TENOR

INTERNATIONAL CONFERENCE ON TECHNOLOGIES FOR MUSIC NOTATION & REPRESENTATION

PRISM Laboratory
(Perception, Representations, Image, Sound, Music)
Marseille, France
May 9-11, 2022

Seventh International Conference on Technologies for Music Notation and Representation

Edited by
Vincent Tiffon, Jonathan Bell
and Charles de Paiva Santana

TENOR

INTERNATIONAL CONFERENCE
ON TECHNOLOGIES FOR MUSIC
NOTATION & REPRESENTATION

The Seventh International Conference on Technologies for Music Notation
and Representation – TENOR 2022
Marseille, France May 9-11, 2022

Edited by Vincent Tiffon, Jonathan Bell & Charles de Paiva Santana.

Proceedings published by Les éditions de PRISM
ISBN 979-10-97498-03-0

Logo design: Nicolas Taffin, updated for 2022 by Jonathan Bell
& Charles de Paiva Santana.

http://tenor-conference.org
https://tenor2022.prism.cnrs.fr

Preface

The 7th International Conference on Technologies for Music Notation and Representation (TENOR) is hosted by the interdisciplinary art-science laboratory PRISM (CNRS), in collaboration with the InCIAM Institut, Aix-Marseille University (AMU), Université Côte d'Azur (UCA).

For this 7th edition, in addition to the usual topics of the TENOR conference, we propose a focus on *Comprovisation* and its musicological dimensions, as well as on AR, VR, 3D technologies applied to music writing and interpretations.

In addition to its scientific collaborators, the Salle Musicatreize and the Marseille Conservatory contribute to the organization of the artistic side of the conference. For the Marseille Conservatory on May 9, several proposals will be offered around *Comprovisation*. On May 10, we have the pleasure to gather on the stage of *Musicatreize* the excellent vocal ensemble *Neue Vocalsolisten* for the first reading session, and four instrumentalists from Marseille for the second reading session.

Many thanks to Craig Vear and the DigiScore project of the European Research Council (ERC), as well as to the TENOR Network Funding, and the AFIM (*Association Francophone d'Informatique Musicale*).

May 2022

Vincent Tiffon
Jonathan Bell
Charles de Paiva Santana
For the TENOR 2022 organizing committee

Organization

General Chair

Vincent Tiffon	PRISM-CNRS, France

Organizing Committee

Richard Kronland–Martinet	PRISM-CNRS, France
Mitsuko Aramaki	PRISM-CNRS, France
Jonathan Bell	PRISM-CNRS, France
Charles de Paiva Santana	PRISM-CNRS, France
Javier Elipe-Gimeno	PRISM-CNRS, France
Christine Esclapez	PRISM-CNRS, France
Julien Ferrando	PRISM-CNRS, France
Dominique Fober	GRAME-CNCM, France
Julie Perret	PRISM-CNRS, France
Jocelyn Roze	PRISM-CNRS, France
Joséphine Simonnot	PRISM-CNRS, France
Adrien Vidal	PRISM-CNRS, France
Sølvi Ystad	PRISM-CNRS, France

TENOR Steering Committee

Sandeep Bhagwati	Concordia University, Canada
Jean Bresson	Ableton, Germany
Dominique Fober	GRAME-CNCM, France
Georg Hajdu	HfMT Hamburg, Germany
Richard Hoadley	Anglia Ruskin University, UK
Cat Hope (**General Chair**)	Melbourne University, Australia

Paper Committee

Mitsuko Aramaki	PRISM-CNRS, France
Andrew Blackburn	Federation University, Australia
Jean Bresson	Ableton, Germany
Juan Parra Cancino	Orpheus, Belgium
Clement Canonne	Ircam-CNRS, France
Myriam Desainte-Catherine	Université de Bordeaux, France
Frédéric Dufeu	Univeristy of Huddersfield, UK
Emile Ellberger	Zurich University of the Arts, Switzerland
Dominique Fober	GRAME-CNCM, France
Ichiro Fujinaga	McGill University, Canada
Jean-Louis Giavitto	IRCAM-CNRS, France
Gérald Guillot	Ecole Supérieure d'Art de Lorraine, France
Daniele Ghisi	Conservatorio di Torino, Italy
Georg Hajdu	HfMT Hamburg, Germany
Richard Hoadley	Anglia Ruskin University, UK
Cat Hope	Melbourne University, Australia
Florent Jacquemard	INRIA, France
Mieko Kanno	Sibelius academy, Finland
David Kim-Boyle	Sydneu University, Australia
Richard Kronland-Martinet	PRISM-CNRS, France
Douglas Nunn	Anglia Ruskin University, UK

Benjamin O'Brien	Parole et Langage Laboratory, France
Charles de Paiva Santana	PRISM-CNRS, France
Philippe Rigaux	CNAM, France
Ryan Smith	Monash University, Australia
Vincent Tiffon	PRISM-CNRS, France
Sølvi Ystad	PRISM-CNRS, France

Music committee

Jonathan Bell	PRISM-CNRS, France
Julien Ferrando	PRISM-CNRS, France
Javier Elipe-Gimeno	PRISM-CNRS, France

Workshop committee

Jonathan Bell	PRISM-CNRS, France
Joséphine Simonnot	PRISM-CNRS, France

Proceedings Committee

Jonathan Bell	PRISM-CNRS, France
Charles de Paiva Santana	PRISM-CNRS, France
Vincent Tiffon	PRISM-CNRS, France

Webmasters

Jonathan Bell	PRISM-CNRS, France
Sølvi Ystad	PRISM-CNRS, France

Table of Contents

MUTABLE GESTURES: A NEW ANIMATED NOTATION SYSTEM FOR CONDUCTOR AND CHAMBER ENSEMBLE

Ciaran Frame, Alon Ilsar, Sam Trolland
SensiLab, Monash University
Melbourne, Australia
{ciaran.frame, alon.ilsar, sam.trolland}@monash.edu

ABSTRACT

This paper outlines the creation of a new real-time scoring work, *Mutable Gestures*, for any combination of chamber instruments with a conductor. The work translates a conductor's gestures into real-time animated notation, relayed to performers over a wireless network to generate musical material for improvisation. Drawing on recent real-time notation works, *Mutable Gestures* proposes a new form of gestural notation creation through the use of the AirStick, a new gestural musical instrument. The creation of this work contributes to the growing field of real-time animated notation, a field that reinterprets the traditional roles of score, conductor, composer and performer.

1. BACKGROUND

1.1 The role of the conductor

Conductors are often viewed as the mediator between musical work and performer, conveying the composer's intentions through a single position of leadership and direction. However, the prevalence of conductors in smaller chamber ensembles has contracted, with many ensembles such as quartets usually opting out of a conductor entirely.

Contrary to this trend, recent findings by Wyatt and Hope suggest that the conductor is in fact desired in an animated notation context within the chamber ensemble, offering greater accuracy and musical insight beyond the animated score itself [1].

With this in mind, it is interesting to posit what other roles a conductor may play in the ever-growing field of animated notation, perhaps ceding some roles like timekeeping and gaining others, like running notation systems or software debugging, particularly in an environment where so-called 'virtual conducting' is already possible through smartphones [2] or infrared technology [3].

Thus, the combination of new technologies in conducting and the changing role of the conductor suggests fertile ground for a notated work that generates musical material from the conductor themselves.

1.2 Gesture and notation

Kinetic movement, particularly gestural movement from musicians, is a productive field for real-time notation. The use of motion sensors such as IMUs (inertial measurement units) provides clear input data, offering exploration of causal relationships between action and sound and inferring meaning from gesture.

Work in both Digital Musical Instrument (DMI) and music notation communities of practice has seen an explosion of complex, disembodied musical works that use kinetic movement to produce sound or sound representations [4].

The data provided by a gestural interface in the context of music has been the subject of much related work. Of particular note is Dori's recent work *Arcos*, which captures a cellist's gesture, informing the creation of both action-based notation and sound [5].

Notably though, the source of gesture in musical works is usually a musician, dancer or unspecified performer whose movements and actions are translated into notation [5, 6] or sonic material [7, 8, 9].

It is therefore interesting to note that the conductor is rarely used in a notation or sonic material context, despite being one of the clearest examples of gesture and prescriptive actions provided to musicians.

Conductors have been the subject of gestural scrutiny in music technology, particularly in a motion capture and gesture recognition space [10, 11]. However, this work is almost exclusively focused on musical control and understanding actions and gesture within the conductor's usual scope of practice, such as in a traditional Western Classical concert.

While the use of musicians with gesture data provides intimate composition possibilities, especially in a solo context, the conductor holds a unique perspective and position of power within the chamber ensemble - a power explored in *Mutable Gestures*.

2. MUTABLE GESTURES

2.1 Overview

Mutable Gestures was written for up to five chamber players, and was created from a simple premise of a conductor creating musical content from their baton, for each respective player to play. A conductor gestures towards one or many players in an ensemble, and generates notation. Pointing at a specific player generates live notation on that player's device (such as in in Figure 1), and general gestures produce notes for all players at once.

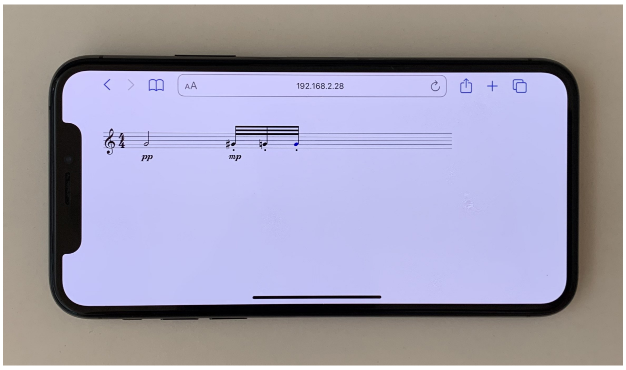


Figure 1. An example of live notation streamed to an iPhone on Safari.

The work is an improvisation system, in which musical material is provided in the form of real-time traditional notation, to be interpreted alongside the conductor's gestures. One does not replace the other — notation cannot replace the nuance in a conductor's gesture, nor can gesture provide a level of detail in musical content that notation supplements. Instead, the work provides a platform for 'comprovisation' [12], where notation and the conductor's gesture provide musical material to be used alongside the players' own inspiration and improvisation in a chamber ensemble context. The degree to which players balance these competing inspirations is up to them.

Attached to the conductor's baton is an *AirStick* (shown in Figure 2), which relays sensor data back to a computer, outlined in Section 2.2. The mapping system in turn makes decisions about what notation appears on performers' devices, outlined in Section 2.3.

Figure 2. The *AirStick* attached to a conducting baton.

2.2 Technology

2.2.1 System architecture

Notation generation begins at the *AirStick* [13, 14], an established wireless gestural Digital Music Instrument (DMI) now in its second design iteration. Attached to the baton, the *AirStick* is capable of determining absolute orientation (the direction the baton is pointing) and linear acceleration (the acceleration of the baton along the three axes local to the device's frame of reference).

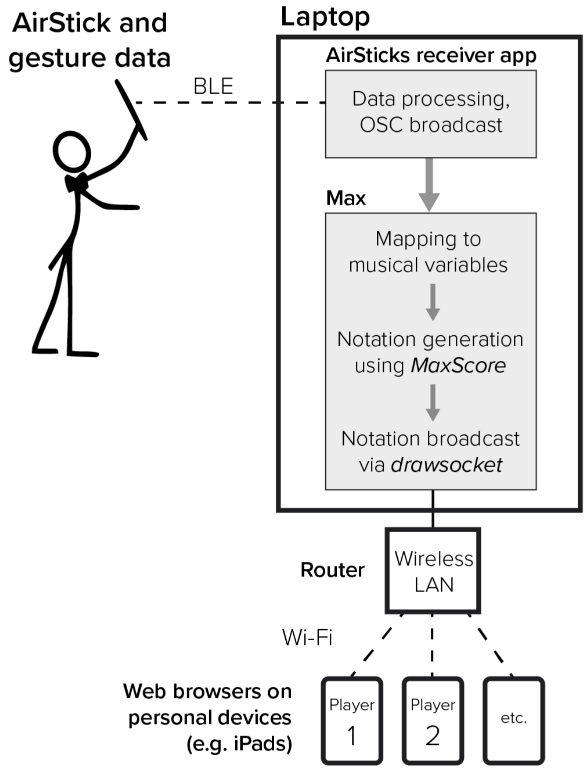


Figure 3. Technical flowchart of *Mutable Gestures*.

The *AirStick* is a gestural instrument by design, and is best placed to deliver these types of gestural data that a conductor conveys, particularly with its natural drumstick-like affordances capturing percussive gestures and orientation data.

The piece is designed to cater for up to five players, in order to distinguish between players from the conductor's point of view, set up in an arc around the conductor. Instrumentation and performer seating positions are calibrated before the performance, to ensure notation is within playing range and using the correct clef.

The *AirStick* communicates via a low-latency Bluetooth Low Energy (BLE) connection to the receiver software on a laptop. Here, sensor data is transformed into readable OSC data, broadcasting sensor data locally to an instance of Max. [1] This sensor data is separated into five key data streams, outlined in detail in Section 2.3.1).

Within Max, this data is mapped to predetermined note values, articulation and dynamics, separated into different parts for respective musicians depending on where the conductor is gesturing towards (described further in Section 2.3). These musical variables are transformed instantly to music notation across multiple parts using MaxScore, [2] then broadcasted to live web pages using Drawsocket, [3] which integrates seamlessly with the MaxScore environment.

Each musician connects to their own 'part' through a unique

[1] https://cycling74.com/products/max
[2] http://www.computermusicnotation.com/
[3] https://forum.ircam.fr/projects/detail/drawsocket/

web address, meaning performers only require a personal device (such as an iPad), and a Wi-Fi connection to the local performance network. This reduces barriers to performing the piece, bypassing additional software and hardware requirements such as the installation of apps in favour of a system that will work with no prior purchases or setup time.

This process is described in Figure 3.

2.3 Gesture and mapping

When the conductor gestures, notation is generated live based on a select number of variables from the *AirStick*. Different gestures produce different notation, loosely based on the norms of traditional conducting in order to create a sense of embodiment of the notation, and therefore sound the conductor is triggering.

This can manifest itself in simple (e.g. big gesture maps to strong dynamics) and complex (e.g. long-term variables maps to piece structure) forms, an example of which can be seen in Figure 4.

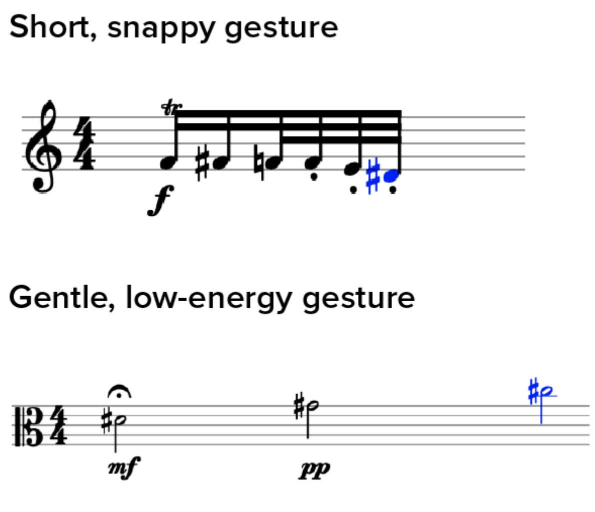


Figure 4. Two examples of generated notation based on different gestural inputs from the baton.

Importantly though, the work does not seek to categorise and learn different styles of conducting, instead choosing to directly 'map' kinetic variables directly to musical ones. This is partly to bypass the adjacent discipline of Machine Learning and gesture (discussed further in Section 3.3), but also to explore how this gestural data might translate to musical material. Indeed, the motivation behind mapping particular kinetic variables to specific notation elements evokes sonification of physical gestures, with the added layers of musical interpretation by conductor, system, and performer adding further lenses through which one might view bodily movement. This point becomes more relevant when we consider the fact that the conductor does not receive instructions around how to conduct with the *AirStick*, nor are they aware of the exact notation their own movements are generating.

The real-time notation is an intentionally imperfect representation of the conductor's gesture, in order to create notation materials that inspire and provoke musical improvisation from performers in conjunction with the gestures they are receiving from the conductor.

AirStick variables are translated from their raw form (described in Section 2.3.1) to notation (described in Section 2.3.2).

2.3.1 AirStick variables

The *AirStick* produces five key data streams that are used in notation generation in the piece. They are:

- **'Energy' over time** — the amount of force put into the *AirStick* at a point in time. Calculated in the receiver software as *linear acceleration* mean over n samples in a given time window. This variable also used in 'Energy variation', the rolling Standard Deviation over n samples in time window, calculated in Max.

- **Yaw** — where the *AirStick* is pointing across a horizontal plane (i.e. left to right), relative to its position in space. This variable also used in 'Yaw over time', the mean over n Yaw samples in a given time window. Euler angle calculated in receiver software from quarternion values, values over time calculated in Max.

- **Pitch** — where the *AirStick* is pointing across a vertical plane (i.e. up and down), relative to its position in space. Euler angle calculated in receiver software from quarternion values.

- **Roll** — where the *AirStick* is rotating along the axis of the baton (i.e. twisted right or left), relative to its position in space. Euler angle calculated in receiver software from quarternion values.

- **'Poke' gesture** — a poke in the forwards direction, sent as a single trigger. Calculated in the receiver software as a threshold of directional *linear acceleration* in a forwards direction.

2.3.2 Notation variables

In mapping the above variables, different structural levels are applied to map notation to structural levels of the piece:

- **Note-level structure** — decisions on a note-to-note basis (e.g. the pitch and duration of each note)

- **Phrase-level structure** — decisions across a small passage of up to three bars (e.g. the influence of modes of pitches)

- **Piece-level structure** — decisions across the work (e.g. duration of piece, structural movements or sections)

AirStick and notation variables are applied according to the mapping criteria outlined in Table 1, illustrated in the example in Figure 5.

Whilst in theory a replicated gesture will produce the same notation result, in reality, the subtle differences in motor movements and the high sensitivity of the AirStick

Musical variable	**AirStick variable**	**Structural level**	**Example**
Player part	Yaw	Note	Notation generated only on Player 1's part
Pitch values	Pitch and Yaw	Note	An E♭
Rhythmic values	Yaw	Note	A dotted quaver note value
Articulation	Energy variation over 5 sec	Note	A tenuto marking on a note
Phrase length	Energy	Phrase	The end of a phrase in a player's part, marked by an instruction to augment it for 1 minute
Dynamics	Energy over 10 sec window	Phrase	A *pp* dynamic marking on a phrase
Tempo	Energy over 30 sec window	Phrase	A *Largo* tempo marking on a phrase
Pitch influence	Pitch and Yaw over 30 seconds	Phrase	Pitch values limited to natural minor note set
Extended techniques	Roll	Phrase	An *air tone* marking on a flute phrase
Structure	'Poke' gesture	Piece	Change the structural direction of the piece, resetting variables

Table 1. Musical variables and their respective AirStick mappings.

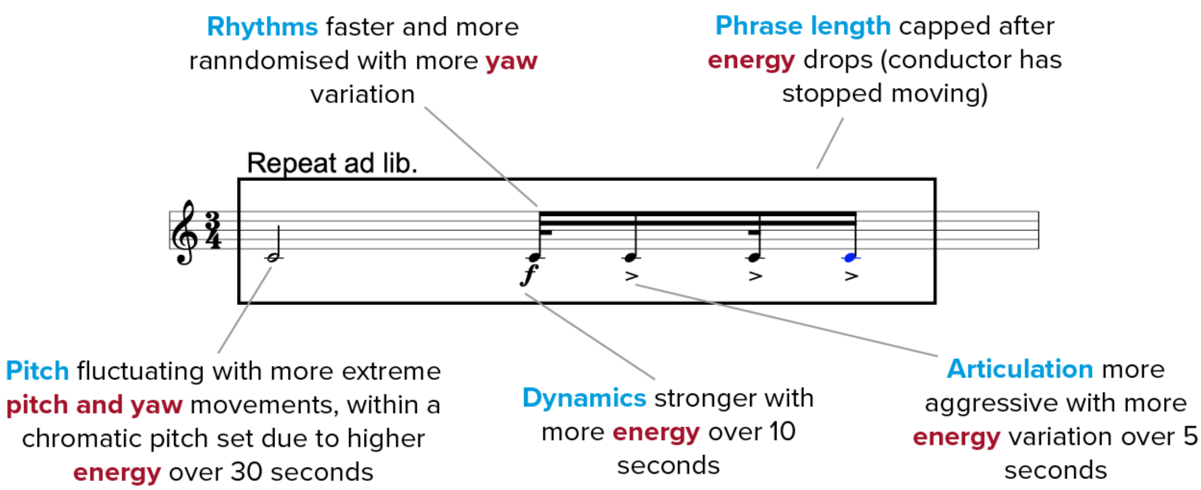


Figure 5. An an example of mapping variables applied to a single part's phrase described in Table 1.

sensor gives rise to varying notation results – an interesting proposition for improvisation. Additionally, the presence of phrase- and piece-level structures in mappings means that 'local' structures and patterns in notation tend to form, such as when the conductor remains at a particular energy, pitch and yaw over an extended period of time, shaping both the pitch mode and dynamics of the notation.

2.4 Performance system

Performers begin the work connected to the same server, each with different parts, as seen in Figure 6.

Figure 6. Multiple devices connected to real-time notation generation using *drawsocket*.

The conductor activates the notation system by picking up the *AirStick*, and bringing performers to attention. The conductor also receives a static paper score — a work composed by the author, created for the performance, that contains music that will never be 'heard' in a traditional sense. Instead, the score provides material for the conductor to follow.

The score has been deliberately designed to accentuate different gestures, and to focus on players in both a discrete (i.e. solo) and general (i.e. unison tutti) manner. In some senses, the conductor's score is a caricature of a traditional score, designed to be interpreted as a normal score, but producing accentuated gestures for the performance system. For instance, an particular section of the score may be written with a *fortississimo* dynamic marking and hocketed rhythms, encouraging strong and discrete gestures from the conductor. The static score may also be changed from performance to performance, as the notation system is independent of the score.

Both the static score and real-time notation use Western Classical notation to reduce the cognitive strain on performers and conductors (as discussed in [5]), allowing for a holistic approach to performance that allows space to improvise and listen.

While the notation received on devices is near real-time (with roughly 6ms latency between the *AirStick* and a note appearing on a device), the presence of different time scales in notation generation described in Section 2.3.2 provides an interesting context for improvising performers. That is, there is a theoretical minimum time from a gestural event to music notation (outlined in Figure 7), but this notation is not occurring at every conceivable sample of the sensor, which would produce an endless stream of notes. Instead, the presence of thresholds means that the performer is met with musical instructions on multiple time scales – from the immediate gesture of the conductor themselves, to a phrase-level dynamic marking based on 10 seconds of gestural data, to a piece-level event that might occur every 30 seconds based on the conductor using the poke gesture. The result is a series of rolling notation snapshots from a performer's perspective, that advance forward but do not edit previously generated notation.

