the conversation is in need of movement) or they feel into a directed approximate timing. In these cases the round switches can be more fluid which allows the conversation to stay within a topic and not get excessively sidetracked by an abrupt transition. This has been further enhanced by initiating the switch via a directive for bodily movement, which signals a new scene or platform for dialogue and can allow a conversation to remain verbal, while peripherally and simultaneously identifying an embodied cue.

Figure 5. Example of a pooled card hand from *Oscillations of One-to-Many* (2017) by Hannah Kaya and Aaron Finbloom.

Many event-scores and conversation scores also utilize cue cards to display the directives, which makes the interactive transitions of rounds even more rule-based and formulaic. Some examples of this include Ellen Burr's *Ink Bops (2017)* or John Zorn's *Cobra (1984)*. In each card-based gaming piece, the rules governing the use of the cards and the mechanisms of card sequencing differ from round to round and even from card to card. In some conversation pieces players can have a hand of cards, each representing a conversation cue to be activated only *by* the card holder, but potentially *on* either herself, another interlocutor or the entire group (see Figure 4). Another option is to create a pooled hand, whereby all the players share an open hand and any player can, at a given time, play any card from this hand (see Figure 5). In the former, card choices are activated by one player's individual discretion which then alters the dynamic system; in the later, all players have the capacity to play a given card at any time which allows for a more collaborative conversational modality. A number of mechanisms can also be deployed for determining how cards can be distributed, chosen, discarded, etc. For example in some pieces, cards can be used twice before being discarded, in others cards are never discarded and can be used any number of times. In addition some pieces provide players with the opportunity to generate their own cards thereby giving participants the opportunity to design directives unique to the conversation that is occurring.

9. TECHNOLOGY

The above examples and theoretical implications of conversation scoring are presented in a somewhat preliminary manner given that the practice of conversation scoring is still within an embryonic phase of development. As such, the research that this paper provides is intended to lead towards the eventual development and realization of scores made for conversation. Up until the present, the actual number of implemented conversation scores are few, and their main method of presentation derives from their Fluxus background – on sheets of paper giving instructions – or from game pieces like *Cobra* – with cue cards giving instructions to performers. I anticipate that the next stages of conversation scoring development will most likely follow from implementing diverse digital and computational technologies.

One advantage of developing scores with greater technological implementation is the increased ability to reduce extraneous physical elements involved in the performance of the scores. As of now most scores demand for someone to physically turn a page, hold up a card, or write down a new instruction, all of which create theatrical assumptions that these movements themselves carry meanings. By displaying the scores on a screen or with headphones this would allow for directions to shift seamlessly without an added action imparting its own non-intended performative meaning. Additionally technological innovations such as headphones or projected instructions create the potential for a greater range of performative movements, as both reduce a conversationalist's necessity to stay in a single place to see an instruction or to be burdened by holding cards or sheets of paper. Furthermore by giving individually microphones to interlocutors and supplying audience members with headsets, this can create the potential for the audience to choose which conversationalist they are listening to, giving added interactivity to a score's performance.

Another major advantage afforded by involving technology lies in the ability to play with imbricating textuality into the conversational pieces. One piece that I created which plays with this potential, *Memory Pharmacy (2014)*, was inspired by a passage on the origins of writing at the end of Plato's *Phaedrus*. The process of creating this piece began as I replaced Plato's interlocutor's lines of agreement with a semantic directive asking, "what do you think?" The textual passage was then read aloud by participants, however, when they reached these moments in the text, the semantic directives would pull the readers out of the text and prompt them to conduct a spontaneous conversation about the passage. After either reaching a discursive conclusion or achieving boredom, the interlocutors would then return to reading the text aloud. I then took audio recordings of these conversations and transcribed them to create a new text that included both original Platonic passages and the interlocutor's responses. I repeated this procedure a number of times until I obtained an extensive supply of responses, after which I then attempted to combine these responses into a unified text. I found this task of unification difficult, if not near impossible, for each time this procedure was enacted, different conversational choices were chosen. For example, sometimes a passage was agreed with, other times a repetition was asked for, or in other instances an interlocutor emphatically disagreed with Socrates. I soon realized that if I wanted to create an amalgamated text that honored these conversational divergences, that I would have to take advantage of software used for designing such works of electronic literature – Twine.

Socrates
Do you only consider who the speaker is and where he comes from, or do you not more rightly consider whether his words are true or not. So then, tell me, are the words of King Thamus true or false? What do you think? Does writing hinder remembering?

Repetition
Phaedrus: "Can you repeat the passage please?"

Disagree
Phaedrus: "I'm not sure I agree Socrates"

Agree with Socrates
Phaedrus: "Your rebuke is just; and I think that Thamus is right in what he says about letters."

Figure 6. Segment from *Memory Pharmacy* which depicts conversational choices.

I used Twine to combine the conversational transcriptions and represent divergent dialogical pathways by making an interactive conversation game whereby interlocutors would read aloud the text while choosing which conversation pathway they wanted to embark upon. In Figure 6 we can see an example of how one moment of these multiplicitous conversational pathways were codified using Twine. The bolded text indicates the speaker. The italics tell the reader the type of option a given conversational pathway opens. The blue lines must be spoken aloud but also clicked on, upon which a new conversational passage is opened. Twine creates an interactive textual interface that allows for polyvocality and non-linearity by not forcing authorial decisions such as which conversational pathway deserves greater attention, focus or dominance; rather, users are given the agency to chose a given conversational path. However, somewhat problematically, Twine forecloses the potential for users to generate new conversational pathways, and creates a conversation that is mainly the re-reading and re-enacting of previously constructed utterances. I attempted to remedy this by using open-ended directives within the piece to create opportunities for the conversation to generate new possibilities (see Figure 7) and to step outside of the pre-programmatic text. Integrating semantic directives into the interactive story-telling platform allowed users to create new conversational content and thereby created the potential for a spontaneous dialogue to form alongside a textual interface. *Memory Pharmacy*'s use of technology afforded the opportunity to play on the edge of a spontaneous, live, oral dialogue alongside visual, textual, static transcriptions. The digital interface allowed for the creation of a dialogic game which integrated transcriptions with conversation, writing with speaking and non-linear pathways with pre-determined directionality. This complicated dance between these elements would have not have been possible without the digital interface utilized.