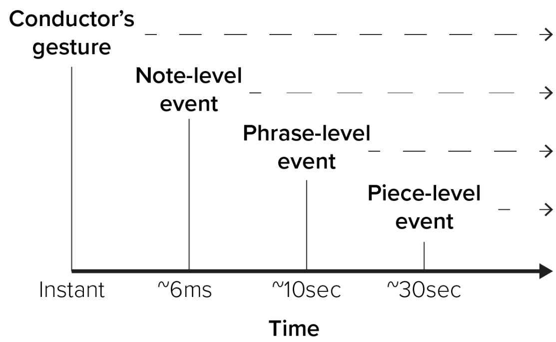


Figure 7. Theoretical minimum times from gesture to performer.

These time scales allow for a notation system that dynamically reacts to gestures, producing both clear immediate notation and longer-term structural instructions, and also ensuring that the conductor retains a certain level of control over the performance. For example, a neutral rest position provided by the conductor can still indicate silence amongst players, maintaining an intricate dialogue between the score, notation system, conductor and performers.

3. DISCUSSION

3.1 Interpretation and evaluation

Musical interpretation, a form of translation, is reshaped in *Mutable Gestures*.

In traditional chamber ensemble performance, the cause and effect relationship between conductor and performer, and performer and sound, is relatively clear, and this causal link provides a context and foundation for listening [15].

From the audience's perspective, *Mutable Gestures* could very well be a traditional notated work, with a conductor fulfilling their normal role and all players playing off a score, as illustrated in Figure 8.

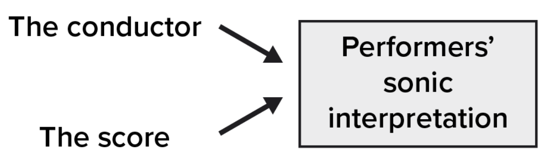


Figure 8. A possible audience perspective of interpretation in *Mutable Gestures*.

Yet beneath this flowchart lies a more complicated reality, based on the mapping of gesture to sound.

The conductor uses a score which is never 'heard', but is communicated to performers, who use a translated form of these gestures (using the notation generated live and their own interpretation) to create sound.

The work becomes a translation of a musical work that is never sonically realised from its score form in a conventional musical relationship, instead interpreted through gesture and a generative notation system, with three steps of interpretation or evaluation of musical information along the chain, as seen in Figure 9.

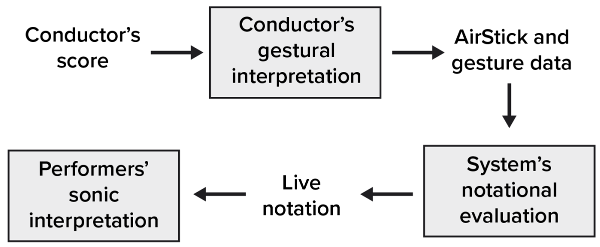


Figure 9. Interpretation and evaluation in *Mutable Gestures*.

Evidently, the link between gesture and sound making is less clear. Beneath the deceptively simple correlation between gesture and sound, the role of conductor is redefined as someone who is not just controlling, but creating musical material through gesture.

The *AirStick* is capable of generating its own sound using the receiver application, and this dimension has been explored in other work with the instrument. However, this project 'mutes' that layer, adding back in another layer of human interpretation in the process of mapping gesture to sound.

Yet, if the sound of the players were to be disembodied from the performance space, like a traditional DMI, *Mutable Gestures* could very easily be interpreted as a form of instrument - what Maestri and Antoniadis describe as "liquidizing the limit between notation and instrument" [16].

Hierarchy of interpretation and evaluation creates a complex network of relationships to technology, reminiscent of works within the DMI community.

3.2 Democratising roles

The breaking down of traditional roles in *Mutable Gestures* alludes to a form of democratisation in that the fixed structures of the relatively autocratic role of the composer are replaced by a system that distributes creative control and inverts the power structures within a traditional chamber music context. This is despite the appearance on the surface that the work is functioning as a traditional chamber performance, as discussed in Section 3.1.

Of course, many chamber works change where creative control lies. Where *Mutable Gestures* differs is in the way it takes advantage of the the natural affordances and illusions provided by roles in a chamber music context, taking fixed structures such as gesturing by a conductor, and reinterpreting and relocating within a generative system.

In a traditional sense:

- conductor becomes performer (playing a DMI, almost entertaining the audience through movement) and composer (generating notation from gesture);
- composer becomes curator (setting the rules of engagement and designing the system to the point that the piece 'works'); and
- performer adds the role of composer (improvising musical material)

One musical system with similar characteristics to *Mutable Gestures* is the concept of Soundpainting, where a 'Soundpainter' controls one or many musicians through a gestural language. Notably, Marc Duby suggests that Soundpainting provides a balance between two extremes in power relations, between an "...orchestral performance, in which the conductor plays a pivotal role, and those of freely improvised music in which, ostensibly at least, there is no leader" [17]. In contrast to Soundpainting, I suggest that *Mutable Gestures* once again reinterprets power relations, this time between improvisation and Soundpainting, retaining the leadership position of a conductor, but distributing the control of information and notation generation amongst multiple blurred roles.

3.3 Future work

Due to COVID restrictions, *Mutable Gestures* performances have been postponed. The piece will naturally evolve through workshops and performances and is created in such a manner that will allows for adaptation.

Aside from performance, the work could also benefit from expanding the number of players and instrumentation possible, perhaps in a larger ensemble or orchestral context. More players would allow for additional textural, and move the *AirStick* closer to a DMI in the sense that it would have greater sounds at its disposal, almost like a software instrument.

Additionally, machine learning in the gesture recognition field is a logical next step for the work, particularly growth of ML and gesture within other DMIs [18, 19].

4. REFERENCES

[1] A. Wyatt and C. Hope, "Conducting animated notation: Is it necessary?" in *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR'20/21*, 2020, pp. 169–174.

[2] Y. K. Lim and W. S. Yeo, "Smartphone-based music conducting," in *Proceedings of the 14th International Conference on New Interfaces for Musical Expression*, 2014.

[3] T. M. Nakra, Y. A. Ivanov, P. Smaragdis, and C. Ault, "The ubs virtual maestro: an interactive conducting system," in *Proceedings of the 9th International Conference on New Interfaces for Musical Expression*, 2009.

[4] I. Jarvis and D. V. Nort, "Posthuman gesture," in *ACM International Conference Proceedings*. Association for Computing Machinery, June 2018.

[5] G. Dori, "Using Gesture Data to Generate Real-Time Graphic Notation: a Case Study," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*. Hamburg University for Music and Theater, October 2020, pp. 68–74. [Online]. Available: https://doi.org/10.5281/zenodo.4105932

[6] R. Hoadley, "Making people move: Dynamic musical notations," in *Proceedings of ICMC-SMC 2014 : 40th International Computer Music Conference*, September 2014.

[7] C. Erdem, K. H. Schia, and A. R. Jensenius, "Vrengt: A shared body-machine instrument for music-dance performance," in *Music Proceedings of the International Conference on New Interfaces for Musical Expression*, F. Visi, Ed. Porto Alegre, Brazil: UFRGS, June 2019, pp. 63–65. [Online]. Available: http://www.nime.org/proceedings/2019/nime2019_music017.pdf

[8] J. Jaimovich, "Emovere: Designing sound interactions for biosignals and dancers," in *Proceedings of the International Conference on New Interfaces for Musical Expression*. Brisbane, Australia: Queensland Conservatorium Griffith University, 2016, pp. 316–320. [Online]. Available: http://www.nime.org/proceedings/2016/nime2016_paper0062.pdf

[9] L. Andersson López, T. Svenns, and A. Holzapfel, "Sensitiv – designing a sonic co-play tool for interactive dance," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Shanghai, China, June 2021. [Online]. Available: https://nime.pubpub.org/pub/y1y5jolp

[10] A. Sarasua, "Context-aware gesture recognition in classical music conducting," in *MM '13: Proceedings of the 21st ACM international conference on Multimedia*. Association for Computing Machinery (ACM), October 2013, pp. 1059–1062.

[11] H. Je, J. Kim, and D. Kim, "Hand gesture recognition to understand musical conducting action," in *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, 2007, pp. 163–168.

[12] S. Bhagwati, "Notational perspective and comprovisation," *Sound & Score. Essays on Sound, Score and Notation*, pp. 165–177, 2013.

[13] A. Ilsar, "The airsticks: a new instrument for live electronic percussion within an ensemble," Ph.D. dissertation, University of Technology Sydney, 2018.

[14] A. Ilsar and M. Hughes, "A new audio-visual gestural instrument for unlocking creativity," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–4.

[15] G. Emerson and H. Egermann, "Gesture-sound causality from the audience's perspective: : investigating the aesthetic experience of performances with digital musical instruments." *Psychology of Aesthetics, Creativity, and the Arts*, vol. 12, p. 96, 2017.

[16] E. Maestri and P. Antoniadis, "Notation as instrument: From representation to enaction," in *International Conference on Technologies for Music Notation and Representation – TENOR'15*, 2015. [Online]. Available: http://vimeo.com/80558397.

[17] M. Duby, "Soundpainting as a system for the collaborative creation of music in performance," Ph.D. dissertation, University of Pretoria, January 2006.

[18] M. E. Benalcázar, A. G. Jaramillo, Jonathan, A. Zea, A. Páez, and V. H. Andaluz, "Hand gesture recognition using machine learning and the myo armband," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 1040–1044.

[19] B. Caramiaux and A. Tanaka, "Machine learning of musical gestures: Principles and review," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. Graduate School of Culture Technology, KAIST, 2013, pp. 513–518.

MUSIC NOTATION USING REACTIVE SYNCHRONOUS PROGRAMMING

Bertrand Petit
INRIA
Sophia-Antipolis, France
Bertrand.petit@inria.fr

ABSTRACT

This article presents a notation system for music based on *patterns* (or *clips*) as they have been popularized for more than twenty years with the Digital Audio Workstations on the market like *Ableton Live*, *Bitwig Studio* or *FL Studio*. This system named *Skini* uses the *HipHop.js* programming language to describe music pieces. This language, belonging to the family of synchronous reactive languages, was initially designed for the orchestration of Web services. Skini, by combining HipHop.js and queuing mechanisms, was developed for interactive and generative music performances. It has also proven to be an efficient tool for notating musical pieces outside of these interactive and generative contexts because of its ability to describe the structure of a piece of music in a form close to its expression in everyday language. Moreover, Skini, while using certain concepts specific to electronic music, can be used for the creation and performance of instrumental and orchestral music.

1. INTRODUCTION

According to a common definition, music notation consists of transcribing a musical work onto a medium to interpret, preserve, protect, and disseminate it. The systems of notation are dependent on the media available to support this notation. For example, Hurrian songs have been found on clay tablets from around 1400 BC. The paper and the staff system from the neumes of the Middle Ages was for a long time the only support in the occidental societies until the information sciences appeared and proposed other ways to notate music. With the help of computers, it is not only a matter of facilitating the manipulation of the staff system, but also of introducing other ways of representing music. Of the many ways to score music, we will focus here on a system that uses a computer language from the family of *synchronous reactive languages* [1]. These languages, imagined in the 80's, are not initially intended for music but for critical systems (airplane, train, nuclear power plant...). However, we will see that one of these languages, HipHop.js [2], is well adapted for representing music in the form of *patterns* or *musical elements* similar to what is called *clips* in current Digital Audio Workstations (DAW). The HipHop.js language is implemented in a platform named Skini which was initially designed for collaborative music in interaction with an audience, but which can also be used to produce generative music or the musical notation of clip-based pieces.

2. RELATED WORK

There are several families of tools allowing the notation of musical pieces using programming languages or computer systems.

In the family of programming tools for sound creation, *Open Music* from IRCAM [3] is an example of a solution allowing to describe musical processes using graphical programming that generates LISP code. *Csound* [4] is another popular tool for producing music from a computer language, this time from the C language.

Another family of tools is constituted by the *Live Coding* solutions. These solutions deal with algorithmic music improvisation, which *Open Music* or *Csound* were not designed for. Popular languages in this category are for example *ChucK* [5], *SuperCollider* [6] or *Fluxus*.

Graphical languages like *MAX/MSP* and *PureData* [7] are not intended for music notation per se but for the combination of musical processes and signal processing tools. However, many pieces are designed from these tools without any other notation mode than the *Patches* written in these languages.

Some Digital Audio Workstation (DAW) like *Ableton Live* [8] or *Bitwig Studio* [9] are also de facto notation tools. A piece of music is expressed as a matrix of clips with a whole set of properties. Each of the clips can be expressed in the form of MIDI *piano-roll*, which is a notation system commonly used by DAWs.

We will not discuss here the case of *score editors* such as *Finale*, *Sibelius*, *MuseScore* or *Dorico* which are tools for entering and formatting scores as they have been written for several centuries. These tools, although computerized, do not fundamentally call for programming skills on the part of the composer.

Each of these families of tools constitutes a solution to particular music production problems. *Open Music* is intended for composers with a process approach to music creation. This tool is aimed at musicians with good algorithmic skills. The Live Coding tools were originally designed for performance and improvisation. Tools such as *MAX/MSP*, *PureData* or *Csound* are intended for general sound production, not specifically for an improvisation context. DAWs like *Ableton Live* or *Bitwig studio* are initially designed for live performance while offering a relatively simple way of structuring music compared *to MAX-MSP* for example, whose learning curve is long.

Skini is somewhere between clip-based DAWs and computer-based programming in the sense of *Open Music* or *Csound*. Skini is not a *Live Coding* tool because it is not designed for improvisation, even though it is a tool initially designed for live and interactive performances. We only discuss here the use of Skini in the well-defined context of music notation and not in the context of generative music described in the article [10] or for collaborative music described in the article [11]. We will see that Skini is mainly interested in a method of conception of complex musical structures. "Structure design" is not the basic problem of Computer Music solutions which will deal with other topics such as signal processing, live production, score editing or improvisation. We will see that, in this sense, Skini is a complementary notation tool to most of those already available.

3. SKINI NOTATION BASIC CONCEPTS

We will review the main concepts behind the music notation used by Skini. These are patterns, instruments, interaction, and scenarios.

3.1 Patterns

Musical patterns, or musical elements, are the raw material of Skini. They are short musical phrases designed by the composer and arranged in such a way as to constitute the musical piece. They are close to clips in the vocabulary of DAWs, except that clips cannot be graphic extracts from scores, which is possible with Skini when the music is played by musicians and not by a DAW.

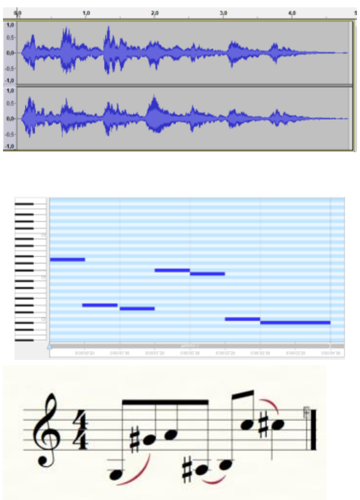


Figure 1: Three ways to describe a pattern.

As we can see in **Figure *1***, a pattern can take the form of a sound file, a MIDI sequence, or a few bars of a score. Skini does not impose any constraints on the design of the patterns, nor on their duration. The forms that the composer will give to the patterns will depend on the style of music and the type of performance imagined. A composer who wants to control synthesizers will naturally use MIDI patterns. A composer writing for instrumentalists will more naturally use score elements.

3.2 Instruments

Patterns are played by instruments. We will see that in his notation system, Skini can request the execution of several patterns by one instrument at a time. As we consider that an instrument can only play one pattern at a time (as an instrumentalist does), if several patterns are requested for the same instrument at the same time, they will be placed one after the other in a *queue* to be played one after the other. The principle of queues for patterns was initially implemented to guarantee a good coherence of the musical pieces in interaction with the audience, and to allow the use of Skini with instrumentalists. In the simpler context of notation of a pattern-based musical piece without calling for interaction, queues allow to simplify the implementation of musical sequences in parallel. For example, the composer can decide to load a complete musical sequence at an instant of the score without having to worry about this sequence while the rest of the score is processing other sequences on other instruments. Schematically, queues allow to implement musical sequences without worrying about their duration.

3.3 Interaction

Skini natively considers the possibility of interacting with music in the form of events produced by sensors, web interfaces proposed to the audience, messages received by a video game or random processes. We will not discuss in detail the complex question of interaction here. But this is an important aspect of music notation with Skini. Indeed, few tools have a way to describe music and interactions with events outside the music. *Antescofo* [12] and *InScore*[13] are examples of these tools. For more information, we refer you to another article on interaction [11].

3.4 Scenario or score

As we have seen, Skini proposes a way to describe a musical piece based on the concepts of pattern, instrument, and interaction. Skini is interested in the way the composer will organize these concepts. It is a tool to define structures, *orchestrations* but in a computer science sense. The term *score* would be well adapted for this description, but it is most often associated with the notation of pitches in time, which is not the purpose of Skini. The term *orchestration* in a musical context would be closer, but it does not include the notion of pattern. We retain the rather vague term of *musical scenario* associating patterns with interaction, and queues by instruments. Skini scenarios are written using the computer language: HipHop.js[1].

4. PRINCIPLE OF SYNCHRONOUS REACTIVE PROGRAMMING WITH HIPHOP.JS

Synchronous reactive programming languages are languages meant for programming *reactive systems*. Computer systems are often classified into three categories. *Transformational systems* that take inputs, process them, provide their outputs, and terminate their execution. *Interactive systems* that continuously interact with their environment, at their own speed. A typical example is a web application. *Reactive systems* that continuously interact with their environment, at a speed imposed by the environment. A typical example is the control system of a vehicle. Reactive systems react to environmental stimuli. HipHop.js belongs to this family of *synchronous reactive languages*, and is very close to the *Esterel* language [14] initially conceived in the 80s. As with *Esterel*, programming with HipHop.js consists in approaching an algorithm by thinking in terms of *events* and *reactions*. Events are materialized by means of *signals*. A HipHop.js program, once compiled, is like a black box to which one submits signals linked to events, and which produces other signals when it is solicited. This solicitation provokes what is called an *immediate* ,and thus *synchronous*, reaction since it does not introduce any delay (at least in a theoretical way). This model is represented by the **Figure** ***2***. The signals A, X, W, Z are purely indicative. There is no limit or constraint on the number of signals and their types.

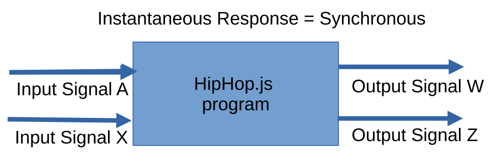


Figure 2: Reactive Synchronous principle.

The syntax of HipHop.js uses a set of about twenty statements whose operation is quite intuitive. For example, it is possible to wait for a signal (`await` statement) to move to the next statement. It is possible to emit a signal (`emit` statement). One of the important characteristics of HipHop.js is to integrate natively the parallelism. To allow two blocks of statements to run at the same time. Few languages natively integrate this important feature, which allows to act on musical sequences by easily modifying their juxtaposition for example. Skini offers the composer two different ways of programming. For composers who are familiar with computer tools and textual programming, it is possible to program in the HipHop.js language which also supports JavaScript. For composers less familiar with textual programming, it is possible to use a graphic programming tool. This tool offers the same primitives as HipHop.js in textual form but is less well adapted to the integration of JavaScript code to realize, for example, complex logical operations within the notation of the piece.

[1] Without reference to a particular musical genre. The authors of this language have chosen this name as a pun towards a platform developed by INRIA called Hop.js.

5. SKINI ARCHITECTURE

Skini provides a notation system within a music production environment. **Figure** ***3*** is a logical view of the Skini platform used to implement HipHop.js scenario programming in its music production environment. The architecture is composed of 3 layers.

The central layer *Control and decision* layer is the one that contains the HipHop.js program and the instrument queues. These queues will receive pattern commands from the HipHop.js program. It can also receive pattern commands from a stochastic engine or an audience. This layer is implemented by means of the multi-tier Web platform *Hop.js* [15], developed by INRIA, or *Node.js* [16]. These platforms use JavaScript. HipHop.js is thus a Domain Specific Language (DSL) supported by JavaScript.

The *event* layer deals with events external to the course of the music piece. It can be a clock or various events produced by sensors, video games, etc. In a simple implementation dealing with notation only, this level can be reduced to the use of a clock.

The *Music Production* layer is necessary to experiment or simulate a musical result and especially when Skini communicates with a DAW that implements a set of patterns. For live performances, instead of using a DAW, Skini can call upon musicians equipped with an interface

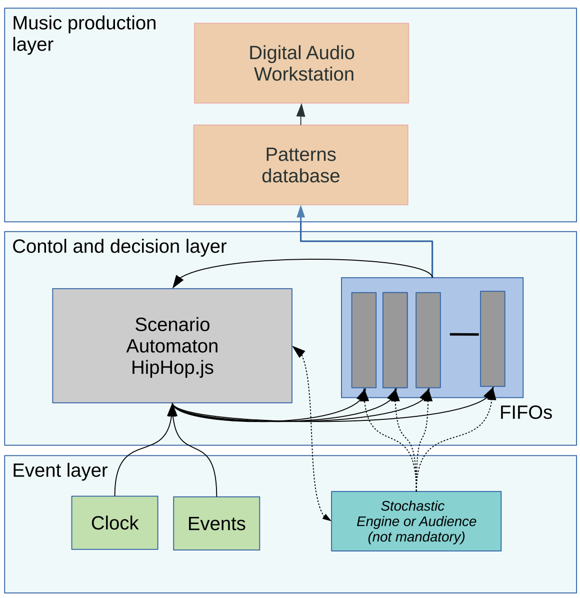


Figure 3: Skini architecture.

that can display patterns in the form of scores (PC or Tablets).

For a composition and simulation work, in the case of a use of MIDI patterns with a DAW, the DAW will be able to record the piece in MIDI format (cf. Figure 4) to verify that Skini's notation corresponds to the expected musical result.

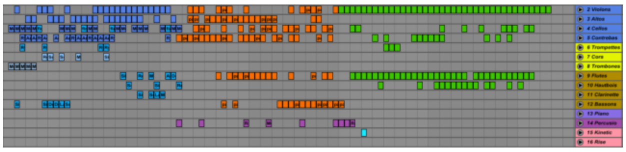


Figure 4: A Skini score view in Ableton Live.

It is then possible to export the results of the composition to an editing software (*Finale* or *Sibelius* for example) to produce a "classical" score (cf. Figure 5). Skini is therefore not limited to the production of electronic music, or live performances but can also be a tool for composing orchestral pieces.

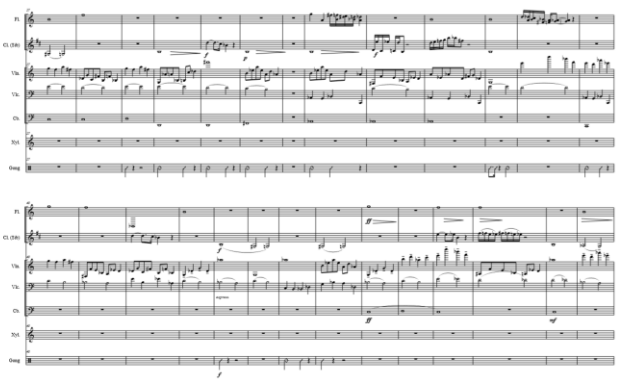



Figure 5: A Skini score view in Finale.

6. EXAMPLES OF SCORE IMPLEMENTATIONS

Without worrying about the details of the implementation of a complete score by means of HipHop.js programming or the graphical programming tool allowing to abstract from a part of the HipHop.js syntax, we will see some simple examples. For more details on HipHop.js programming applied to music, we recommend the article *"Interactive Music and Synchronous Programming*"[17].

6.1 Programming loops

Here is an example of programming a loop on a pattern in HipHop.js.

```
every count (4, tick.now) {
   hop{ putPatternInQueue("Morricone");}
}
```

Using the graphical programming tool, the equivalent will be:

Figure 6: A loop with every.

The statement `every` will execute the body between braces every time 4 ticks. The *tick* signal emitted by a clock has been set according to the duration of the patterns. Here the pattern *Morricone* has been described in a configuration file with a duration of 4 ticks. The `hop` statement is a facility of HipHop.js to pass JavaScript commands. The graphical programming is perfectly equivalent.

Here is another example of loop programming involving two patterns on the same instrument. Each pattern lasts 4 ticks.

```
loop{
   hop{putPatternInQueue("Morricone");}
   hop{putPatternInQueue("Rosenman");}
   await count (8, tick.now);
}
```

Using the graphical programming tool the equivalent will be:

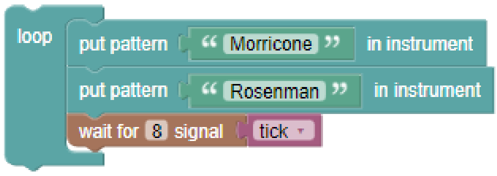


Figure 7: A loop using the loop statement.

Instead of an `every` statement we use here a `loop` statement, which loops indefinitely over the body between braces. The two `putPatternIn-Queue` commands or "put pattern" blocks are executed "at the same time". This means that at each loop, 2 patterns are sent to the instrument's queue to be played one after the other. The `await` statement will stop the loop until 8 ticks have been received. This is a way to avoid overloading the queue immediately.

6.2 A more complex scenario

These two simple examples give an idea of how to program with Skini, but they are not enough to demonstrate the relevance of this programming compared to other solutions such as those offered by a clip-based DAW. Let's look at a slightly more complex case.

We want to loop two patterns *Mancini* and *Silvestri*, let's name this loop *loop1*. At each occurrence of two *loops1* we want to execute another sequence of patterns which consists of playing the pattern *Rota* twice on one instrument and a pattern *Geoffroy* on another instrument. At the same time as these two loops are running, we want to play another sequence a trumpet solo which consists of a rather long sequence of trumpet patterns. We could express this scenario graphically in a sequencer. Space does not allow us to do so in this article, but it is easy enough to imagine that this is a simple but tedious job. In a clip-based DAW, one way to implement this scenario would be to create a control track that sends MIDI commands to tracks containing the patterns (clips) of the instruments. Using a MIDI virtual cable, the commands from the control track could be sent back to the DAW to trigger the patterns. We could create the loops using a *follow action* of each clip and create a MIDI command to stop the track that would be driven by the control track. To implement the scenario, we just need to create control clips in the control track that will issue the start and stop commands for the loops. Technically, this method works. However, it is not very easy to read, and it is difficult to scale up as soon as the scenario becomes more complex.

Here is how this scenario is expressed with Skini:

```
fork{
   every count (8, tick.now){
      hop{putPatternInQueue("Mancini");}
      hop{putPatternInQueue("Silvestri");}
      emit boucle1();
   }
}par{
   every count (2, boucle1.now){
      hop{putPatternInQueue("Rota");}
      hop{putPatternInQueue("Rota");}
      hop{putPatternInQueue("Geoffroy");}
   }
}par{
   run ${soloTrompette}(...);
}
```

Or graphically:

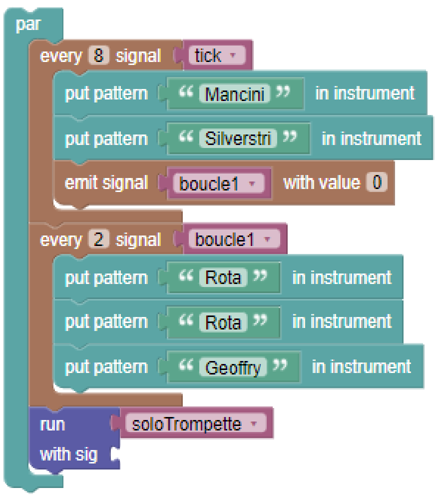


Figure 8: A more complex scenario.

In HipHop.js programming, the `fork` and `par` statements are used to parallel blocks of statements between braces. In graphical programming the `par` block parallels the `every` and `run` blocks. The `run` block calls a module that contains the trumpet solo. The `soloTrompette` module is written in HipHop.js as follows:

```
var soloTrompette = hiphop module () {
   fork{
      hop{ableton.putPatternInQueue("Altenburg");}
      hop{ableton.putPatternInQueue("André");}
      hop{ableton.putPatternInQueue("Hardenberger");}
      hop{ableton.putPatternInQueue("Thibaud");}
      hop{ableton.putPatternInQueue("Gambati");}
      hop{ableton.putPatternInQueue("Foveau");}
      hop{ableton.putPatternInQueue("Friedrich");}
      hop{ableton.putPatternInQueue("Sauter");}
      hop{ableton.putPatternInQueue("Arban");}
   }par{
      await count(44, tick.now);
      hop{ console.log("END OF TROMPET SOLO"); }
   }
}
```

with as graphic equivalent in the module Figure 9. In this module, there are two branches in parallel. One loads the

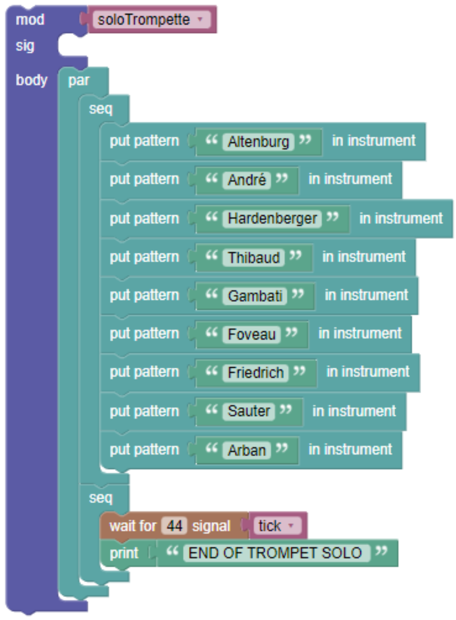


Figure 9: The trumpet solo.

queue of the trumpet instrument with a sequence of patterns. The other branch counts down the total duration of all the queued patterns. Here it gives 44 ticks. After 44 ticks a message is displayed on the console using a JavaScript statement to inform us that the solo is finished.

The first `every` of the main scenario of Figure 8, in which `run` is called, has the same structure as the previous scenario example ***Figure 6***. The difference is that a `boucle1` signal is sent every 8 ticks. The second `every` uses the `boucle1` signal from the first `every` to count down. The trumpet solo is executed independently of the two `every`. Although simple, this dependency between the `every` statements is difficult or even impossible to implement with a clip-based DAW. This would require that a clip of an instrument can issue a control command, which is not standard with current DAWs. But beyond feasibility, the major difference between Skini and a DAW is readability and maintainability. With Skini it is very easy to add more loops. It is also easy to define complex logical combinations between signals on the conditions of the `every` or other signal processing constructs such as `await`, `abort`, `if`, `suspend`, and to structure the piece of music in modules that can be combined and recombined at will.

Regarding the comparison between HipHop.js and general languages (Java, JavaScript, C++...), see chapter 3 of the article [17]. The conclusion of this comparison is close to the one made with a DAW. Writing a script with a general-purpose language quickly becomes very difficult to implement and maintain when the project gets complicated.

6.3 Other writing features

Beyond the description of complex scenarios, Skini includes primitives allowing the introduction of random phenomena in the writing of a piece of music, and to use different OSC or MIDI controls for electronic music. The use of HipHop.js in combination with JavaScript makes it easy to extend the Skini primitives to complex control processes. This is not useful for most composers, but it does mean that it is simple to extend the vocabulary of the notation system.

7. CONCLUSION AND FUTURE WORK

Skini was initially designed for the creation of musical performances in interaction with an audience. The goal of our research was to provide a set of tools for composing, executing, and verifying the aesthetic coherence of interactive music pieces. Examples of music designed using Skini are available at: https://soundcloud.com/user-651713160.

The definition of scenarios that can handle complex combinations of events led us to use a language designed for automata programming. We chose HipHop.js because this language based on the Synchronous Reactive Language *Esterel* [18] seemed to us well adapted to our problem which deals on the one hand with complex automata but also with interaction. Indeed, HipHop.js works on development platforms for the Web using JavaScript. Nowadays, Web technologies are the most common for large-scale interactions. Beyond this dimension of interaction with an Audience, Skini with its ability to manage events, has proved to be a solution to produce music from random processes controlled by events from sensors or video games for example. This association between random process and scenario brought Skini into the world of generative music solutions and more particularly combinatorial generative music. After these first use cases, the use of Skini for score notation without random phenomena or interaction has proved to be interesting. It is indeed complementary to clip-based approaches developed for more than twenty years by manufacturers such as *Ableton*, *Bitwig* or *FL studio* for example. Skini is not a toolbox competing with *Open Music* or *MAX/MSP* because its vocation is to express the structure of a musical work rather than its detail. For Skini, the detail comes from the patterns for which there is no recommendation or constraint imposed by our system. The composer can create each pattern manually, or use a tool like *Open Music*, or use an Artificial Intelligence solution for example.

We think that the creation of music based on patterns in the form of clips has the merit of being widely diffused and that it offers a strong potential for development in combination with tools allowing the management of complex structures as proposed by Skini. DAW editors such as *Ableton* or *Bitwig* are integrating more and more functionalities that go in the direction of scenario management in the sense that we have detailed in this document. However, they have not yet addressed the problem in the form of a notation system as powerful as Skini. This is not a coincidence because the problems raised by the notation of complex scenarios are difficult to solve. *Synchronous reactive languages* are the only ones to have provided a viable solution through substantial research work, but although they have been very successful in the industry, they have not yet been widely used in the music world.

Skini can therefore address a population of DAW users wishing to create musical pieces with complex structures that cannot be done with the generalist tools.

8. REFERENCES

[1] P. A. Laplante and S. J. Ovaska, “Programming Languages for Real-Time Systems,” in Real-Time Systems Design and Analysis, 2011.

[2] G. Berry and M. Serrano, “HipHop.js: (A)Synchronous reactive web programming,” in Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2020, pp. 533–545.

[3] J. Bresson, “Reactive Visual Programs in OpenMusic,” [Research Report] IRCAM / ANR-13-JS02-0004-01 Effic., pp. 0–12, 2015.

[4] V. Lazzarini, S. Yi, J. Ffitch, J. Heintz, Ø. Brandtsegg, and I. McCurdy, Csound: A sound and music computing system. 2016.

[5] G. Wang, “The ChucK Audio Programming Language “ A Strongly-timed and On-the-fly Environ / mentality,” 2008.

[6] J. McCartney, “Rethinking the computer music language: SuperCollider,” CMJ, 2002.

[7] M. Puckette, “Max at seventeen,” CMJ, 2002.

[8] Ableton, “Ableton Live 10 - Ableton,” Ableton, 2018. [Online]. Available: https://www.ableton.com/fr/live/. [Accessed: 10-May-2020].

[9] S. Truss, “Bitwig Studio 3,” Electron. Music., vol. 35, no. 11, 2019.

[10] B. Petit, “Generative Music Using Reactive Programming,” Proc. ICMC 2021, pp. 320–324, 2021.

[11] B. Petit and M. Serrano, “Composing and Performing Interactive Music using the HipHop . js language,” in New Interfaces for Musical Expression 2019, 2019, pp. 71–76.

[12] A. Cont, “Antescofo: Anticipatory synchronization and control of interactive parameters in computer music,” in International Computer Music Conference, ICMC 2008, 2008.

[13] D. Fober, S. Letz, Y. Orlarey, and L. France, “Programming Interactive Music Scores with INScore,” SMC, vol. October 20, pp. 2–7, 2013.

[14] G. Berry, “The Foundations of Esterel,” in Proof, Language, and Interaction, MIT Press., 2000.

[15] M. Serrano and V. Prunet, “A glimpse of hopjs,” in ICFP 2016 - Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, 2016.

[16] R. Prediger, R. Winzinger, R. Prediger, and R. Winzinger, “Node.js,” in Node.js, Carl Hanser Verlag GmbH & Co, 2015, pp. I–XV.

[17] B. Petit and M. Serrano, “Skini: Reactive programming for interactive structured music,” Art, Science, and Engineering of Programming, vol. 5, no. 1, 2020.

[18] G. Berry, The {Esterel} v5 Language Primer. Sophia-Antipolis, France, 2000.

THE TWITTERING MACHINE

David Kim-Boyle
The University of Sydney
Sydney, Australia
david.kim-boyle@sydney.edu.au

ABSTRACT

This paper describes the development of *The Twittering Machine* (2021) for HoloLens 2 and prepared piano, which features a three-dimensional (3D) performance score holographically projected on the surface of the piano keyboard. The score presents a real-time visualization of Twitter tweets scraped during the performance and generated through the application of various Natural Language Processing (NLP) techniques. Various technical aspects of the work are discussed including the NLP processes, network architecture facilitating communication with the HoloLens 2, and techniques through which the holographic score is accurately mapped to the surface of the piano keyboard. The paper describes the work's aesthetic focus and details how mapping process from language to musical notation provides structural form.

1. INTRODUCTION

It is estimated that around 500 million tweets are posted to Twitter every day. From this extraordinary abundance of words much of which is fleetingly expressed and ephemeral in standing, Twitter has become not only a highly visible influence in political discourse [1] and protest [2], but also an invaluable source of data to help inform theorisation on a range of concepts such as trust [3], identity [4], and cultural appropriation [5]. Through its public Application Programming Interface (API), data scientists have even been able to explore how Twitter can provide important insights on human movement and mobility [6, 7] with instrumental application in fields ranging from economics to epidemiology.

For the author, Twitter presents first and foremost, an extraordinary map of differential relations which can be aestheticized in unique forms of musical expression. *The Twittering Machine*, for prepared piano and HoloLens 2 henceforth *TM*, explores this concept through a small microcosm of the Twittersphere, charting a sonic map of difference between tweets ostensibly similar in topic but often wildly divergent in expressed sentiment. The work features a 3D holographic score projected above the surface of the piano keyboard with features that constantly transform as incoming tweets are analyzed during the performance with various NLP techniques. Measure of difference or flux across tweets are determined and used to spatially transform (compress, rotate, stretch) this cartographic representation.

2. LANGUAGE

From the application of generative grammars [8] to facilitate understanding of the listening experience in the work of Lerdahl and Jackendoff [9] through to the exploration of Jakobson's theory of aphasics [10] in the work of composer Aaron Cassidy [11], linguistic insights and frameworks have been a source of inspiration for music theory, cognition, and creative practice for decades. More recently, with the development of powerful NLP libraries such as *spaCy*,[1] and *NLTK*,[2] which run in easy-to-use programming environments such as Python, unlike the intimidating development environments of early computational linguistic models [12], composers have unprecedented access to a wide range of powerful tools that can analyze a text's formal and semantic properties and provide data that may be applied to musical organization.

Exploring musically and aesthetically satisfying ways to create musical structure from NLP data was an important and particularly challenging stage in the development of *TM*. The poetic inspiration for the work, however, was not motivated by how these relationships might manifest. Rather, as previously noted, the work sought to focus on how qualities of difference and repetition propagate through the tweet-retweet paradigm that underpins much of the Twittersphere [13].

3. SCRAPING AND NLP

3.1 Twitter Scraping

The Twitter public API allows a plethora of data to be scraped from Twitter ranging from the textual content of a tweet and the number of likes or retweets that a tweet might have, through to tweets that have unique keywords or hashtags (#) embedded within their content. While the data types accessible through Twitter's API have ostensibly been guided by the needs of market and business analytics, privacy concerns have recently factored into these

[1] https://spacy.io

[2] https://nltk.org

determinations particularly with the 2021 deprecation of the ability to obtain the precise GPS location of scraped tweets, much to the chagrin of data analysts and others who may have used this information to provide helpful insights on social demographics [14].

In *TM*, tweets are scraped on a discrete keyword, e.g. "delta", with the Python library *Tweepy*,[3] every twenty seconds through the following generic API call –

```
for tweet in tweepy.Cursor(api.search, q=data, count=6,
lang="en", tweet_mode="extended").items();
```

where *data* corresponds to the stated keyword "delta". Additional conditionals are attached to the call to ensure only English language tweets are returned (using *lang="en"*) and that the entire text rather than a truncated text is returned (by using the command *tweet_mode="extended"*).

The subsequently returned text from each *Tweepy* call returns a string which is processed with the NLP Python libraries *spaCy* and *TextBlob*.[4]

3.2 Natural Language Processing

spaCy is a powerful Python library that returns a wealth of information on a text's formal structure and to a lesser depth, its semantic content. More advanced applications include the use of transformers for producing textual summaries, translations or developing chat bots. In *TM*, *spaCy* is applied at a relatively high-level to analyze a text's formal properties.

The *spaCy* pipeline first tokenizes a tweet by tagging each word with a part-of-speech (POS) tag which classifies its structural function within a sentence, as demonstrated in the simple example shown in Figure 1.

```
Autonomous cars shift insurance liability
towards manufacturers.

ADJ NOUN VERB NOUN NOUN ADP NOUN PUNCT
```

Figure 1. Sample POS tagging of a sentence. Text from *spaCy* documentation [15].

POS tagging is an important early stage of NLP more broadly as it facilitates dependency parsing, i.e. finding the relationships between the constituent words and phrases within a sentence. In *spaCy*, the first stage of dependency parsing, is to identify noun phrases, or *noun chunks* in the *spaCy* vernacular, within a sentence. Noun phrases are those chunks of a sentence which have a noun at the head. Curiously, in *spaCy* verb phrase identification is not part of the dependency parsing pipeline. Noun chunking of the sample sentence from Figure 1 returns the following –

```
chunk ['Autonomous', 'cars']
chunk ['insurance', 'liability']
chunk ['manufacturers']
```

Figure 2. Noun chunks of the sentence "Autonomous cars shift insurance liability towards manufacturers." Note that verbs ('shift'), adverbs, and adpositions ('towards') are not contained within the chunks.

In *TM* no further use of *spaCy* occurs beyond noun chunking however, to conclude the preceding example, once noun chunking has been completed, *spaCy* can produce a parse tree which presents the formal relationships between each constituent word of the sentence, see Figure 3.

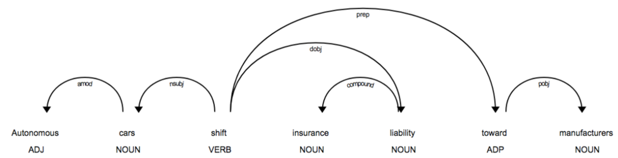


Figure 3. Graphic representation of the parse tree of "Autonomous cars shift insurance liability towards manufacturers." Example taken from *spaCy* documentation [15].

In *TM*, the very first tweet scraped upon commencement of a performance establishes a baseline against which all future tweets are measured for *similarity*. This measurement is undertaken with the *TextBlob* NLP Python library rather than *spaCy* as the latter can only provide similarity measurements between individual words rather than complete sentences and hence was deemed unsuitable. While *TextBlob* can also be used as a pipeline component within *spaCy*, in *TM*, it is used independently to measure the *similarity* between tweets but also the *sentiment* of an individual tweet.

In Twitter analytics, *sentiment* is expressed as a floating-point value ranging from -1.0 (negative sentiment) to +1.0 (positive sentiment) with tweets with *sentiment* values close to zero considered to be neutral. For example, a phrase such as "I love learning about the wonderful world of natural language processing" will return a more positive value for *sentiment* than "I hate studying natural language processing. It is very difficult." Similarity is a slightly more nuanced concept. While the two phrases just presented return polarized values for *sentiment*, they have similar subject matter. This is reflected in *TextBlobs* with a measurement known as a *similarity index*, a floating-point value ranging between 0.0 (highly dissimilar) and 1.0 (identical). The *similarity index* of the two preceding sentences is 0.83. In contrast, the *similarity index* of the first sentence of the pair with "All cows eat grass" is 0.12.

As an example of how NLP is applied in *TM*, which will be followed through in the next section, consider Figure 4 which presents two tweets returned on a scrape of keyword "delta."

[3] https://www.tweepy.org

[4] https://www.textblob.readthedocs.io/en/dev/

Figure 4. Two tweets containing keyword "delta" scraped on November 2nd, 2021.

A noun chunk analysis with *spaCy* of the first tweet of Figure 4 returns the following –

```
chunk ['Vaccine', 'impact', 'data']
chunk ['Vic']
chunk ['lower', 'hospitalization', 'burden']
chunk ['NSW', 'Delta', 'wave']
chunk ['higher', '💉', 'coverage']
```

Figure 5. The first five noun chunks from the first tweet of Figure 4.

A sentiment analysis with *TextBlob* returns the value 0.0625 suggesting a generally neutral tone with perhaps the phrase "Great to see…" skewing the result positively.

The returned *similarity index* between the two tweets of Figure 4 is 0.697, implying a degree of similarity that is ostensibly more apparent than real. While the two tweets are similar by the mere sharing of the keyword "delta" the semantic and contextual understanding is markedly different. Correcting for these nuances may be achieved by refining the search terms to include additional keywords or other flags that might yield more topically relevant results. The use of an additional keyword such as "vaccine" on the above search, for example, would not have returned the second tweet of Figure 4 and upon comparison with *TextBlob* would most likely returned a higher similarity measure.

4. VISUALIZATION AND MAPPING STRATEGIES

In *TM*, Twitter scraping and NLP is performed live in the Visual Studio IDE, and all associated data sent via a User Datagram Protocol (UDP) socket to a prototype Max patch. This patch was specifically developed to facilitate compositional planning and design from NLP data and tweet texts. While the processing of NLP data in Max could have been performed directly within Python, the ease with which intuitive graphical user interfaces (GUIs) can be built in the Max environment, together with its built-in tools for matrix analysis and transformation, was a particularly attractive feature during *TM*'s development.

In *TM* a tweet is visualized in the form of a set of colored nodes holographically projected above the surface of the piano keyboard via the HoloLens 2. A sample visualization of one such tweet is shown in Figure 6.

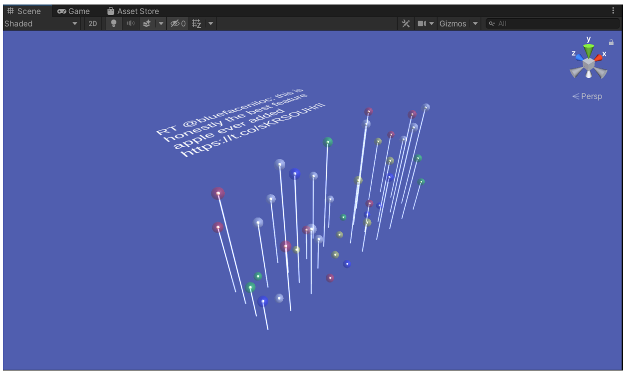


Figure 6. Visualization of a single tweet in *The Twittering Machine* from Unity3D's scene view. Note that the text of the tweet is included in the visualization. While this is mostly for visual aesthetics, it also helps demarcate node distribution groupings.