Phaedrus *Discuss*

keep discussing until you come to an agreement or you grow tired of discussing

Socrates Shall we proceed?

Figure 7. Segment from *Memory Pharmacy* which depicts departure from the text.

I anticipate that the next stages of scoring for conversation will be set on various technological platforms that help expand the potential of this alternative notational process and further integrate and intertwine oral and written discourses. Writing affords one the ability to see, dissect and rearrange ideas more easily than oral discourses; however it also detracts from the speed of utterance possible with spoken discourse, as well as the wider range of bodily arrangements that one can perform while speaking. I am most excited about the potential to integrate text-to-speech technologies into conversational performances, which would allow for participants to write via speech – to utilize the benefits of writing without detracting from the embodied fluidity of speech. Moreover, were an oral conversation to be quickly translated to a textual medium, it could then be analyzed using data analysis tools and AI. One could search for patterns within the conversation and generate directives based on these patterns. For example, one could set parameters for how many questions need to be asked during a particular round and then utilize analysis of the conversation already produced to then determine the next directive that would appear on the screen. Conversational scoring such as this is likely to push this nascent practice's potential to create new conversational situations, and find new ways of dynamically investing conversational content.

10. CONCLUSIONS

My hope is that this article has shed some light on the beginnings of a conversation scoring practice alongside offering considerations of notational methodologies of such a practice. As far as my research disclosed, this practice of creating conversation scores (which neither creates completely scripted content nor merely creates exterior conditions for conversations to exist) is an innovative practice and in this sense I believe conversation scoring to be an emerging field of composition. The focus on this paper has been in looking at conversation scores that 1) feature directives and sequencing which help to foster an emergent and quasi-spontaneous conversational arc and which 2) aim towards an aesthetic coherence of content, mood and form. Therefore, this paper presents a rather narrow conversational scoring focus, and the variety of notational techniques for conversation scoring remains quite open and in development. There are many semantic and para-semantic fields that I have not discussed, including: gestural directives, props, roleplay, location, durational pieces, etc. There are also a great many notational systems that I did not discuss which include graphic or imagistic notation for spatial arrangement, gestural notation, or even, as seen in some Conversation Analysis practices, notations for eye gazing [14]. It is my hopes that in the forthcoming years of development and dissemination, that this practice will receive more attention, that more artists will devise conversation scores, and that a wider variety of writing regarding its techniques will become available.

Acknowledgements

I would like to thank my adviser Sandeep Bhagwati for all the time, support, feedback and creative input that he has put towards this developing practice. I would like to thank Hannah Kaya, Bronwen Moen and Emma Lee Iverson for all their continued and invaluable contribution as conversation score participants and for their creative collaboration.

11. REFERENCES

[1] L. Yakubinsky, "On Dialogic Speech." *PMLA* vol. 112, no. 2, 1997.

[2] S. Bhagwati, "Notational Perspective and Comprovisation" in P. de Assis, W. Brooks, K. Coessens (Eds.): *Sound & Score. Essays on Sound, Score and Notation*, pp. 165-177. Leuwen University Press, 2013.

[3] K. Friedman, "Restaurant Event," in K. Friedman, O. Smith, and L. Sawchyn (Eds.): *The Fluxus Performance Workbook*, 2002.

[4] G. Jefferson, "Glossary of transcription symbols with an introduction." In G. H. Lerner (Ed.): *Conversation Analysis: Studies from the First Generation*, pp 14-31. Philadelphia: John Benjamins Publishing Company, 2004.

[5] J. Saunders and J. Lely, *Word Events: Perspectives on Verbal Notation*. New York, NY: Continuum, 2012.

[6] A. Finbloom, S. Bhagwati, "Question Animals," "A Few Silences," "Exegetical Reading Machine," in M. Firunts, D. Snelson: *Present Tense Pamphlets*. Evanston, IL: Northwestern University, 2016.

[7] D. Kennedy, *Philosophical Dialogue with Children*. Lewiston: Edwin Mellen Press, 2010.

[8] M. Lippman, *Thinking in Education*. Cambridge: Cambridge University Press, 2003, pp 139-161.

[9] The Circling Institute, "What is Circling?", 2017. [Online]. Available: http://www.circlinginstitute.com/circling/what-is-circling/

[10] E. Benveniste, *Problems In General Linguistics*. University of Miami Press, 1971, pp. 217-230.

[11] D. Nikulin, *Dialectic and Dialogue*. Stanford University Press, 2010.

[12] W. J. Ong, *Orality and Literacy*. London: Methuen, 1982.

[13] Plato, *Phaedrus*. Ed. Alexander Nehamas, trans. P. Woodruff. Indianapolis, IN: Hackett Publishing Co, 1995.

[14] F. Rossano, P. Brown, S. Levinson, Stephen, "Gaze, questioning, and culture" in J. Sidnell (Ed.): *Conversation Analysis Comparative Perspectives*, pp. 187-249. Cambridge University Press, 2009.