As can be seen from the above figure, there are only a small number of discrete graphic properties that constitute the visualization – node color, node size, node height, and node position with respect to pitch. From previous experience developing graphic scores, the constraints of our visual perception [16], and feedback from performers, the need to constrain the number of visual elements was an important factor for performability. In development of *TM*, then, there were two fundamental compositional questions to be asked, each of which was informed by the other, namely: 1) how might a tweet be visualized in the form of a performance score? and 2) how might that visualization be musically interpreted?

With respect to the first question, any visualization first requires an a priori decision about the type of data to visualize. In most cases this ultimately reduces to questions of utility or aesthetics. For *TM*, noun chunks, POS tags, tweet sentiment and similarity were found to yield the most useful and consistently usable material to help provide musical structure. At the same time, these linguistic features were most aligned with the aesthetic focus on difference and repetition.[5]

In *TM*, each word contained within a tweet text is represented by a node in the performance score. Node colors, with the exception of those nodes represented by the very first "baseline" tweet analyzed upon commencement of a

[5] In initial conceptualization, the use of precise geolocations embedded within tweets was of interest as this created the opportunity to draw correspondences between the geospatial propagation of tweets and retweets and proportional notational systems however, the deprecation of precise geotags from the Twitter API, meant that this data could no longer be gathered.

performance, are defined by noun chunks and POS tags. Each word within a noun chunk is assigned a uniform node color with colors in succeeding noun chunks cycling through colors red, green, blue, yellow, and orange, see Figure 7.

```
chunk ['Vaccine', 'impact', 'data'] -> [R, R, R]
chunk ['Vic'] -> [G]
chunk ['lower', 'hospitalization', 'burden'] -> [B, B, B]
chunk ['NSW', 'Delta', 'wave'] -> [Y, Y, Y]
chunk ['higher', '💉', 'coverage'] -> [O, O]
```

Figure 7. Mapping of words within noun chunks to node colors where 'R' = red, 'G' = green, 'B' = blue, 'Y' = yellow, and 'O' = orange. Note that emoticons, such as that represented in the fifth line, are not assigned a node in the performance score.

Any word not contained within a noun chunk (verbs, adverbs, adpositions etc.) which will have been identified by the POS tagging process, is represented in the performance score with a white-colored node. These nodes serve a different musical function in *TM* and thus need to be represented in a distinct way.

Musically, *TM* is built from harmonic structures (chords) and linear phrases. Each noun chunk is mapped to one of eight possible harmonic structures with each word within a chunk, or node in the performance score, assigned to a particular pitch within that structure, see Figure 8. Words that fall outside the noun chunks are mapped to a different set of pitches. The mapping and selection process is skewed by tweet *sentiment* and is managed within the prototype Max software.

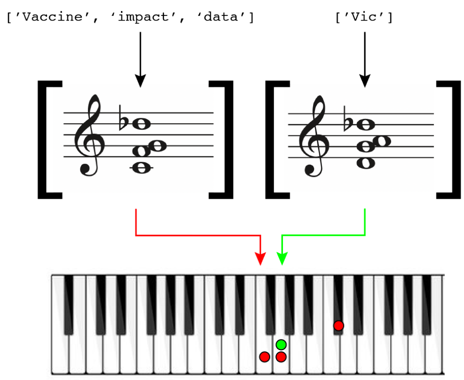


Figure 8. Sample mapping of noun chunk to harmonic structure to node positioning. The selection of active pitches from the harmonic structure is determined within the prototype Max software.

The harmonic structures of *TM* are bound within the interval of a major tenth. This constrains the range of pitches to ensure that tweets can be visualized to contiguous regions of the piano keyboard as shown in Figure 9. To facilitate visual recognition by the pianist, the tweet text from which NLP data is obtained is projected, with a left-justified margin, at the lowest note of the range, see Figure 6. Note that the first tweet visualized upon commencement of a performance, the initial baseline tweet against which all future tweets are compared, is anchored to the lowest region of the piano keyboard. As new NLP data is received via UDP, Max cycles through visualization mappings to adjacent keyboard regions.

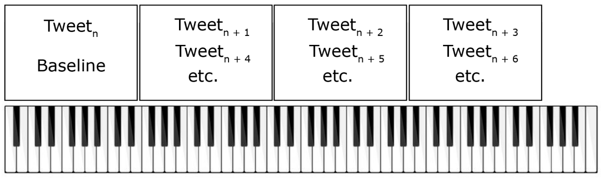


Figure 9. Pitch region mappings in *The Twittering Machine*.

The size of nodes in the performance score and the height above the keyboard at which they are visualized through the HoloLens 2, is also managed within the Max software and is not affected by incoming NLP data. Node height is interpreted by the pianist as an indication of duration while node size denotes dynamic level. Each of the nodes referenced by a word in a noun chunk is ascribed a uniform height and size. Nodes may be placed at a height of 50mm, 100mm, or 150mm above the piano keyboard surface corresponding to temporal durations of 5 seconds, 10 seconds, and 15 seconds respectively. Thin white lines are used to connect the key to the centre of nodes which helps facilitate node-pitch identification particularly when node density increases. White nodes are always positioned just above the surface of the keyboard at a height of 20mm. In a similar way to height assignments, nodes may be in one of three sizes, small (5mm), medium (10mm), large (20mm) corresponding to dynamic levels *pp*, *mp*, and *mf* respectively.

The first tweet analyzed upon commencement of a performance, the initial baseline tweet against which all future tweets are measured for similarity, is mapped and remains anchored to the lowest region of the piano keyboard. Unlike color mappings of all subsequent tweets, each word within this baseline tweet is mapped to a white colored node, again chosen from a predetermined set of pitches. As new tweet data is scraped and analyzed, the spatial distribution of the nodes contained within this baseline tweet visualization is transformed as measures of its similarity with new tweets varies. Spatial transformation of the visualization is performed with simple Max matrix rotation objects which displace node distributions within the octave. The two types of transformation are shown in Figure 10. The correlation of a spatial transform to a melodic permutation has, of course, numerous precedents in compositional practice most notably perhaps in the work of Xenakis [17] and Kagel [18] not to mention the ultimately spatial transformations of twelve-tone rows.

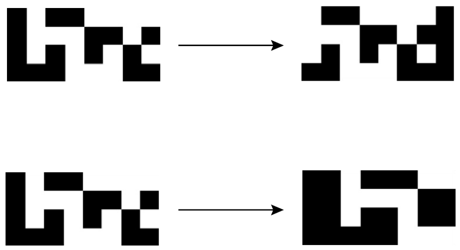


Figure 10. Matrix transformations – a) left displacement by one cell (upper), and b) stretching.

As new tweet information is visualized and transformations of the initial "baseline" tweet are processed, the interpretive options presented by the performance score vary. During performance, the pianist cycles through performance of each noun chunk harmonic unit in an order of their choosing. Linear phrases may be constructed from any visible white nodes or they may alternatively be performed as grace note filigrees to the harmonic structures referenced by noun chunks. The mapping of Figure 11a, for example, may be interpreted as shown in Figures 11b. For simplicity, dynamics and durations are not precisely prescribed.

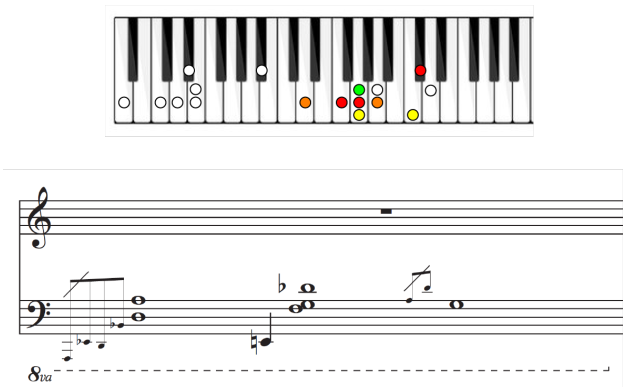


Figure 11. Possible interpretation of the performance score of *The Twittering Machine* – a) the performance score omitting node size and height information, and b) a possible interpretation.

5. TECHNICAL OVERVIEW

Although *TM* runs as a standalone application developed in Unity3D, it is dependent on both Twitter data, and the prototyping mapping software. Developed in C#, it comprises code that parses data received via UDP which is then mapped to the parameters of various 3D game objects (spheres, lines, and text fields). The developed application is then compiled deployed to the HoloLens 2 via the Visual Studio IDE.

5.1 Holographic Anchoring

The most significant technical challenge faced during development of the *TM* application, was the accurate spatial placement of the holograms. The work was initially developed on the original HoloLens hardware, a now deprecated device superseded by the HoloLens 2. Consequently, the readiest solution to the challenge of hologram placement was with fiducial markers, see Figure 12, placed on the inside lid of the piano. While there are several commercially available SDKs specifically designed to facilitate image detection and hologram placement, notably those developed by Vuforia,[6] and Wikitude,[7] neither proved to be effective solutions for *TM*. Vuforia exhibited significant latency between detection of an image and placement of a hologram and the reliability of image detection was inconsistent or at least not consistent enough for the purposes of live musical performance. The Wikitude SDK was even less suitable because it was unable to access the passthrough, built-in camera of the HoloLens.

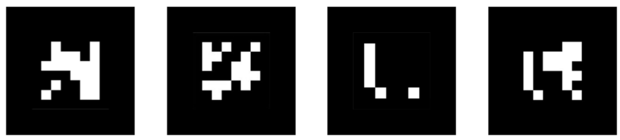


Figure 12. Sample markers used in hologram placements. These markers were placed at discrete locations inside the piano lid with hologram placement correlated accordingly, for example, displaced forward along the *z*-axis towards the pianist and slightly down along the *y*-axis towards the piano keyboard.

The most robust, reliable, and efficient image detection with the original HoloLens hardware was achieved with the ARToolkit through a modification developed by Long Qian [19] based on the use of ArUco markers [20].

On the HoloLens 2, many of the challenges experienced in accurate hologram anchoring had been resolved thanks to several new device affordances including built in QR-code detection,[8] and the extraordinarily powerful new object anchoring features integrated within Microsoft Azure. While the original HoloLens had the ability to use spatial anchors to facilitate the sharing of holograms across multiple users in the HoloLens 2, these were somewhat cumbersome to use and did not maintain persistence. In contrast, the HoloLens 2 features object anchoring, which unlike spatial anchors, can be aligned to objects such as a piano keyboard which will persist across instantiations. This capability mitigated the need to use fiducial markers physically attached to the piano, instead directly aligning holograms with particular piano keys which proved sufficiently reliable for application in *TM*.

[6] https://developer.vuforia.com
[7] https://www.wikitude.com

[8] https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview

5.2 UDP Control

In *TM*, the data received by the HoloLens 2 is sent from Max via the UDP protocol [21]. This requires the data to be processed as symbols and sent via standard *udpsend* objects to the internet protocol (IP) address of the HoloLens 2. For ease of development and testing, the four tweets visualized on the piano keyboard are assigned unique port numbers. As the data received by the HoloLens 2 via UDP consists of only score control data, issues of latency were not of concern.

Figure 13 presents a schematic of *The Twittering Machine*'s entire communication protocol.

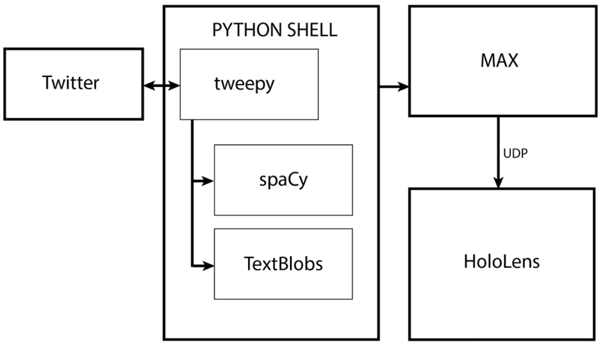


Figure 13. Technical schematic of *The Twittering Machine*.

6. FUTURE WORK

The development of *The Twittering Machine* involved the resolution of more complex technical challenges than any of my previous work with either the HoloLens or generative scores more broadly. While much of this was brought on by technical constraints and limitations, much was also related to challenges of a more aesthetic nature, namely on how Twitter data might be correlated to engaging musical structure and expression.

Further study into how the user experience (UX) of the pianist might impact score design considerations is also an area requiring considerable investigation. Factors such as head mounted display (HMD) comfort, color fidelity, and image stability [22], for example, are just some of the areas in which UX concerns play an important role in determining the effectiveness of a data visualization.

Extended reality (XR) hardware and applications are evolving rapidly with consumer awareness and interest in the metaverse ensuring a significant degree of concomitant commercial development. It is hoped that with the release of more consumer-focused hardware,[9] more researchers and creative practitioners will be able to leverage the affordances of the technology for innovative creative expression. The author is particularly interested in exploring how networking affordances might enable new forms of collaborative experience for users in a shared 3D virtual space and how these experiences might be facilitated through new representational paradigms.

[9] Consumer technology journals predict new mixed reality hardware from Apple, Facebook, and Magic Leap within the next 12-18 months.

7. REFERENCES

[1] J. H. Parmelee and S. L. Bichard, *Politics and the Twitter Revolution*. Lexington Books, 2011.

[2] A. R. Hosterman, N. R. Johnson, R. Stouffer, S. Herring, "Twitter, Social Support Messages, and the #MeToo Movement," in *The Journal of Social Media in Society*, vol. 7, no. 2, 2018, pp. 69-91. Available: https://www.thejsms.org/index.php/JSMS/article/view/475/198

[3] C. DeCarlo, *Twitter Culture: On the Trustworthiness of Tweets*. Enslow Publishing, 2018.

[4] A. Brock, "From the Blackhand Side: Twitter as a Cultural Conversation," in *Journal of Broadcasting & Electronic Media*, vol. 56, no. 4, 2012, pp. 529-549. Available: https://doi.org/10.1080/08838151.2012.732147

[5] D. Rosen (Ed.), *The Social Media Debate: Unpacking the Social, Psychological, and Cultural Effects of Social Media*, Routledge, 2022.

[6] R. Jurdak, K. Zhao, J. Liu, M. AbouJaude, M. Cameron, D. Newth, "Understanding Human Mobility from Twitter," *PLOS ONE*, vol. 10, no. 7, July, 2015. Available: https://doi.org/10.1371/journal.pone.0131469

[7] S. F. Khan, *Understanding Human Mobility Using Twitter*. PhD Thesis, The University of Queensland, 2020.

[8] N. Chomsky, *Syntactic Structures*. Mouton, 1957.

[9] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. MIT Press, 1983.

[10] R. Jakobson, *On Language*. 2nd Ed. Harvard University Press, 1995.

[11] A. Cassidy, *I, purples, spat blood, laugh of beautiful lips*. Self-published, 2006.

[12] L. Karttunen and K. R. Beesley, "Twenty-Five Years of Finite-State Morphology" in A. Arppe et al. (Eds.), *Inquiries into Words, Constraints and Contexts*, CSLI Publications, 2005.

[13] G. Deleuze, *Difference and Repetition*. Paul Patton (Trans.), Columbia University Press, 1995.

[14] A. Kruspe, M. Häberle, E. J. Hoffmann, S. Rode-Hasinger, K. Abdulahhad, and X. X. Zhu, "Changes in Twitter Geolocations: Insights and Suggestions for Future Usage," in *Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text*, Association for Computational Linguistics, 2021, pp. 212-221.

[15] https://spacy.io

[16] W. Thompson, *Visual Perception from a Computer Graphics Perspective*. CRC Press, 2020.

[17] I. Xenakis, *Formalized Music*. Revised Ed. Pendragon Press, 1992.

[18] M. Kagel, “Translation/Rotation,” in *die Reihe*, vol. 7, 1965, pp. 32-60.

[19] L. Qian, A. Deguet, and P. Kazanzides, “ARssist: Augmented Reality on a Head-Mounted Display for the First Assistant in Robotic Surgery,” in *Healthcare Technology Letters*, vol. 5, no. 5, 2018, pp. 194-200.

[20] S. Garrido-Jurado, R. Muñoz-Salinas, F. José Madrid-Cuevas, and M. Jesús Marin-Jiménez, “Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion,” *Pattern Recognition*, vol. 47, no. 6, 2014, pp. 2280-2292.

[21] D. Kim-Boyle and B. Carey, “3D Scores on the HoloLens,” in *International Conference on Technologies for Music Notation & Representation (TENOR 2019)*, Melbourne, 2019.

[22] D. Kim-Boyle, “96 Postcards in Real Color,” in *Proceedings of the TENOR2022 International Conference on Technologies for Music Notation and Representation*, Marseille, 2022.

HOLISTIC PERSPECTIVES: A REPORT ON SCORING BEYOND EUROLOGICAL TRADITIONS

Terri Hron
TENOR Network, matralab, Concordia University
Independent researcher
Montreal, QC
terrihron@gmail.com

ABSTRACT

In 2021, the TENOR Network supported a consultation with artists to investigate scoring practice beyond Eurological traditions towards a publication of edited interviews. This paper presents results from the initial round of interviews with a report on the emergent connections that brought out a relational ontology and a holistic perspective of scores. Starting with a critique of the composer-centered work concept, the author presents how consulted artists reflect on roles implied by scores, temporal considerations and definitions of scoring technology, and how these can be expanded with a holistic perspective. Orality, ancestral knowledge, witnessing practice and collective creativity are recurrent themes. The last section offers a number of ways to consider scores that might open the TENOR community to practitioners outside its current purview. Interviewed artists are quoted at length in anticipation of the publication of edited interviews.

1. INTRODUCTION

This paper offers a status report on a consultation project on scoring sound practice beyond Eurological practice funded by the TENOR Network at matralab. Within the scope of this project, I interviewed and consulted with 18 artists; my initial aim was to expand the TENOR community beyond its current cultural and geographical focus on Western European art traditions. During this ongoing project, through the discussions and further reading, I have started developing new definitions and boundaries around the notion of score that come together under a holistic approach. As these interviews are as yet unpublished, I often quote the respondents at length here to give their voices space to resonate.

2. CONTEXT & BIAS

2.1 The TENOR Network consultation project

My relationship to the TENOR community was, pre-pandemic, as the coordinator of the TENOR Network, and the part of that job that I have taken most to heart is trying to expand that community beyond its heavily Eurocentric research base. A Survey project was organized by the Network that revealed that a practitioners concerned with TENOR issues outside Western European traditions were not simple to find or connect with, and that a more invested, in-depth approach would be necessary. Meanwhile, my own practice and cultural work outside of TENOR was focusing increasingly on decoloniality, equity, pluralism and access, which similarly required intense investment, time and a critical stance towards existing methodologies and definitions. From the outset, then, I committed to proceeding with the project willing to allow the collapse of my plans and premises rather than seek confirmation, and to reconsider the very basic definition of score and its practice.

2.2 Questioning Eurological bias

In framing my initial research, overviewing the proceedings of TENOR conferences, and drawing up an initial list of respondents, I conceived of the project as an exploration of scoring beyond Eurological traditions, underscoring not only the Eurocentrism of TENOR research, but also the sensibility that Lewis describes [1] in the initial formulation of Eurological music, which answers challenges to the accepted narrative with racialized denial and unacknowledged appropriations. I would argue that there is an inherent Eurological bias in conventional and widespread definitions and examples of scores for creating and organising sonic experience. I consider the regular interchangeability of the term notation with score as underlining the assumption and privileging of written or marked forms of scoring. Furthermore, I see most canonical or historical narratives of the musical work, and by extension especially art music itself, celebrate the increasing expertise, precision, and affordances of writing (and by extension printing and eventually digital mark-making technologies): complexity, repeatability, and fixed ownership/capital. These narratives seem to follow on the tracks of the principles of the industrial revolution, with its divisions of labour and privilege within capitalist, colonial and dominant-culture enterprise. Such principles lead to investment in the standardisation and universalism that deeply affected and shaped the pedagogy, institutions, functions and creative tools of music and continue to do so. Although a worthwhile discussion of the intersections of these and the narratives around scoring are beyond the scope of this paper, they are some of the driving forces behind this research project itself, with a desire to "destabilize cultural hierarchies" and for "cross-cultural contact."[1]

- What happens if we pull at the strings that tie definitions of scores to reading or writing?
- What happens if we consider all the senses in the memory, definition and transmission of sound practices?
- What happens if we ask people positioned outside or troubling Eurological scores--by practice, by choice or by geography--to respond to these ideas?

These questions were the starting point for the Scoring beyond Eurological traditions consulting project.

2.3 Practitioners interviewed

The respondents included: composers whose practice started in the Western classical music tradition (WCMT) and moved into interdisciplinary or cross-cultural work (Sandeep Bhagwati, Linda Bouchard, Giorgio Magnanensi), visual or dance artists working with (sound) scores (Hannah Fischer, Charlotte Hug, Lou Sheppard), Indigenous artists (Suzanne Kite, Dylan Robinson), composers actively working on scoring outside WCMT (Cat Hope, Luke Nickel), artists from non-WCMT traditions (Kohei Nishikawa, BC Manjunath).[2] Some respondents fit into several categories.

Once I had completed a first round of interviews and their transcriptions, I noticed that the emergent cohesive tissue that could bring many of the reflections together was a more holistic notion around scoring on several different levels. What follows are some initial observations with connections to certain respondents—ideally each could be the subject of its own paper or chapter. Ideally, a continuation of the project would include their reaction to these ideas and further clarification of how their practice might enlighten this perspective.

Because most of the categories we use to speak about scores in music, especially in academic or research contexts come from WCMT, I often quote the respondents at length because I did not propose to them neat categories to respond to, and therefore my groupings of their observations are not always very succinct.

3. HOLISTIC ROLES

In many scores, the sonic exchange described assumes certain roles, whose boundaries can be fluid and overlapping, and include initiator(s), creator(s), facilitators(s), participant(s) and witness(es).

3.1 Beyond a composer-centered work-concept

In Eurological music and its scoring practice, however, these assumptions create a specific definition. Lydia Goehr argues that *Werktreue* and the work-concept took firm grip around 1800 and focused almost all creative energy on fixing the roles of the composer as initiator/creator/meaning-maker, with the executant performer preferably participating as an invisible vessel for the composer's intention [2]. The target consumers, a learned audience, as witnesses, consumed the composer's meaning, possibly acquiring prestige by subsidizing the composer's genius through patronage. This increasingly firm division of formerly – and in other cultures often still – more fluid labour roles, alongside the creation of the cultural capital of "art music," coincides with industrialization, imperialism, and universalism.[3] Unsurprisingly, the standardization and normalization of Eurological music's notation developed in parallel. As Jesse Stewart notes, "[i]n general, the field of new music has actively maintained hierarchies of this sort. This is due in large part to the institutional contexts in which new music continues to circulate and be discursively constructed, notably within university music departments, festivals of new music, and concert halls designed for performances of Western classical music and/or chamber music." [4] I would argue that what is true of new music is true of its scoring, and by extension the design of software for that end and the standards by which we evaluate merit.

Many other configurations of roles do exist, however. Oral tradition might replace the composer with legacy and/or tradition, a connection to ancestral knowledge. When I questioned him about how music is transmitted in the different Japanese flute traditions he practices, Kohei Nishikawa kept circling back to the hereditary nature of the music and its unbroken practice. In *Hungry Listening,* respondent Dylan Robinson explains that the transmission of sound or songs themselves might be considered living entities that can only be shared through embodied experience [5]. In such situations, could ceremony be the only appropriate score format? In describing her scores to me, Indigenous artist Suzanne Kite scores highlighted the importance of the participatory witnessing role, with a special focus on Lakota semiotics and ontology: "I'm very much interested in scoring and manipulation and the arrangement of bodies and my body's relationship to the audience.

[1] For an in-depth discussion of these issues, see Jesse Stewart's "Intervections" [2]. The arguments and connections he makes around "new music", most clearly illustrated in the vector diagram on p. 324, are valid for scoring and TENOR as well.

[2] In order to find artists from non-WCMT traditions that would be able to speak to me about scores (in English or French), I consulted with a number of (white) musicians with a specialty and knowledge of non-Western traditions, specifically Japanese music. These included Nancy Beckman & Tom Bickley, Ralph Samuelson and Elizabeth Brown, who eventually connected me with Kohei Nishikawa and others who I am yet to interview. I also consulted with Craig Vear, to know to what extent his recent work on Digital Scores takes non-WCMT into account.

[3] "In particular, music's growing reliance upon the score is almost unanimously understood as a major development in the advent of the musical work. In reality, the score is only one part of a much larger story, which must necessarily also include issues such as compositional (or authorial) control, the possibility of repeatability, the notion of permanence, and the emergence of aesthetic autonomy as a core European ideology." [3] I found Steingo's analysis of and expansion upon Lydia Goehr's placement of this shift towards conflating the musical score and the musical work at the beginning of the 19th century particularly useful, including the footnotes that contextualize the relationship between the work and commodity. The fascinating nexus of score, performance and the industrial revolution is a topic beyond our scope here; Goehr provides a starting poing: "[A]s long as the composers provided incomplete or inaccurate scores, the idea of performance extempore could not acquire its distinct opposite, namely, the fully compliant performance of a work. Such a contrast emerged fully around 1800, just at the point when notation became sufficiently well specified to enable a rigid distinction to be drawn between composing through performance and prior to performance." [2]

The most important thing to say about my composition practice is it's very much focused on a circular relationship between my body and potential non-human beings." [6] So while most Eurological scores focus on doing, what might come of considering scores for listening, reacting or witnessing as a creative act? Likewise, perhaps even the binary of oral/notated traditions is anchored in Eurological ontologies or even the prevalance of European languages in writings on music transmission has narrowed our understanding of both scores and roles: "Western research that serves to extract and externalize knowedges in categorical groupings aligns well with the categorical premises of Western languages" [7].[4]

3.2 Annotation and community

A holistic discussion of roles also involves the experiences of performers and researchers. The WCMT, at least since *Werktreue*, concentrates on scores as the composer's domain, with less attention, value and tools created for annotations. When important enough, these are upgraded to "arrangements." In my conversation with Cat Hope, the importance of annotation came up with relation to upcoming versions of the Decibel Score Player, which would add the important annotation functionality, making it a tool not only for composer but also for performer creativity [8]. Likewise, validating work on scores beyond the composer, (critical) editions could work not only to distinguish among manuscripts, but to support the importance of contextual and performative informations for better and broader potential for sonic transmissions: "editorial art [is] just one step in the imagination of a musical score, using the edition not to satisfy the ned of the user but to encourage the user to question, explore and reinterpret. Editing music is an act of creative interpretation as criticism" [9].

To sum up the importance of people and community in the transmission of sonic ideas, I offer two more moments from the consultations that decenter the composer within the creation and discussion of scores. Craig Vear affirms that: "For me, as somebody who came in through theater and the world of performance, having spent 20 years or so doing that, the notion of any discussion of transference of ideas between people that doesn't take those people into consideration, or put that in prime place is just ridiculous" [10]. And weaving in some Indigenous ontology, Suzanne Kite explains: "I just had a really long conversation with Santee [Witt] about this, it is definitely related to Lakota concepts of truth and epistemology where their relationship to data or fact is not remotely similar to Western European ideas of fact. One of the scholars I read on this subject is Jim Cheney, who says that things are true in these communities, if they're responsibly true for the whole community. So you take that concept, and then you get all the way to me trying to make scores, and there's no way I could tell a musician what to do or that the note they played was wrong. I couldn't even begin to have a relationship with notes like that" [6].

[4] Kovach's later paraphrasing of R Struthers also underlines that discounting oral scores is unecessarily exclusive: "Given the philosophical basis of a complementary, non-binary thought pattern, it makes sense that narrative encased in the form of oral history would be the natural means to transmit knowledges" [7].

4. HOLISTIC TIME

Many scores manipulate and shape sound in time, in the lived experience. Most Eurological scores are to be interpreted sequentially with a maximum of synchronicity. In the WCMT, the technologies of writing and distributing scores, extended by the increasing normalization of printing, allowed for and encouraged increasing polyphony, complexity and repeatability, creating an addiction to specific synchronous sonic moments [11].

4.1 Spontaneous and protracted practices

Other temporal organizations exist, however, where moments might be built or emerge spontaneously or through long-term interaction. A a Canadian artist working in interdisciplinary audio, performance and installation based practice, Lou Sheppard describes this space of devising: "I've been pushing my work more towards trying to figure out how to notate time and space within some system that people can look at and fall into and then having a conversation about what happens within that time and space and that being more what the score is, rather than actually this sound" [12].

In other contexts, music can be created coordinatedly or in quasi unisons using a common language, formulas, site or occasion. Karnatic mridangam player and konnakol virtuoso, BC Manjunath describes learning a common inter-arts language: "When I went for the first time to play for a dancer in India, I played just as I would for classical music, and they said, 'No no no, it doesn't work like that. You're so used to playing to the vocalist and play along with that.' But here, the main person I had to be watching was the dancer, and not even the dancer, the feet of the dancer. The rest of the body might be doing something else. That's why I would put Bharatanatyam as the highest form of polyrhythm. They're probably the masters of polyrhythm, but they don't know how to explain it… you have to go there, be with them for a long time and then you try to understand, decipher it yourself" [13].

4.2 Avoiding fixed timelines

Unlike such long processes of coming into unison, interdisciplinary artist Charlotte Hug developed a form of scoring that allows for individual flexibility in timing within a group setting of diverse artists and makers: "It's really fantastic to work with InterAction Notation (IAN) in an intercultural context, especially because of the timing. In conduction, you still have a certain kind of timing even if the conductor is listening and very receptive. In an intercontinental, inter-cultural context, the feeling of a timing can be very, very diverse. And I had this experience with a dance company in South Africa, where even each dancer could dance the Son Icon[5] in their own tempo and timing and IAN has this particular, precise quality that the timing is flexible. Each person who has the signal for the next sign has the responsibility for how long the section should be.

[5] Son Icons are a hybrid score/visual artwork developed by Charlotte Hug in her practice. See https://www.charlottehug.com/en/about-me/son-icons-gallery

The duration that creates had me very much puzzled sometimes: I thought 'Oh, it has to move on,' but then the person just kept going and it was wonderful. So I feel IAN is a real melting point for cultures to understand different timings and also to invent symbols because you might need them" [14]. Indeed, the linearity and often fixed timing that WCMT notation assumes and imposes is one reason why verbal, visual and digital technologies can be so liberating.

4.3 Beyond a single human lifetime

Scores offer relationships beyond human lifetimes. If the category of score were to include oral transmissions beyond individual lifetimes, such scores could carry legacy, tradition, performance practice and the past itself in ways we do not often discuss in TENOR. The notion of *rag* in India or *shōga* in Japan are both systems that connect and develop specific sonic ideas over time and decenter the notion of single authorship. When attempting to correlate score with the various Japanese flute traditions he practices, Kohei Nishikawa said: "I still understand the onomatopoeic phrases [*shōga*] as coming down from a very, very old hereditary system. That is very important, like the score in Western music: a score coming from a composer doesn't change. But my master, perhaps he or she plays differently and they teach me that they are person with their own personality. I can read the 'score' and understand what the composition wants, and still realize my master has their own personality" [15]. Complementing this is Morita Toki's *shōga* research: "the mnemonics become a medium that transmits musical substances. She adds that those who have experienced oral transmission can look at the *shōga* and hear oneself chanting the shōga and thereby reimagine one's own performances" [16].

In Eurological, WCMT practice, contemporary scores often endorse innovation; this novelty obsession leads to scores possibly only meant for the future, created by an avant-garde of "visionaries," to be understood and valued posthumously. Outside this paradigm, there are transmissions of (sonic) ideas that rely on repetition, participation and the embedding of collective description, carried across generations and that survive if they adapt and inspire for each present moment. Dylan Robinson: "We have protocol, which is a guide, that is always still in relation, it's not a guide that says, 'It always needs to be done in this way.' I think this is actually the one of the ways in which protocol is misunderstood through a Western framework: a protocol is understood sometimes as the law or the unchanging method to do something to be in good relations, but we understand protocol as always shifting, as a score with a wide amount of variation, that actually seeks to standardize maybe only a value or a sentence, that only serves as a mnemonic for value, that is expanded quite a bit in relation to the context specificity of what we're doing" [17].

4.4 Forgetting with time

In another different realm, there are (oral) scores that are meant to exist only ephemerally, both in time and memory, again decentering the composer. This is true for the work of Luke Nickel: "In the end, I arrived at using my own voice and recording instructions and poetic concepts, then transmitting that to either a musician or to an ensemble's members separately, who would then communicate it to each other. And those were temporary, only to be listened to once and then they would disappear. The forgetting/memory of the person who listened to it also became its owner. In a weird way, they knew more about it than me because they had listened to it more recently than I had. There was a difference in power, where I suddenly stepped back a bit" [18].

If a holistic conception allows for and seeks out all these different ways of organizing sound in and over time, it should also consider imagining beyond human generations into geological and cosmic temporal relationships in which we also participate. This might bring non-human beings and agents into the score-making potentiality. Further research is ongoing and needed in this area.

5. HOLISTIC TRANSMISSION TECHNOLOGIES

5.1 Notation ≠ Scoring

Now that the agents involved in the transmission of sound ideas can be expanded, perhaps also the definition of the technologies used might be as well. First of all, let us address the interchangeability of the term notation and score in most conversations about Eurological music, and the subsequent privileging of mark-making technologies in the definitions of sound transmission. In my consultation with Craig Vear, he agreed with my discomfort with this lack of holistic vision: "I think there's a real distinction between the score and notation; they are two completely separate things. Notation is a very closed space. It's a very, like you say, privileged vehicle with which to communicate ideas, because it presumes the other person you're communicating ideas to knows the codes. But actually, my notion of the score is just a communications interface, which could be verbal, oral, it could be tactile, it could involve robotics, or motorizing wheelchairs." [9] What follows is that familiarity with the interface of exchange is the key.

5.2 Fluency & Musicianship

Often the word *literacy*, when speaking musically, is used to denote familiarity with conventional WCMT notation, once again assuming reading and writing as the only means. The word *fluency* might offer a broader fit, and move out from (mostly) reading- and writing-based technologies to encompass other senses and ways of knowing.[6] What might fluency aspire to within scoring technologies? Cat Hope connects knowledge of the interface with musicianship, which "has to do with their craft and training… I know that my pieces are made for trained musicians (not necessarily classically trained). People who have a very deep musicianship, whether it be Western or any other kind of musicianship. I really believe in musicianship. That's what interests me, is drawing on musicianship." [19] How might the notion of fluency act as a way to

[6] Quote from adrienne maree brown on water.

respect musicianship in communities and bodies of knowledge and encourage deep study? A holistic view of fluency – or rather fluencies – celebrates multiple knowledges and means of transmission, multiple technologies and communities.

5.3 (C)overt interfaces

Widening our understanding of scores as technologies of transmission or the interface for (sonic) ideas does not necessarily mean a more universal perspective. Respecting scores might entail the privacy or primacy of connection within a specific group. Just as with languages, the worldview that generates such scores is not always meant for translation, at least not without initiation. In speaking of what she considers to be a successfully functioning score, Suzanne Kite explains: "when I make things, there really must be layers. One of the layers must be easily interpretable by my community, my family. Obscurity of meaning doesn't happen for them. It happens to everybody else. That's how I know I'm successful in my meaning-making or lack thereof. When I made Listener, there's a text that goes with it, and it was up in a space I was performing in and some family came in. A woman who had forgotten she was my family member came in and she could interprete the entire piece that was up, she knew every reference, she knew what was sacred text for us, she knew what was a dream of mine, she knew references. It was clear as day to her. Then I did this piece in Austria, three or four times: just meaningless. It was horrible. I knew I was good piece, because that's what I want." [6]

Other times, composers fluent in the technologies of dominant culture might use these to bring about sonic events that might subvert the usual directions of those technologies and even work towards healing. This seems to me to be the case in Raven Chacon's (in progress) *Amercian Ledger* series,[7] where elements of Eurological scoring and accounting are used to recount and grieve sonically the forced migrations and violence towards oppressed communities. In this series, the score is to be present in the form of a flag, a billboard, a blanket, a newspaper; this along with the *ledger* in its title creates an uneasy relationship between the score, the accounting and reality, to say the least.

Culture-, place- and kin-specific transmission technologies and interfaces, and their inversion/subversion, do not narrow the possibilities of scoring. On the contrary, refusing the imposition of a dominant convention or language of research might make a broader field of tools emerge.

5.4 Performance Practice

In WCMT, instrument-specific notation – systems of marking legible only to practitioners of one instrument – were all but sidelined until the resurgence of extended techniques and electronic instrument scoring, which have mostly eschewed standardization. In other parts of the world, instrument-specific notation or language is fairly common, and is intricately tied to oral tradition, serving as a mnemonic aid. My conversations with BC Manjunath about *konnakol* and Kohei Nishikawa about various Japanese flute traditions confirmed that these auxiliaries to scores abound. Indeed, even in medieval (and to some extent pre-industrial Europe), it is assumed that what is marked is but one part of the transmission—the instruments themselves and the oral tradition would have been essential interpretive collaborators.[8] Once again, such instrument-specific scores assume that there is no bypassing the performer and their intimate relationship with the interfaces of both the score and their instruments.

5.5 Current and future technologies

To take this further into digital technologies and artificial or non-human intelligences, possibly in augmented or virtual realities, there is an expanse of opportunities to engage not only with other senses but several at once, multiplied by the possibilities of poly-dimensionality, multiple formats, transdisciplinarity and more. In these early years of digital realities, a holistic approach to technologies of scoring can work to disrupt normative, universalist, capitalist and/or colonialist values (and perhaps if we succeed in the virtual/augmented world, we can do so IRL).

6. HOLISTIC SCORES

The holistic perspective that guided this consultation process was tuned towards listening for the emergent properties, qualities and characteristics that are more likely invisible when focussed on individual and/or dominant culture practice. Despite its broad aspiration, however, the choice of consultants is still specific to the positionality and network of a white woman settler musician interviewer, working adjacent to academia, with curiosity but limited knowledge of non-eurological, even non-WCMT traditions. Nevertheless, a holistic perspective is assumed possible even from such a local node, acknowledging that this is but one iteration of something much greater, an invitation.

Similarly, a holistic score is not itself an object or a goal, it may be one iteration from within a perspective of something larger. It may be a conventionally-written eurological score from the sixteenth century performed with an awareness that much about the organization and quality of sound remains orally transmitted. It may be the sense for what does or does not fit melodically within the pitch and ornament combinations of a certain *rag* or *shōga*. It may be the ceremonial and spiritual context or protocol within which a sound or song can exist. Any score or organization of sound or music to share can be understood holistically, and therefore I offer no other working definition for a holistic score. There are, however, many useful ways to look at and experience scores that add to a holistic definition.

[7] To view the first two scores (at the time of writing) of this series, go to Chacon's website: http://spiderwebsinthesky.com/music/

[8] For a more complete discussion of early European scores, see for example Leo Treitler's chapter "What kind of a thing is musical notation?" where he writes: "Although melodies were represented for centuries by [neumes], the other side of the 'mnemonic' assessment–that th transmission and the singing of the melodies would have depended also on unwritten processes in collaboration with which the neumes must have been adequate–was long ignored and is still resisted in some quarters" [20].

The consultations revealed understandings of scores that might help broaden perspectives or definitions and include a larger number of practices and practitioners. What follows here is an incomplete list of some configurations or practices of score, which came up: scores as *mnemonic devices,* as *ancestral knowledge*, as *spaces of resonance*, as *interfaces*, as *boundary objects*. This pluralism encourages relationships beyond individual experience and facilitates processes rather than outcomes. This points to a relational ontology when considering the nature or definition of scores, wherein, depending on the relationships between those involved, multiple answers are encouraged and may apply.

6.1 Mnemonic Device

As in the pre-industrial Eurological tradition or in the instrument-specific notations mentioned earlier, scores in many cultures serve as memory aids, simply to help either a single practitioner or a lineage recall knowledge that was communicated orally. Orality, therefore, is a fundamental component in this type of transmission, repeatedly underlined by consultants Kohei Nishikawa, Nancy Beckman, Elizabeth Brown & Ralph Samuelson (Japanese music), and BC Manjunath (Karnatic music), who all describe mark-making systems as essentially auxiliaries to teaching within a guru or master system. One might argue that the score itself is a combination of both these rudimentary markings and the explication of a practitioner with acquired knowledge. Luke Nickel offers a different approach to the notion of scores as imperfect and fallible mnemonic devices as his practice often relies solely on the incomplete memory of collaborators [17]. Both he and Cat Hope point to Eliane Radigue's oral transmissions which rely on the retention of the performer and their own mnemonic devices in the score of the work. It also implies a nascent performance practice for those works, knowledge held and carried by the performers to be hopfully transmitted orally onwards.

What might happen if we experiment with scores as mnemonic devices rather than considering that somewhat antiquated? This question came up in the conversation with BC Manjunath, when he marvelled how recording and social media technologies are affecting the speed and dissemination of formerly individual oral transmissions, with all the benefits and risks that entails [13]. *Konnakol* in general – and BC Manjunath's YouTube feed in particular[9] – has seen a mushrooming of practitioners since its availability online, so much so that since the pandemic, a new *konnakol* competition has been established.

On the other end of the speed spectrum from digital connectivity is Kundera's argument for the place of memory in a too-frenetic world: "there is a secret bond between slowness and memory, between speed and forgetting… The degree of slowness is directly proportional to the intensity of memory; the degree of speed is directly proportional to the intensity of forgetting" [21]. Perhaps the slowness of scores written in memory and the use and interpretation of memory devices is an asset to investigate as well.

6.2 Ancestral Knowledge

More time-defying qualities come up when considering trans-generational knowledge-holding and transmission practice. While innovation and individuality is often a focus in scoring research, with an arrow of development pointing forwards, ancestral – and what Kohei Nishikawa describes as hereditary – knowledge seems rather to spiral and orbit around communities. Once again related to orality and transmission beyond the fixity of a medium, ancestral knowledge is therefore quite challenging for colonial or authorially-minded score and research understanding. Likely the strongest conclusion and/or suggestion that emerged from these interviews, is that the concept of scores would benefit from expanding to engage with and consider ancestral knowledge traditions, since these are quite prevalent in pre- and extra-colonial communities and practices. How to initiate such an expansion remains much murkier as yet, but that's perhaps understandable from within a research landscape so dominated by a different mindset, but conversations about intergenerational score/knowledge holding could be a good starting point. Finding English- or French-speaking interviewees for this has been a challenging (and the colonial irony is not lost on me) but most necessary investment.

Recognizing sonic ideas/transmissions that are part of traditional knowledge as scores, or more broadly, as protected materials is also an issue that can have wide-ranging implications. Some can impact what spaces are appropriate for the singing or playing of such scores, as Dylan Robinson points out in his critique of the concert ritual and halls and the kind of mutual responsibilities, the "new social contract where individuals become accountable in the act of witnessing, of face-to-face encounters," that are inherent in certain Indigenous sound practices [5]. Other issues might involve how to expand and challenge individual authorship so as to legally protect musical expressions as creations belonging to all the members of a community. Copyrightability is a controversial issue: some argue that there is precedent in protections created, curiously enough, for traditional knowledge pertaining to medicinal plants and other resources [22]. However, copyright is underpinned by notions of 'originality' and 'personality', which are again tied to authorship and commodity in a way that essentially devalues knowledge held in community.

6.3 Space(s) of Resonance

Thinking beyond, yet possibly also inextricably linked to language and epistemologies are the many scores that require or encourage response, reception and sympathetic (or perhaps antipathetic) vibration from those using them. Pointing to the work of Hartmut Rosa, Charlotte Hug uses the score as a space of resonance as a guiding principle in her intercultural work, where the oftentimes unexpected

[9] For those curious about the hybrid language/instrument/score nature of *konnakol*, see https://www.youtube.com/c/ManjunathBC The number of views is quite astonishing!

reactions that emerge become integral in her understanding of the score's potential. Sandeep Bhagwati describes his scores as relational, resonating not only between those who play them, but also in their material, beyond the sonic realm: "An important insight that I had was that a score can activate much more than the musician and really engage with the human being in general, with the biome, and so on" [23].

Likewise scores that are created in relation to specific places and their histories, like many of Raven Chacon's works, require the particular resonances of those places and histories in their performance. The resonance or not of places, peoples and their epistemologies comes up also, as mentioned above, in Suzanne Kite's scores.

6.4 Interface

Both Suzanne Kite and Craig Vear give a definition of score as an interface, a place of linkage. If an interface is a shared boundary and place of information exchange, a locus of interaction between a number of communities or systems, then considering scores as interfaces prioritizes the facilitation of relationships and the kinds of protocols that support interaction. This underlines collective agency and creation. Linda Bouchard also talks about interface as one of the iterations of the Ocular Scores project, where a performer plays the score-making device as an instrument, connecting the input and outputs of the other instrumental performers [24]. Charlotte Hug calls InterAction Notation an interface between media and disciplines. An interface allows for a much broader conception and agency for the score, far less dependent on *chronos* but engaged with *kairos*, a quality that appeals to Sandeep Bhagwati, who is likewise interested in devising systems rather than focusing on sounds.

Once again, score as interface implies a relational ontology, where the focus is on connection and collective creativity, rather than a unidirectional device issuing from an individual.

6.5 Boundary Object

Adjacent to the interface, I would suggest considering the score as a boundary object, both plastic enough to adapt to local needs and constraints of their multiple users, yet coherent enough to maintain a common identity. Boundary objects are artifacts or ideas that help people from different backgrounds come to a shared understanding. While thy are weakly structured in common use, they become strong in individual-site use. Their structure is common enough to more than one world to make them recognizable, a means of translation. Raven Chacon's *American Ledger* Series springs to mind as a good example of score as boundary object, made even more powerful in its function as a historical and site-specific calling to account through the personal investment of the performer and the collective. This definition of a score is also in dialogue with notions of standardization or convention, which often are not focused enough on plasticity and local needs or on making many worlds recognizable.

Furthermore, considering scores as boundary objects makes it possible to use them as a site of collaboration, which would give an important place to annotation and the collective improvement and/or critical edition of scores.[10]

7. CONCLUSIONS AND NEXT STEPS

If anything, these initial interviews established that expanding potential of scores beyond Eurological conceptions offers panoply of options. This report is by no means either a comprehensive account of them or even of the many insights shared with me. The main difficulty in launching such research without the expectation of a certain result is that only after a first round of interviews does a pattern or a cohesive story begin to emerge. Yet this is precisely the attraction of a pluralistic and holistic approach to definitions and conceptions of scoring: it strives for greater awareness and openness – to connection, participation and difference. Starting from consultations with an eclectic group of voices, some of whom seem underrepresented in studies and research groups focused on the scoring of sound, it quickly became clear that the hermeticism of the notation/score research community might be related to definitions of scores themselves, and that expanding those might help bring in practices and practitioners outside WCMT. This often involves pushing against the categories, methods and values, not to mention the languages, of Eurological thinking, which certainly is slowed in this case by my background and belonging to white Western musical culture. It has, however, offered many avenues of inquiry that I can follow up on, as well as a much better idea of the amount of time it can take to find respondents I can exchange with—and for me to learn enough about their practice to ask reasonable questions. This is therefore but the very beginning of what is an incredibly vast pool of fascinating practitioners. Furthermore, this research and report on scoring is in dialogue with fundamental questions about what music is and what sounds and music are of interest,[11] yet diving into that is beyond the scope of this paper. Finally and doubtless, communities of sound and their modes of transmission will continue to change as they encounter digital technologies as well as (artificial) intelligences or beings—these are areas of specific interest to me and where I think such expanded definitions might serve.

Acknowledgments

My research would absolutely not have been possible without the investment and support of the TENOR Network, based at matralab, at Concordia University, Montreal, and the encouragement of Sandeep Bhagwati. I am most grateful to have been able to continue working within this academic environment and benefit from the generosity of all the consultants interviewed.

[10] Indeed, the only mention I found of scores as boundary objects was in the ethnographic research on annotation by Megan Winget [25].

[11] Specifically in relation to Georgina Born's work on relational musicology [26].

8. REFERENCES

[1] George E. Lewis, "Improvised Music After 1950: Afrological and Eurological Perspectives," in *Black Music Research Journal*, vol. 16, no. 1, 1996, pp. 91–122.

[2] Lydia Goehr, *The Imaginary Museum of Musical Works: An Essay in the Philosophy of Music*. Revised ed., Oxford University Press, 2007.

[3] Gavin Steingo, "The Musical Work Reconsidered, In Hindsight," in *Current Musicology 97* (Spring 2014), p. 81.

[4] Jesse Stewart, " 'Intervections' " in *Contemporary Music Review*, vol. 29, no. 3, 2010, pp. 323-335.

[5] Dylan Robinson, *Hungry Listening: Resonant Theory for Indigenous Sound Studies*. University of Minnesota Press, 2020.

[6] Suzanne Kite, Interview, conducted by Terri Hron, 27 April 2021.

[7] Margaret Kovach, *Indigenous Methodologies: Characteristics, Conversations, and Contexts*. University of Toronto Press, 2009,

[8] Cat Hope, Interview, conducted by Terri Hron, 9 March 2021.

[9] Mark Clague, "Portraits in Beams and Barlines: Critical Music Editing and the Art of Notation," in *American Music*, vol. 23, no. 1, 2005, 39–68.

[10] Craig Vear, Interview, conducted by Terri Hron, 22 February 2021.

[11] Sandeep Bhagwati, "Prana," *Changing Places Sensing Times Lecture Series,* 19 May 2021, matralab, Montreal.

[12] Lou Sheppard, Interview, conducted by Terri Hron, 31 March 2021.

[13] BC Manjunath, Interview, conducted by Terri Hron, 27 January 2021.

[14] Charlotte Hug, Interview, conducted by Terri Hron, 23 February 2021.

[15] Kohei Nishikawa, Interview, conducted by Terri Hron, 13 April 2021.

[16] Paraphrased in Mariko Anno, *Piercing the Structure of Tradition: Flute Performance, Continuity, and Freedom in the Music of Noh Drama,* Cornell East Asia Series, an Imprint of Cornell University Press, 2020.

[17] Dylan Robinson, Interview, conducted by Terri Hron, 30 April 2021.

[18] Luke Nickel, Interview, conducted by Terri Hron, 9 April 2021.

[19] Cat Hope, Interview, conducted by Terri Hron, 11 January 2021.

[20] Leo Treitler, *Reflections on Musical Meaning and Its Representations*. Indiana University Press, 2011.

[21] Milan Kundera, *Slowness*. Translated by Linda Asher, HarperPerennial, 1997, p. 34.

[22] Giovanna Carugno, "How to Protect Traditional Folk Music? Some Reflections Upon Traditional Knowledge and Copyright Law," in *International Journal for the Semiotics of Law - Revue Internationale De Sémiotique Juridique*, vol. 31, no. 2, 2018, pp. 261–274.

[23] Sandeep Bhagwati, Interview, conducted by Terri Hron, March 25, 2021.

[24] Linda Bouchard, Interview, conducted by Terri Hron, April 2, 2021.

[25] Megan A. Winget, "Annotations on Musical Scores by Performing Musicians: Collaborative Models, Interactive Methods, and Music Digital Library Tool Development," in *Journal of the American Society for Information Science and Technology*, vol. 59, no. 12, 2008, pp. 1878–1897.

[26] Georgina Born, "For a Relational Musicology: Music and Interdisciplinarity, Beyond the Practice Turn," in *Journal of the Royal Musical Association*, vol. 135, no. 2, 2010, pp. 205–243.

96 POSTCARDS IN REAL COLOR

David Kim-Boyle
The University of Sydney
Sydney, Australia
david.kim-boyle@sydney.edu.au

ABSTRACT

This paper describes the development of *96 Postcards in Real Color* (2022), a virtual reality (VR) work for up to eight singers which features a three-dimensional immersive score generated from image captions scraped from Instagram. The poetic inspiration is briefly presented and various technical elements of the work's design, development, and implementation are discussed including how the *Selenium* and *Beautiful Soup* Python libraries were used to scrape and parse images and text from Instagram, and how the multiplayer framework of the work is supported with the *Photon Unity Networking* SDK. Various user experience (UX) considerations influencing the work's design are discussed, together with a discussion of future research directions.

1. POETICS – BACKGROUND

In 1978, French writer Georges Perec composed his playful *Deux cent quarante-trois cartes postales en coulers veritables* [1], a series of postcard texts generated with a set of simple combinatorial rules [2]. Each text describes a location, either a city, region, or hotel at which various activities and entertainments occur, before signing off with a farewell, see Figure 1.

> We're camping near Wood's Hole. Sunning ourselves. Lobster at every meal. I've caught a salmon. Many regards.
>
> A big hello from Biarritz. So nice letting yourself go brown in the sun. I've done a bit of sailing. Love.
>
> We've finally landed in Nice. Lots of lazing about and sleep. Really nice (despite the sunburn). Love.

Figure 1. Sample postcard texts from Perec's Deux Cent Quarante-Trois Cartes Postales en Coulers Veritables, trans. by John Sturrock [3].

The combinatorial writing techniques of Perec and the Oulipo (*Ouvroir de littérature potentielle*) school [4, 5] of whom he is an exemplar member, have produced several widely cited works including Perec's 1969 novel *La Disparition*, which omits the letter 'e', and Raymond Queneau's *Cent mille milliards de poémes* (1961), which is derived from a set of ten sonnets with interchangeable lines. Contemporary readers would perhaps be most familiar with the Oulipo through the work of its current President Hervé Le Tellier whose recent *L'Anomalie* (2020) was one of the bestselling novels of all time in France. Oulipo's impact outside literature has also been felt through the loose assembly of schools in other disciplines each designated with its own title, e.g. Oumupo (*Ouvroir de musique potentielle*), Ouphopo (*Ouvroir de photographies potentielles*), and Oupeinpo (*Ouvroir de Peinture Potentielle*) [6]. More broadly of course, constraint-based creative practice has a long tradition in a diverse range of practices from film [7, 8] and architecture [9], through generative art and music [10].

The constraint-based approaches of the various Oulipo schools have been a particular inspiration to the author with their application of deceptively simple operative constraints offering unexpected forms of creative expression. In *96 Postcards in Real Color*, henceforth *96 Postcards*, for up to eight singers, I sought to adopt a similar constraint-based approach to musical organization, although contemporizing the medium to explore ways in which Instagram can provide source material for a three-dimensional (3D) immersive performance score presented to singers in VR. In a manner reminiscent of Perec's combinatorial processes, in *96 Postcards* text captions are scraped from 96 unique locations, as many as possible of which are chosen from locations listed in Perec's original text. Three locations are scraped for each letter of the alphabet to provide the first 78 data points, e.g. Ajaccio, Antibes, Alhambra, Balearics, Biarritz, Berghof, Cannes, Cyprus, Calabria, etc., while the remaining 18 locations are randomly chosen from this set. The text captions that accompany each of these 96 posts are analyzed with various natural language processing (NLP) techniques and used to define the musical structure of the work.

2. INSTAGRAM SCRAPING

Through their respective Application Programming Interfaces (APIs), data scientists have been able to explore how social media such as Instagram, Twitter, and Facebook can offer unique insights into social, economic, and cultural trends. Indeed, a veritable cottage industry of data visualization has emerged over the past decade which seeks to make the vast swathes of data and relationships they present, aesthetically pleasing yet informative and functionally useful. In my recent creative work I have sought to explore how the contents of this data and the way it is propagated and transformed through the social metaverse can provide innovative modes of musical organization, especially in extended reality projects such as *The Twittering Machine* (2021), for HoloLens 2 and prepared piano [11], and more recently in the present work *96 Postcards*.

In 2020, Instagram deprecated its public API, replacing it with an Instagram Graph API which can only be used to gather data and provide analysis on a user's Instagram Business account. Unlike public APIs developed by platforms such as Twitter, this severely inhibited the ability to perform any social media analysis through the platform. This restriction was exacerbated by an additional update to Instagram's terms-of-service that expressly prohibited the scraping of data or images through any automated service. Despite these constraints and the lack of a public API, workaround techniques have been developed for those conducting social media analysis although such techniques come with the inherent risk of having accounts banned and even exposing a user to potential litigation, although this is widely acknowledged to be a legal grey area. One such technique is with the Python library *Selenium* which provides an automated way of managing web searches.

For *96 Postcards*, *Selenium* automates a search on Instagram for posts at 96 unique geographic locations, e.g. Biarritz or Nice, by invoking and driving Google Chrome. This requires installation of the open-source tool *ChromeDriver*.[1] The critical parts of the Python code for conducting this search are presented in Figure 2. Note that for *96 Postcards* a dummy Instagram account was created from which all Instagram data is obtained.

```
#specify the path to chromedriver
driver = webdriver.Chrome('Path to ChromeDriver')

#open the webpage
driver.get("http://www.instagram.com")

#target username
username = WebDriverWait(driver, 10).until(EC.element_to_be_clicka-
ble((By.CSS_SELECTOR, "input[name='username']")))
password = WebDriverWait(driver, 10).until(EC.element_to_be_clicka-
ble((By.CSS_SELECTOR, "input[name='password']")))

#enter username and password
username.clear()
username.send_keys("account_name") #Instagram account name
password.clear()
password.send_keys("account_password") #Instagram account password


location = "Cyprus"
driver.get("https://www.instagram.com/" + location + "/")
driver.execute_script("window.scrollTo(0, 2000);")

#select images
images = driver.find_elements_by_tag_name('img')

#images = driver.find_element(By.TAG_NAME, "img")
images = [image.get_attribute('src') for image in images]
```

Figure 2. Python code for automating an Instagram search on "Cyprus" with the *Selenium* library and *ChromeDriver* open-source tool.

HyperText Markup Language (HTML) data returned by *Selenium* is analyzed using the *Beautiful Soup* library,[2] which provides an efficient, powerful means of extracting data from HTML files. The text caption and URL of the Instagram post is saved to a .csv file, see Figure 3, which is then called to generate the immersive score displayed in VR space within the Unity3D platform. The image associated with the post is also saved.

```
#urls of all scraped images
posts = []
links = driver.find_elements_by_tag_name('a')
desc_texts = []

for link in links:
    post = link.get_attribute('href')

    if '/p/' in post:
        posts.append(post)

for i in range(len(posts)):
    print(posts[i])


#parse html for descriptive texts
for post in posts:
    driver.get(post)
    soup = bs(driver.page_source, "html.parser")
    desc_text = soup.find("title")
    desc_texts.append(desc_text)

for i in range(len(desc_texts)):
    print(desc_texts[i])


df = pd.DataFrame(list(zip(posts, desc_texts)), columns = ['URLS', 'Texts'])
df.replace(to_replace = "<title>", value = "")
df.replace(to_replace = "</title>", value = "")
df.to_csv(96Postcards.csv')
```

Figure 3. Python code for grabbing Uniform Resource Locators (URLs) and text captions.

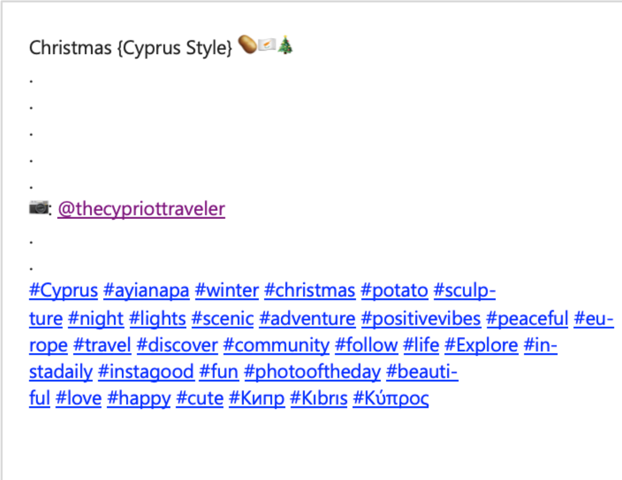
3. VISUALIZATION AND MAPPING STRATEGIES

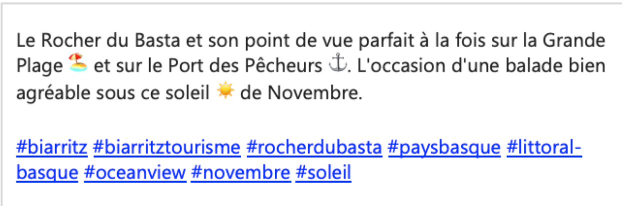
After collating the Instagram text captions from 96 locations, a textual analysis is performed to obtain various formal and semantic properties which are then mapped to features of the performance score. In *96 Postcards*, this analysis is performed with a combination of simple Python queries and the use of the Natural Language Processing (NLP) library *spaCy*.[3]

Two curious insights with respect to typical Instagram text captions shaped decisions on how text might provide useful ways of generating musical structure. Firstly, the overwhelming number of texts polled were positive in sentiment. This effectively mitigated the need for commonly applied analytical techniques designed to provide quantitative data on a text's semantic meaning. Secondly, there is often significant use of emoticons which cannot be readily analyzed with *spaCy* dictionaries nor other popular NLP libraries such as *TextBlobs* or *NLTK*. While emoticon types could certainly be used as a means of providing

[1] https://chromedriver.chromium.org/home
[2] https://www.crummy.com/software/BeautifulSoup
[3] https://spacy.io

structural organization in *96 Postcards*, when they ostensibly replace text, as in Figure 4a, they were not of particular interest, and accordingly such posts were discarded. An example of a caption that might be kept, despite its use of emoticons, is shown in Figure 4b.


Christmas {Cyprus Style} 🥔✉️🎄
.
.
.
.
.
📷: @thecypriottraveler
.
.
#Cyprus #ayianapa #winter #christmas #potato #sculpture #night #lights #scenic #adventure #positivevibes #peaceful #europe #travel #discover #community #follow #life #Explore #instadaily #instagood #fun #photooftheday #beautiful #love #happy #cute #Кипр #Kıbrıs #Κύπρος


Le Rocher du Basta et son point de vue parfait à la fois sur la Grande Plage 🏖 et sur le Port des Pêcheurs ⚓. L'occasion d'une balade bien agréable sous ce soleil ☀ de Novembre.

#biarritz #biarritztourisme #rocherdubasta #paysbasque #littoralbasque #oceanview #novembre #soleil

Figure 4. a) Sample Instagram caption on #Cyprus with no useful textual caption other than emoticons. Note in addition the relatively large number of hashtags which are also ignored in text analysis (upper), b) Sample Instagram caption on #Biarritz with an acceptable textual caption for inclusion (lower). In both examples, the Instagram user account name has been removed.

In *96 Postcards* the results of the textual analysis drive parameters of an immersive score which surrounds the performers, see Figure 5a. The notational schema builds on previous creative work by the author with 3D scores [12] and adopts a similar design aesthetic where a 3D construct of colored nodes, each of which denote the onset of a musical event, are connected by thin white lines which denote the duration of those events. In *96 Postcards*, this node/line construct is itself surrounded by twelve curved panels, or canvases in Unity3D nomenclature, partitioned into various subsections. Each of these partitions contain the Instagram image scraped from one of the 96 locations while the text caption accompanying one of these image postings is positioned above each panel, see Figure 5b.

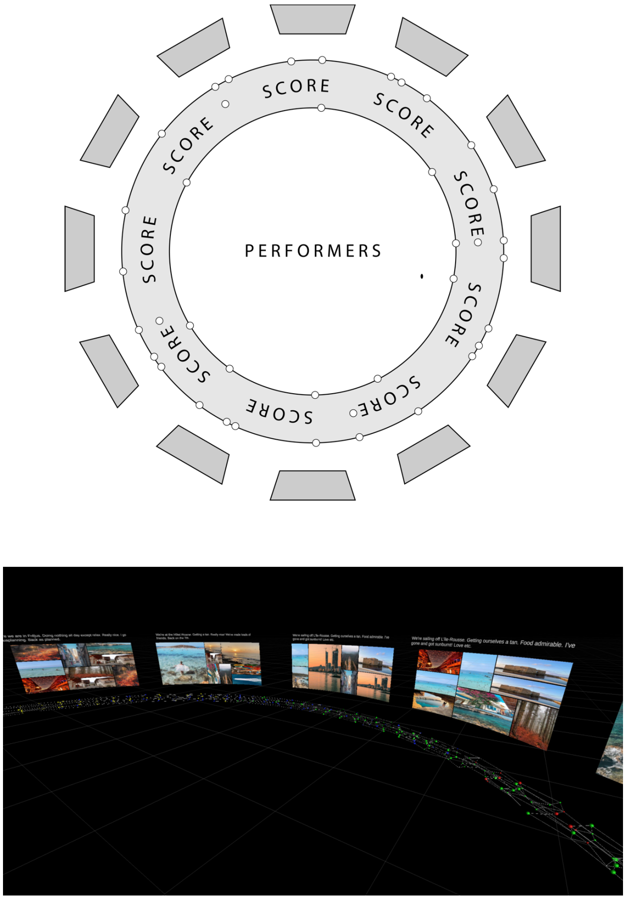


Figure 5. a) A bird's-eye view schematic of the immersive score for *96 Postcards* with performers positioned in the center of the VR scene and surrounded by the 3D score and image canvases (upper), b) a VR scene detail from Unity 3D's scene view (lower). Note the node/line construct of the immersive score below and slightly in front of the image canvases.

The mapping of linguistic data to features of the performance score was guided by various perceptual and musical considerations. In initial development, for example, the correlation of vowels within a text to pitch and harmonic content was thought to have organizational potential. But from a statistical perspective, the distribution of vowels within a series of text captions has negligible difference and hence any direct correlation of vowels to pitch or harmonic content was unlikely to result in a highly differentiated musical structure or at least a musical structure that might prove aesthetically satisfying. For this reason, it was decided that harmonic structure in *96 Postcards* would not be driven by NLP data but rather predefined. This had the additional benefit of ensuring a degree of continuity across performances.

In *96 Postcards* each node denotes the articulation of a pitch with the node's color indicating the specific pitch to be sustained along a line. During the performance, performers are instructed to freely read around the score from one node to another, musically exploring the myriad range of pathways presented as they traverse the full 360-degree node distribution. As in previous works, the performance per se thus becomes an actualization of the latent possibilities presented by the non-linear, open form notational schema.

As noted, the harmonic content in *96 Postcards* is predetermined and not driven by the NLP analysis. In the score, this effectively means that the color distribution of nodes falls within a predefined map with weighted probability distributions applied, see Figure 6.

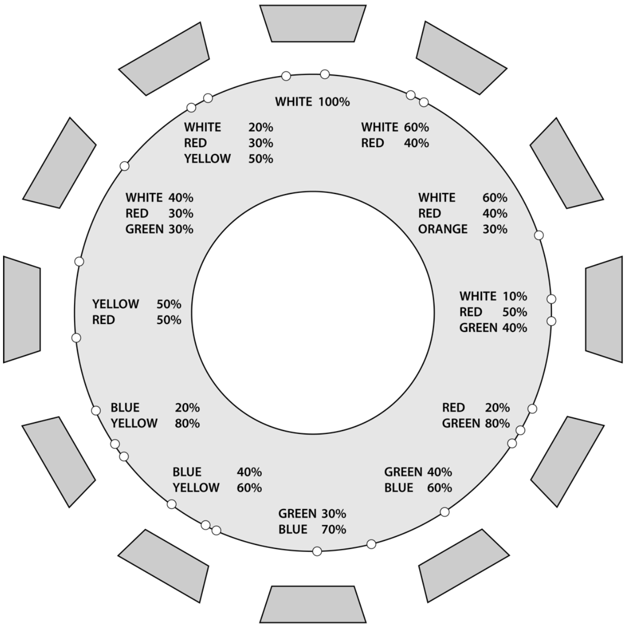


Figure 6. Harmonic mapping with probability weights around the performance score with node color mapped to pitch as follows – White (W) → Unpitched (whispered), Red (R) → C, Orange (O) → C quarter-tone sharp, Green (G) → G, Blue (B) → F, Yellow (Y) → B-flat. Pitches may be sung at any octave.

Unlike color distribution, which is not directly correlated to data returned by the text analysis, the spatial distribution of nodes in the score is driven by the spatial position of vowels within the Instagram text captions and other data returned by *spaCy* on the text's formal properties. Musically, vowel distribution therefore becomes a means of rhythmically striating a harmonic complex [13]. The spatial/temporal mapping is represented in the performance score by the thin white lines that connect nodes. These effectively function as a proportional notation with a line's spatial extension within the VR scene correlated to the temporal duration of a musical event. In *96 Postcards*, the mapping is prescribed at approximately 20 seconds across each canvas with smaller subdivisions resulting in proportionally shorter temporal divisions. Given that line is purely a rhythmic denotation, there was no need to ascribe additional graphic properties such as color or variety of thickness when developing the work, and hence each line is simply colored white with uniform width.

In the VR scene, the performers are surrounded by twelve panels partitioned into seven, eight, and nine subsections. Each partition is in turn filled with one of the 96 images collected in the Instagram scrape, see Figure 7. While the images themselves are not "interpreted" by the performers in any strict sense, they do provide an invaluable visual anchor as the performers traverse around the 360 degrees of the score. The text caption that accompanies each image post generates a discrete node/line construct that is positioned in front of its respective panel with the color and spatial distribution of nodes following the procedure previously outlined. As there are up to nine node/line constructs generated per panel, these are interleaved in order to be contained within the finite space at the front of each panel. While the interleaving of node/line constructs means that performers are not able to distinguish the individual constructs themselves, it does permit them to traverse across the visualizations, or across Instagram locations, as they navigate their way around the score. In this respect, as in previous work, the dèrive model of performance is a particularly apt metaphor [14, 15].

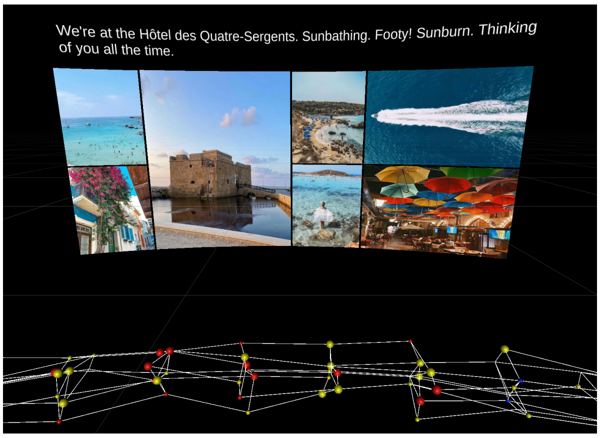


Figure 7. An illustration of a partitioned canvas from the performer's perspective. Note the text caption at top (from Perec's original text for illustrative purposes), the Instagram images in the canvas partitions, and a section of the immersive score's node and line construct slightly beneath.

The *dérive* model is further reinforced through the way in which the Instagram text captions function in the performance score. Above each of the twelve panels which surround the performers, one text caption is placed, see Figure 7. This caption provides a template for how node/line constructs directly beneath are to be enunciated. Considering compositional questions of mapping once again, the use of text was considered a far more natural means of conveying types of vocal articulation than any graphic lexicon especially given the ready source material from which node/line constructs were generated. In performance, the singers enunciate the node/line constructs through the successive vowels displayed in the Instagram text caption. For example, a caption such as "Le Rocher du

Basta et son point de vue parfait à la fois sur le Grande…", see Figure 4, would require performers to articulate and sustain pitches as represented in the node/line constructs along the vowels "e o e u a a e o o i e u e a a i à a o i u e a e…" etc. Extending the dèrive model of performance, each sounded text caption, sifted of plosives, fricatives, affricates and other consonants, becomes a filter through which all other visualized captions are sounded. These latent locations, given actuality only through a node/line construct are thus heard as ghost-like echoes through the exploratory dèrive of the performer's journey around the score.

4. VR DEVELOPMENT

4.1 Photon Engine SDK

The VR application of *96 Postcards* has been developed on the Unity3D platform for the Oculus Quest 2, a VR head-mounted display (HMD) released in 2020 which at of the time of writing is the world's highest selling VR headset.[4] Integral to the work is the use of the Photon Engine SDK, or PUN (Photon Unity Networking), which provides a framework for facilitating shared multiplayer experiences.[5]

Designed for the development of multiplayer games, PUN manages the networking back-end that allows players to be immersed in a common VR scene. The process through which this is achieved is straightforward and adopts a relatively standard protocol. Upon launching the *96 Postcards* application on the Quest 2, each performer is positioned in the center of the same VR scene through a simple series of automated synchronization calls managed by the PUN SDK. In *96 Postcards*, each of the performers passively observe the scene in which they are immersed. They do not interact with any of its constituent elements with hand controllers, nor do they interact with the avatars representing any of the other performers. Furthermore, each performer's individual movements are constrained to a limited set of movements which enable them to observe the score that surrounds them. The limited set of interactions afforded in this novel system architecture, minimizes the data synchronization calls made to the PUN network and ensures that the scene information between performers is not only consistent, but presented with minimal latency.

4.2 UX considerations

Unsurprisingly, there is a paucity of experiential data that can be drawn from to help guide VR design for musical performance. Amongst the more generally cited UX concerns, is the need to ensure that the space in which players are immersed is comfortable [16]. Thus, it is important to minimize visual effects that may induce motion sickness or other vestibular disorders, through careful attention to factors such as object movement, image resolution, field-of-view, and exposure time [17]. For *96 Postcards*, vestibular effects were not a pressing concern as the imagery surrounding the performers is entirely static, thus acting as an anchor to reduce any incumbent disorientation. Nevertheless, smooth headtracking where there is a one-to-one correspondence between the head movement speed and camera rotation within the VR scene, and the reduced number of PUN calls from the overall system architecture, further mitigated any negative vestibular effects.

A less often cited UX principle relates to text presentation. To ensure legibility, VR designers generally try to ensure that text is always static not just with respect to spatial location, but also appearance, for example not blinking and maintaining a stable color and style. In addition, the resolution of most HMDs requires font sizes to be larger than what UX designers may be typically accustomed. For example, the Quest 2 has a display resolution of 1832 x 1920 per eye whereas other popular VR HMDs have resolutions of 1440 x 1600 (HTC Vive Pro), 1280 x 1440 (Samsung Gear VR), and 960 x 1080 (Playstation VR). Even given the relatively high display resolution of the Quest 2, fonts need to be no smaller than 40 point to ensure minimal legibility from five feet (1.5m) away [18, 19]. As such, to include Instagram text captions within each canvas partition in the VR scene of *96 Postcards*, often means that some text could not be fully contained. Consequently, a decision was made to present text from one caption only on each canvas positioned directly above and offset from the images below to ensure maximum legibility, see previous Figure 6. While the score for *96 Postcards* does not employ common practice, stave-based notation, for composers who wish to use traditional schemas in VR scenes, issues of legibility and resolution are of even more pressing concern [20].

Unlike Augmented Reality (AR) technologies, the wearer of a VR HMD is fully immersed in a virtual world and not expected to interact with any physical objects in the real-world. To ensure wearer safety, Quest 2 users usually start their virtual experience by tracing a safe work area on the floor around them with a hand controller. Should the wearer approach the boundary of the safe space, a warning grid appears in their visual display. Full immersion in a VR scene naturally limits the types of instruments musicians can comfortably perform to those not requiring rapid eye-hand coordination or occupying large physical spaces. While the designers of the Quest 2 have recently permitted developers to access the pass-through camera of the HMD and project this image into the VR scene, the resolution is exceptionally poor at 1080x1200 and is also black-and-white. Both these factors make it difficult for the wearer to interact meaningfully with any objects in the real world while continuing to be present in VR space. These various constraints were instrumental in the pre-compositional decision to develop *96 Postcards* for an

[4] Meta/Facebook, the developers of the Quest 2, do not publish official sales figures although current estimates are in the order of more than 1.2 million units per quarter.

[5] https://www.photonengine.com/pun

ensemble of vocalists rather than other musical instruments.

Finally, while perhaps not a UX consideration from the perspective of the performer, the use of VR HMDs in musical performance raises a challenging UX question from the perspective of the audience. Unlike a traditional performance dynamic, the performers of *96 Postcards*, and indeed any VR work, are visually detached from the audience and other performers. Indeed, in VR space the traditional proscenium-based performance model breaks down. With the audience always situated outside the VR space in which the performers are immersed, the performance itself becomes a form of spectacle upon which the audience can only partially observe. How UX might be optimized for the audience is a challenging question. Whether they should be invited into this world, perhaps through a shared screen projection of the VR scene, or remain passively outside is ultimately, perhaps, an aesthetic question to be uniquely resolved for each VR experience.

5. FUTURE WORK

While my previous work in mixed reality has explored the use of AR technologies such as Microsoft's HoloLens as a means of displaying performance scores [21], *96 Postcards* is my first work with a graphic score to be specifically designed for a VR environment. While the technical affordances of AR and VR systems have proven especially attractive, it is their ability to offer new forms of creative expression which has provided the richer field of possibilities.

As a tool for immersive visualizations of large data sets, whether they function as graphically notated performance scores, visual works of art, or more scientifically focused efforts to enrich understanding, VR offers particularly exciting possibilities. In his recently published text *Cultural Analytics* [22], Lev Manovich explores many of these and particularly how computational analysis of images, whether gathered from Instagram, print media, or other sources, can help facilitate cultural analysis. The creation of new forms of aesthetic expression from analyses of other forms of aesthetic expression would itself seem to offer exciting possibilities for innovative, creative enquiry. For the author's creative practice, this is perhaps the most fertile area for future exploration, aestheticizing multiplayer agency and refracting it back into the social media platforms upon which it reflects and is drawn. Further work might also seek to obtain measures for optimizing the UX experience for the audience, and perhaps finding ways in which their agency might itself provide a means of transforming visualized sets of data.

Acknowledgments

Much of my recent research into musical applications of virtual and augmented reality has been made possible thanks to a Churchill Fellowship awarded in late 2019 and completed in 2021. I would like to thank the Churchill Trust for this opportunity and for their generous support.

Thanks also to Dr. John Taylor from The MARCS Institute for Brain, Behaviour & Development at Western Sydney University for his many helpful suggestions on an earlier draft of this paper.

6. REFERENCES

[1] G. Perec, "Deux Cent Quarante-Trois Cartes Postales en Coulers Veritables," *Le Fou Parle*, 1978.

[2] T. Bonch-Osmolovskaya, "Combinatorial Greetings from Georges Perec," in *Bridges 2018 Conference Proceedings*, Stockholm, 2018. Available: https://archive.bridgesmathart.org/2018/bridges2018-253.pdf

[3] G. Perec, *Species of Spaces and Other Pieces*. John Sturrock (Ed. And Trans.), Penguin, 1999.

[4] W. Motte, *Oulipo: A Primer of Potential Literature*. Dalkey Archive Press, 1998.

[5] A. James, *Constraining Chance: Georges Perec and the Oulipo*. Northwestern University Press, 2009.

[6] P. Terry (Ed.), *The Penguin Book of Oulipo*. Penguin, 2021.

[7] H. Philipsen, "Constraints in Film Making Processes Offer an Exercise to the Imagination," *International Journal of Media, Technology and Lifelong Learning*, vol. 5, no. 1, 2009.

[8] K. Waschneck, *Screening Oulipo – From Potential Literature to Potential Film*, PhD Thesis, University of Essex, 2017. Available: http://repository.essex.ac.uk/22413/1/Screening%20OULIPO.pdf

[9] K. Terzidis, *Algorithmic Architecture*. Routledge, 2016.

[10] M. A. Boden and E. A. Edmonds, "What is Generative Art?" *Digital Creativity*, vol. 20, no. 1-2, 2009, pp. 21-46.

[11] D. Kim-Boyle, "The Twittering Machine," in *Proceedings of the TENOR2022 International Conference on Technologies for Music Notation and Representation*, Marseille, 2022.

[12] D. Kim-Boyle, "The 3-D Score," in *Proceedings of the TENOR2017 International Conference on Technologies for Music Notation and Representation*, A Coruña, 2017.

[13] P. Boulez, *Boulez on Music Today*. S. Bradshaw and R. R. Bennett (Trans.), Faber and Faber, 1975.

[14] G. Debord, "Theory of the Dérive," Les Lèvres Nues No. 9 (Paris, November 1956). Reprinted in *Internationale Situationniste* No. 2 (Paris, December 1958). K. Knabb (Trans.).

[15] K. O'Rourke, *Walking and Mapping: Artists as Cartographers*. MIT Press, 2013.

[16] C. Hillmann, *UX for XR: User Experience Design and Strategies for Immersive Technologies*. Apress, 2021.

[17] D. Saredakis, A. Szpak, B. Birckhead, H. A. D. Keage, A. Rizzo, & T. Loetscher, "Factors Associated With Virtual Reality Sickness in Head-Mounted Displays: A Systematic Review and Meta-Analysis," *Frontiers in Human Neuroscience*, vol. 14, 2020. Available: https://www.frontiersin.org/article/10.3389/fnhum.2020.00096

[18] V. Kurbatov, "10 Rules of Using Fonts in Virtual Reality." [Online] Available: http://vovakurbatov.com/articles/10-rules-of-using-fonts-in-virtual-reality

[19] E. Van der Burg, C. N. L. Olivers, J. Cass, "Evolving the Keys to Visual Crowding," *Journal of Experimental Psychology: Human Perception and Performance*, 43:4, 2017, pp. 690-699. Available: https://doi.org/10.1037/xhp0000337

[20] L. Vickery, "The Limitations of Representing Sound and Notation on Screen," *Organised Sound*, vol. 19, no. 3, 2014, pp. 215-227.

[21] D. Kim-Boyle, "3D Notations and the Immersive Score," *Leonardo Music Journal*, Vol. 29, 2019.

[22] L. Manovich, *Cultural Analytics*. MIT Press, 2021.

LINEAR: A MULTI-DEVICE AUGMENTED REALITY ENVIRONMENT FOR INTERACTIVE NOTATION AND MUSIC IMPROVISATION

Giovanni Santini
Hong Kong Baptist University
Hong Kong
giovesan@hkbu.edu.hk

ABSTRACT

LINEAR (Live-generated Interface and Notation Environment in Augmented Reality) is an environment for the generation of real-time 3D interactive graphic notation. The environment is suitable for ensemble improvisative performances featuring acoustic instruments, live-electronics and two Augmented Reality (AR) performers. One AR performer uses an iPhone for drawing virtual trajectories in the space, rendered as a sequence of Virtual Objects (VOs) aligned along the trajectory. VOs trigger samples upon virtual collisions with the iPhone. They are also used as a form of graphic notation for instrumentalists/vocalists: the screen of the iPhone is mirrored to a projector. The second AR performer uses a headset and can use VR controllers to design trajectories used for the spatialization of each audio source in a 3D audio setup. The headset AR performer can use virtual spheres (one per instrument) to control the position of each sound source (one per instrument). The sound of every acoustic instrument is processed live. The mixing of processing effects are controlled by a laptop player. The system has been repeatedly tested during a two-semesters long workshop. The system was also used for two online concerts. Beyond demonstrating the technical and musical viability of LINEAR, the workshop also gave the chance to record student's response to the system. Although the sample size is quite small (four students completed the survey), the answers show encouraging results in terms of engagement and interest. Future work should be conducted to further enhance the user experience and more clearly assess LINEAR's usability and effectiveness as an innovative system for improvisation and musical performance.

1. INTRODUCTION

A new wave of interest for immersive technologies is recently arising, with the hype related to the *Metaverse* and the attempts to actualise it. The Metaverse is a concept about a virtual, distributed, interoperable world that can run in parallel with the real world and includes complex features (from human interaction to trade). It is typically associated with technologies like blockchain and machine learning, and even more strongly with Extended Realities (XR) technologies, i.e. Virtual and Augmented Reality (VR/AR). VR places the user in a completely virtual world, with Virtual Objects (VOs). The only relation to the physical world are the movements of the user detected via motion capture. AR merges virtual objects with the real world and allows interaction with them, possibly enriching the experience with spatial understanding capabilities: the ability to analyze the physical connotation of the surrounding space and have VOs behave accordingly (e.g., by rendering the shadows on the floor). XR applications could become increasingly important for arts making, in term of new ways of expressions and new market segments.

The main interest for the author is to find some creative space opened up by such technological developments. The hybrid reality created by the combination of virtual and physical objects, actors, and environments can reveal uncharted territories for exploration and creation. In designing LINEAR, the aim was to create a complex system, enabling numerous ways of interaction between performers, musical materials and the virtual world. The author wanted to preserve an exploratory attitude and some flexibility of use, without tying the whole design to one single expected outcome. The challenge consisted in doing so while embedding capabilities that are exclusive of AR technology, seen as a new medium for musical expression.

2. BACKGROUND

In the past few years, there has been some research related to musical notation in AR. Mostly, researches exploited the use of the temporal dimension (one of the typical traits of AR) [1, 2]. In some researches, the 3-dimensional spatial nature of AR was also exploited [3, 4, 5]. In most of the cases, researchers adopted graphic notation solutions, as opposed to the traditional descriptive notation. Graphic notation is an umbrella term that refers to numerous different contexts and aesthetics. We could define it as that form of notation that uses graphical solutions that are not part of the Common Western Notation (CWN) lexicon. Graphics can be used either in addition to traditional notation or replace it. An example of a mixed use of CWN and graphic notation could be found in action scores, such as Lachenmann's *Pression* (1969), while an example of a purely graphic scores is Haubenstock-Ramatiì's *Konstellationen* (1971). Graphic notation has also been used in recent technology-based solutions such as real-time scores for animated notation [6], 3D scores [7] and VR scores [8].

Graphic notation has been widely adopted in Augmented

Reality applications for music education, music composition and music performance.

2.1 AR notation in music education

In music education, AR notation seems to be generally conceived as a subsidiary tool to assist traditional music learning. Typically, it replaces (or aids) the descriptive notation of traditional scores with a prescriptive 3D interactive notation that indicates hands or fingers positions requested at a certain time, e.g. [9, 10]. The notational solutions (although not resembling themselves the gestural behavior of the performer) have a clear connection with the physical, spatial displacement of performance actions, rather than with the expected result. An example of this principle is the piano roll. With small variations across different studies, a piano roll is a system that makes use of virtual colored blocks (coming towards the keyboard) to indicate the keys to press at a specific time [11, 12, 13, 14] and Figure 1. Additional indications for dynamics or wrong notes can be delivered through the use of different colors or User Interface (UI) elements.

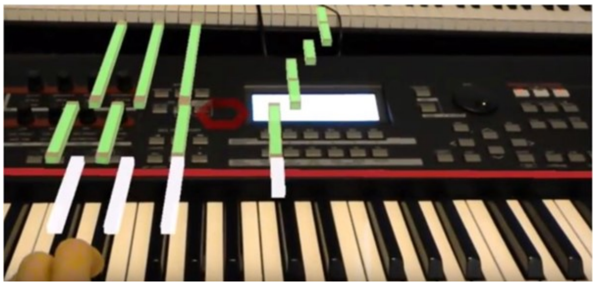


Figure 1. One example of piano roll [15].

2.2 AR for music performance or composition

A different case can be made for AR applications that are not designed for educational purposes but rather as a tool for artistic use (performance or composition or both). Relatively few experiments have been conducted so far, presumably due to the technical challenges and the absence of an already established corpus of background work to help researchers. We can currently identify three directions (not necessarily mutually exclusive):

- AR to solve practical performance/rehearsal issue (performers can visualize a 2D score anywhere without a physical score).
- Immersive scores.
- Interactive notation.

An example of 2D notation in Augmented Reality has been developed in order to facilitate discussion and feedback among musicians. The score (bi-dimensional, written in CWN) is visualized in AR and can be moved at different distances and locked to the point of view of the user(s). This allows for a great flexibility during rehearsal time and favours discussion among musicians, as long as they all have a headset [2].

Smartvox [1] is a browser-based network-distributed environment for synchronized real-time scores. In its recent developments, its use was extended to networked head-mounted displays with Augmented Reality capabilities. This way, a performer can visualize a 2D (animated) score as a semitransparent layer superimposed to the environment. This solution allows for a much freer and comfortable user experience (especially in choir situations) both during rehearsal and performance. This is especially true for animated notation which requires a constant attention from the performer.

In the opinion of the author, the two examples above do not actually represent a case of Augmented Reality notation, but rather a translation of 2D notation in an AR context in order to solve a comfortability issue. Although some interactivity and some use of the time dimension can be identified, these examples do not develop notation using the three spatial dimensions. Essentially, they allow the visualization of a screen with very flexible spatial capabilities. In other terms, the page is in AR but not the score.

In [3], David Kim-Boyle outlines some properties of graphic notation in an AR context. The notation gains an architectural dimension, which becomes essential for the interpretation of the score. Graphic and spatial connotations are then fully exploited in an immersive score.

"In *64x4x4* [...] the physical engagement with the score becomes an essential means of uncovering its various potentialities. The pathways through the score, uniquely instantiated for each performance, may only be discovered when the performers physically navigate the space in which the score is displayed" [3].

Amy Brandon's work has also been widely developed around the use of AR notation. In her works, notation also assumes the function of interface: interactable elements of the graphic notation are also used to trigger samples (*Hidden Motive*, 2018, *Augmented Percussion*, 2019):

"I could add functionalities to the graphic score - make it interactive or animate it. In performance terms, the musician would be able to grab elements of the score, and would be able to trigger audio files in the process" [16]. This combination has been defined notation-interface hybrid [17].

[18] presents a similar concept but more oriented towards sound generation than notation. Pitch values are generated according to the movement in space and then passed to a synthesizer. The sound is panned in 3D according to position in space.

GesturAR [4] is an experimental application that allows to notate performance gestures in the real space. A hand tracking device is used to detect the palm position. When the *record* mode is activated, at each frame, the positions are stored in a trajectory and a line is rendered according to each point of the trajectory. The trajectories can be stored, combined and played back. The coordinates are in real world and associated to an origin point provided by a tracking device which can be positioned on the instrument to be played. The resulting notation has been called embodied interactive notation: the act of notation coincides with the notated act.

The panorama presents a lively, yet relatively small amount of research focusing on AR notation and its connection with VOs, space and physical world. However, the continuous improvement of development frameworks and the increasing generalized interest in XRs seems to slowly foster the growth of the field.

3. DESCRIPTION OF LINEAR

LINEAR is an environment for real-time music improvisation, without a fixed number of performers. The bare minimum number of performers is 4: one iPhone performer, one AR headset performer, one laptop player and one instrumentalist/vocalist. Concerts were performed with a total of 7 performers (4 acoustic performers). The system has been designed in order to be swiftly adapted to a different number of performers and instruments. Each instrument player is required to be miked with at least one microphone for real-time sound processing.

The environment is composed of two different applications, one for iPhone, the other one for an Augmented Reality headset. The two applications are independent from each other but connected to an Ableton Live/M4L project via OSC protocol. The applications have been developed with the framework Unity 3D. The HTC Vive Pro with a ZED Mini VR camera has been used for the headset app.

3.1 The iPhone app

The iPhone app allows the performer to draw trajectories in space by using physical gestures while holding the device. Such trajectories are visualized as a sequence of Virtual Objects (VOs) placed along the the device' movement. By using three different buttons, the iPhone performer can choose between three different types of VOs associated to three different colors and particle effects (graphic effects composed by up to millions of instances of a same fundamental object, a particle). Such effects have a different "energy" and size according to the different speeds of the user's gesture. Various parameters of the particle effects regulate those levels of energy: number, speed, life time and size of each particle. The speed of the device in the moment of the creation of a VO (averaged over 5 frames) is mapped to those parameters in order to deliver different tiers of "excitement".

The visual effects are rendered on the screen of the device, altogether with the real environment. Each VO is connected to a sample stored in a sample library loaded in the Ableton Live project. In the moment of the creation of the VO, the sample is played. It is also played anytime the performer moves the device onto the position where the VO is instantiated.

The screen of the iPhone is mirrored to a projector. Therefore, the instrument/vocal players are able to read the graphic notation. The precise way they are asked to do so is explained later in the article. In order to provide orchestration and behavioral constraints, the screen of the iPhone is divided in as many parts as there are instrumental/vocal performers. In Figure 2, the screen is divided in four parts as there are three instruments and one singer. The names are indicated around the center of the screen, one for each quadrant.

3.2 The AR headset app

The application for AR headset is also based on the creation of virtual trajectories associated to different VOs, each of them corresponding to one of the performers' audio channels (including the iPhone performer). Each object has a different color which is inherited by the corresponding trajectory. The app allows four functions: *select*, *draw*, *play* and *play all*, connected to different inputs on the VR controllers. The *draw* function allows one to draw trajectories in space, visualized as continuous lines. Those trajectories correspond to points of coordinates in space that are communicated to the Ableton Live project via OSC to control a sound spatialization module. The coordinates of those trajectories correspond to the positions of virtual sound sources (one per trajectory). The *select* function enables the choice of different VOs, each of them controlling the coordinate in space of a different sound source: the sound processing (and sometimes amplification) of acoustic instruments and the iPhone player's samples. When a VO is selected, only its trajectory is played (therefore, only the linked sound source is moved in space). The *play* function plays back the trajectory already created for the selected VO. *Play all* plays all trajectories together.

The point of view of the headset performer is mirrored to a projector. The other performers are not asked to "read" the trajectories the same way they do with the iPhone screen. However, when a single VO is selected, the name of the corresponding instrument is shown on the screen and becomes an orchestration indication: the selected object/instrument is a soloist and therefore the rest of the ensemble should adjust their dynamics in order to let the soloist be in the foreground.

Figure 2. A view of the two screens in a concert setting.

3.3 The Ableton Live project

The Abletone live project includes processing modules with an effect chain composed of dynamic EQ, spectral delay, octaver, distortion and a multi-buffer granulator. All the effects are custom-made and created with Max4Live. Each processing chain is conveyed onto a mono signal bus routed to the spatialization module, based on IRCAM's *Spat* (the *spat* track in Figure 3), which positions the different sound sources in a 3D audio panorama. This effect is controlled via OSC by the AR headset app. Although the amount of processing on each mono signal would sometime require more channels for a better result, the author preferred to avoid moving stereo signals in order to better guarantee spatial separation between sound sources. The sound spatialization has been tested in rehearsals and concerts in two different spaces, respectively equipped with a 3D 24.2 setup and with a 7.1 setup. The algorithm used was 3D Vector-based Amplitude Panning (VBAP3D).

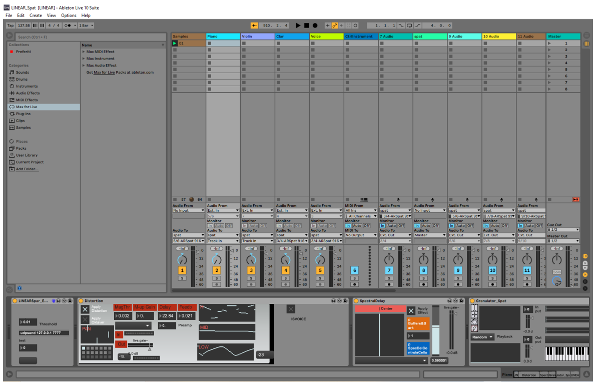


Figure 3. The Ableton Live project for LINEAR.

On each channel, each effect can be separately bypassed and balanced, according to the needs of the different instruments, microphones etc. Presets are available to instantly control numerous combinations of effects and loudness on each channel: e.g. full chain applied to clarinet and violin, only granulator to the voice, only amplification to the piano. Presets and loudness balance are controlled by the laptop player.

4. SCORES AND PERFORMANCE NOTES

The trajectories drawn in space by the iPhone performer form an embodied interactive score [1]: the trajectories represent spatial points of reference, precisely linked to sounds, and are derived from bodily actions. Repeating those actions will reproduce the same sounds. The score is a consequence of the act it is meant to notate.

The mirrored iPhone screen is also read as a graphic score by the non-AR performers. In the example, the screen is divided in four parts, one for each instrumental/vocal performer, who are asked to read only their quadrant and to not produce sound when there is no VO in their slot.

4.1 How to read the graphic score

Each one of the three VO types is connected to a different sample library. The generic sound quality of the samples could be labelled as:

- Long, voice-like low pitches.
- High, articulated, fast attack pitches.
- Noise, breath sound quality, either long or short attack.

The non-AR performers are asked to imitate those sounds by using extended techniques on their own instrument. For example, a violin could imitate as follows:

- 4th string detuned one fifth lower, with continuous movement of the bow between bridge and fingerboard, and alternating the pressure of the right hand while performing microtonal glissandos with the left hand, at most one tone above the open string.
- Fast pizzicatos with very high fingering (close to the bridge) on 1st and 2nd string.
- Muted strings with left-right and up-down bow movements, either slow or fast and either short or long.

The set of techniques required is bigger. Typically, at least 3 techniques for each sound are required to each performer.

Performers need to read left-to-write and imitate the spatial disposition of VOs (e.g., few objects separated = sound alternated with silence, high spatial density = no silence). Ideally, their quadrant should be read as a 5 second loop. Also the “energy” of the effect needs to be taken into account (e.g., steady effect = sustained sound with flat profile, highly cynetic effect = fast articulation on a moving profile).

4.2 Solo instrument information

The point of view of the AR headset is mirrored to a projector. The result is not read as a score by the musicians, but it indicates the name of the instrument corresponding to the virtual source moved in that moment (e.g., voice in Figure 2). The instrument indicated is meant to be treated as a soloist, while the other musicians (if playing) are required to stay in the background. When *play all* is activated, all players should consider themselves as soloists, while when no sound source is selected, all of them should think about staying in the background. These rules might create absurd combinations: sometimes, an instrument could be selected as solo, but no VO is shown on that instrument’s quadrant, and therefore the instrument is not allowed to play. In that case, the other players should stay in the background of an instrument that is not playing. Future iterations of the environment might solve this issue. However, this paradox can also stimulate the seek for creative solutions: how can sounds be in the background of silence?

4.3 How to structure a performance

The iPhone player is the conductor/real-time composer. They create the real-time score that needs to be read by the other performers and also decide the point of view on the VOs, therefore what each perfomer sees in their quadrant. To some extent, the iPhone player decides what the performers will or will not do. The challenge is to create an overall development that delivers some structural interest

[1] “The notation is created as a direct consequence of an embodied act (detected through sensors) and is a 4D representation in space and time of the original gesture, in the form of a trajectory or some other kind of spatial marking” [17].

over time. As a reference for the development of the performance, the author created a performance outline (Figure 4), subdivided in “bars” of approximately 10 seconds. The colored lines indicate the different types of VOs. Rectilinear lines notate a low level of activity, while broken ones do the opposite. This indication translates into the speed of the gesture that creates the VOs. Dynamics are used to indicate the density (FF = screen filled with VOs, PP = one or two Vos in the screen). Indications are also used for audio processing effects (G = granulator, SPD = spectral delay, OKT = octavter, DIST = distortion). The performance outline does not provide other details, such as which instruments should play, which gesture should be used to create VOs and what the point of view should be. All these details, impacting the real-time graphic score, are left to the sensitivity and reactivity of the iPhone player.

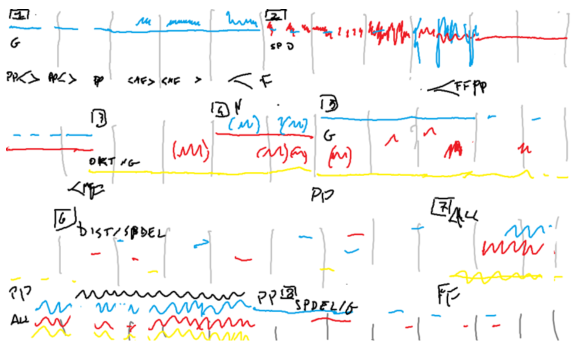


Figure 4. The performance outline for one concert. Different colors indicate different types of VOs, therefore different timbral references for the instruments (and different sets of extended techniques). The shapes of the lines indicate the level of excitement and dynamics describe density.

5. EVALUATION AND DISCUSSION

The current version of LINEAR has been developed through a year of continuous experimentation in a New Music class. Two concerts were made, one mid-term, the other one at the end of the academic year. A small evaluation has been conducted. Out of the 6 students taking part to the final concert, 4 replied to the form. The aim of this informal evaluation was essentially to obtain some feedback and reactions about the notational concept as well as the perceived effectiveness of LINEAR as an innovative tool for musical performance. In order to do so, participants were asked to fill an online form divided in two parts. In the first part, five questions with a 1-5 Likert scale were asked (see Table 1). In the second part, questions with free answers were asked. The forms were submitted through an online anonymous document.

Four free questions were made in the second part of the survey.

- What do you think about the use of notation in LINEAR?
- How would you compare an improvisation using LINEAR with other ways of improvising?
- Use three adjectives to describe your experience with LINEAR.
- Any additional thought/feedback?

The evaluation shows a certain appreciation of LINEAR. In particular, participants seemed to find the visuals engaging and the system quite innovative. Interesting suggestions were made on how to improve the effectiveness and interactivity of the notatio. For example, one response to the first question was: “A set of clearly defined and notated gestures would have been useful; sort of like a taxonomy of possible sounds and their relationship to the virtual objects”. Although such relationships were actually presented during the rehearsals, the use of the system would probably benefit from a very structured detailing of playing techniques. While the author approached it quite experimentally (hands-on trial), future iterations should probably include clear indications established in advance.

Strongly disagree (1), Strongly agree (5)	**1**	**2**	**3**	**4**	**5**
The use of spatial movement of the iPhone player is inspiring/effective	-	-	1	1	2
The use of notation helps to find interesting/new solutions	-	-	2	1	1
3D virtual objects contributed to make the experience more engaging	-	-	1	-	3
The sound spatialization is clear/well-working	-	-	1	1	2
You would want to work more with LINEAR	-	-	1	1	2

Table 1. Likert-scale questions and responses.

Another comment stated: “Some special effects/sound can be created when certain movement of phone/VR appeared (e.g. draw a circle and some wind sound is responded)”. Including machine learning and visual recognition algorithms to increase the interactivity of the system is a solution that might be experimented in future iterations. This would allow to add pre-composed, fixed gestural material to the performance. The multi-sensory nature of LINEAR was also pointed out, in one of the reponses to the second open quesion of the form: “Improvisation using LINEAR evokes multiple-sense of reactions visually and auditory”. The environment is found innovative: in response to the free question number 3, 2 adjectives were repeated twice: *innovative* and *futuristic*. Also, one free comment: “It seems to me that this type of work and technology is very promising, looking forward to future developments”.

The result of the performance needs to be considered as part of the evaluation too. The author finds the outcome promising, while challenges and needs of improvement should be pointed out. The overall system has some expressive capabilities, and the reactivity of performers to different graphic changes is sometimes convincing. However, there is some lack of precision in the relationship between score and sound, for what concerns VO-extended technique, enegy-articulation, solo-tutti relationships. The continuous unpredictability of the score is certainly a reason. One of the rehearsals strategies was to practice on steady points of view, rehearsing the techniques until the sound result was adhering to the picture. Part of the reason of the lack of precision could maybe be found in the long time span between rehearsals (typically one month). However, the environment itself can result hard to decipher, especially for the distribution of information across two screens. The ecosystem of mutually listening and reacting performers, with a human-controlled real-time score is fascinating and promising. Future adjustments need to be made in order to render more clearly the different information, thus increasing the efficacy of the notation and the possibility of control.

6. CONCLUSIONS AND FUTURE WORK

LINEAR is designed to be an articulated performance environment that requires continuous practice and exploration to be perfected, similarly to any musical instrument. It is composed of two AR applications for different devices (iPhone and AR headset) and one AbletoneLive/M4L sessions. The application for iPhone is responsible for playing sounds (via Abletone Live) from virtual sources and generating a live AR graphic score. The AR headset app is used to 3D pan virtual sources by controlling an effect based on IRCAM's *Spat*. The live-electronics processing is controlled by a laptop player. For non-AR performers, there are two sources of information to read from. Mainly, the mirrored iPhone screen (divided in four quadrants), containing the AR graphic score. Performance information is contained in a different quadrant for each performer. VO type (color), energy and density are linked to expected performance outcome: extended techniques used, articulation, dynamics and/or density of the sound texture. They also need to read the mirrored AR headset screen to know which instrument is soloist in that moment. For the iPhone player, the performance outline is the primary source of information while creating the live AR score. The AR score is also a form of embodied interactive notation to follow. The headset performer does not need to follow any score or indication. The laptop player follows the lead of the iPhone player in activating the presets.

The environment presents an initial learning curve, as it includes different groups of performers which follow different sets of indication (the non-AR performers, the iPhone performer, the AR headset performer). Additionally, the performance heavily relies on the performers' knowledge of extended techniques. However, LINEAR proved to be viable for rehearsal and performance after a training process.

Future improvements of the system shall include bigger sample libraries with enhanced processing capabilities. Using concatenative synthesis could be a solution to create more lively and differentiated sound results. A more structured definition of performance techniques for each instrument should be considered. Currently, the two AR applications run separately and just talk to Ableton Live. Future enhancements shall allow interoperability between the systems and have VOs on one device impacting rendering on the other device. The clarity of on-screen indications could be enhanced, for example by finding a way to condense all the information needed by the interpreters on one screen. It would also be interesting to find solutions for replacing the laptop player. For example, commands to start a new preset could be designed for the iPhone or for the AR headset app UI.

Recording of one concert

The file is a recording over zoom (the concert was in online format) and mostly the audio is heavily clipped.

Acknowledgments

The developments described in this article have been possible thanks to the collaboration with the *Contemporary Music Workshop* class, held by Dr. Camilo Mendez at Hong Kong Baptist University.

7. REFERENCES

[1] J. Bell and B. Carey, "Animated notation, score distribution and VR-AR environments for spectral mimetic transfer in music composition," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, Melbourne, Jul 2019, pp. 7 – 14.

[2] Z. Liu, M. Adcock, and H. Gardner, "An evaluation of augmented reality music notation," in *Proceedings - VRCAI 2019: 17th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2019, pp. 1 – 2.

[3] D. Kim-Boyle, "3D notations and the immersive score," *Leonardo Music Journal*, pp. 39 – 41, 2019.

[4] G. Santini, "Composition as an embodied act: A framework for the gesture-based creation of augmented reality action scores," in *Proceedings of the Sound and Music Computing Conferences*, 2020, pp. 357 – 363.

[5] A. Simion, A. Iftene, and D. Gîfu, "An Augmented Reality Piano Learning Tool," in *International Conference on Human-Computer Interaction, RoCHI*, 2021, pp. 16 – 17.

[6] C. Hope and L. Vickery, "The DECIBEL Scoreplayer - A Digital Tool for Reading Graphic Notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, M. Press, Ed., 2015, pp. 58 – 69.

[7] J. Han, "An architectural spatial approach to 3d spatial drum notation," *Tenor 2017 International Conference on Technologies for Music Notation and Representation*, pp. 39 – 49, 2017.

[8] R. Hamilton, "Trois machins de la grâce aimante: A virtual reality string quartet," in *Proceedings of the 2019 International Computer Music Conference,*

ICMC-NYCEMF 2019 - International Computer Music Conference New York City Electroacoustic Music Festival, 2019.

[9] J. Martin-Gutierrez, M. S. Del Rio Guerra, V. Lopez-Chao, R. H. S. Gastelum, and J. F. V. Bojórquez, "Augmented reality to facilitate learning of the acoustic guitar," *Applied Sciences (Switzerland)*, pp. 1 – 7, 2020.

[10] Y. Zhang, S. Liu, L. Tao, C. Yu, Y. Shi, and Y. Xu, "ChinAR: Facilitating Chinese Guqin learning through interactive projected augmentation," in *ACM Interna-tional Conference Proceeding Series*, 2015, pp. 23 –32.

[11] M. Cai, M. A. Amrizal, T. Abe, and T. Suganuma, "Design of an ar-based system for group piano learning," in *Adjunct Proceedings of the 2019 IEEE International Symposium on Mixed and Augmented Reality, ISMAR-Adjunct 2019*, 2019, pp. 20 – 21.

[12] D. Hackl and C. Anthes, "HoloKeys - An augmented reality application for learning the piano," in *CEUR Workshop Proceedings*, 2017, pp. 140 – 144.

[13] W. Molloy, E. Huang, and B. C. Wunsche, "Mixed reality piano tutor: A gamified piano practice environment," in *ICEIC 2019 - International Conference on Electronics, Information, and Communication*, 2019.

[14] L. Rigby, B. C. Wünsche, and A. Shaw, "PiARno-An Augmented Reality Piano Tutor," in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, 2020, pp. 481 – 491.

[15] E. Strasnick, A. Chambers, L. Jiang, and T. Xiaonan, "Pianolens: An Augmented Reality Piano Interface." [Online]. Available: https://www.youtube.com/watch?v=5TExa2L1rOM

[16] A. Brandon, "Hidden Motive: on the development of interactive graphic scores for AR and VR," *The Association of Canadian Women Composers (ACWC)*, pp. 19 – 28, 2018.

[17] G. Santini, "Augmented Piano in Augmented Reality," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2020, pp. 411– 415.

[18] K. Yoo and E. Schwelling, "Spatially accurate generative music with AR drawing," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, 2019, pp. 1 – 2.

MUSICKING DEEP REINFORCEMENT LEARNING

Hugo Scurto
EUR ArTeC / Université Paris 8 / EnsadLab
Paris, France
hugo.scurto@ensad.fr

ABSTRACT

In this paper, I relate an auto-reflexive analysis of my practice of designing and musicking deep reinforcement learning. Based on technical description of the Co-Explorer, a deep reinforcement learning agent designed to support sonic exploration through positive or negative human feedback, I discuss how deep reinforcement learning can be seen as a form of sonic comprovisational agent, which enables musicians to compose a parameter sound space, then to engage in embodied improvisation by guiding the agent through sound space using feedback. I then relate on my own musicking experiments led with the Co-Explorer, which resulted to the creation of the *ægo* music performance, and build on these to sketch a music representation for deep reinforcement learning, highlighting its original aesthetics, as well as its ontological shifts between performer and agent, and epistemological tensions with engineering-oriented representations. Rather than discrediting the latters, my wish is to create space for practice-based approaches to machine learning in a way that is complementary to engineering-oriented approaches, while contributing to further music representations and discourses on artificial intelligence.

1. INTRODUCTION

Reinforcement learning defines a computational framework for the interaction between a learning agent and its environment [1]. The framework provides a basis for agents that learn an optimal behaviour within their environment by taking actions in it, then receiving positive or negative feedback from it, as a reward or punishment signal. Recent advances in deep learning enabled reinforcement learning to be applied to high-dimensional spaces, through the so-called deep reinforcement learning framework [2]. Such a framework actively contributed to the growing field of artificial intelligence, with application domains ranging from robotics and finance to healthcare and science [3].

Deep reinforcement learning was recently explored in the domain of music. Kotecha used deep reinforcement learning to generate symbolic polyphonic music [4]. Karbasi *et al.* explored deep reinforcement learning to create rhythms for a collective of interactive robots [5]. Ramoneda *et al.* applied deep reinforcement learning to learn optimal piano fingerings based on simulated piano performances [6]. Yet, all these works privileged an engineering-oriented approach to deep reinforcement learning, using it as a computational model for existing symbolic music representations. In addition, they did not include musicians in the research and design of these generative models, leaving both analytic and performative aspects of music practice aside.

As a musician, designer and researcher, I was interested in adopting a design-oriented approach to deep reinforcement learning. I was especially interested in exploring novel forms of musicking where a deep reinforcement learning agent would learn interactively from a musician, that is, by receiving positive or negative feedback from them. I was expecting that such a creative process could in turn lead to new designs and representations for deep reinforcement learning that originate from music practice as much as from engineering. I was notably inspired by previous works from Bevilacqua *et al.*, who pioneered such interactive approaches to machine learning for gestural control of sound [7], and by Fiebrink *et al.*, who highlighted musical attributes of machine learning by leading in-depth studies of the creative process of musicians creating gesture-sound mappings with machine learning [8].

In this paper, I relate an auto-reflexive analysis of my practice of designing and musicking deep reinforcement learning. In Section 2, I describe the Co-Explorer, a deep reinforcement learning agent that supports sonic exploration based on positive or negative human feedback, designed in collaboration with sound designers. In Section 3, I discuss how deep reinforcement learning may be seen as a form of sonic comprovisational agent, enabling musicians to compose a parameter sound space, then to engage in embodied improvisation by guiding the agent through sound space using feedback. In Section 4, I relate on my own musicking experiments with the Co-Explorer, which resulted in the creation of *ægo*, a music performance for one human improviser and one learning machine, presented at this year's TENOR music track. I end by discussing in Section 5 how musicking deep reinforcement learning helped me sketch a music representation for this computational framework, highlighting the epistemological, ontological and aesthetic shifts produced by musicking compared to its standard, engineering-oriented applications. Rather than discrediting the latters, my wish is to create space for practice-based approaches to machine learning in a way that complements engineering-oriented approaches, with the hope that it will contribute to further music representations and discourses on artificial intelligence.

2. CO-EXPLORER

In this section, I describe the Co-Explorer, a deep reinforcement leaning agent designed to support sonic exploration based on positive or negative feedback provided in real-time by a musician. The Co-Explorer was developed as part of my doctoral thesis, which sought to approach machine learning as design material in the context of new interfaces for musical expression [9]. Specifically, we adopted a human-centred design approach to deep reinforcement learning, involving sound designers in diverse steps of our design process, and studying their creative processes with our software agent [10]. The next sections describe technical foundations of deep reinforcement learning, and more specifically, the exploration method and interaction modalities that we developed within the Co-Explorer [1]. I refer the reader to my previous papers for technical details and qualitative evaluation of implementation.

2.1 Interactive Deep Reinforcement Learning

Deep reinforcement learning is a generic computational framework for the interaction between a learning agent and its environment. Our first design step thus consisted in defining a model of the environment and the agent that could be adapted to the use case of sonic exploration.

We opted for an elementary model of the environment, consisting of a parameter space of arbitrary dimension (*e.g.,* a synthesis space). Technically speaking, let $\mathcal{S} = \{S\}$ denote the state space constituted by all possible parameter configurations $S = (s_1, ..., s_n)$ reachable by the agent, with n being the number of parameters, and $s_i \in [s_{min}, s_{max}]$ being the value of the i^{th} parameter living in some bounded numerical range. Let $\mathcal{A}(S) = \{A\}$ denote the corresponding action space as moving up or down one of the n parameters by one step a_i, except when the selected parameter equals one boundary value. The resulting agent would thus iteratively explore the parameter space while producing continuous sound synthesis variations.

Crucially, we assumed that a musician observes the state-action trajectories taken by the agent in real-time, and interactively provides positive of negative feedback, or reward R, to the agent. As such, the agent would progressively learn a mapping between states and actions, leveraging deep learning to tackle learning and generalisation in high-dimensional parameter spaces. The resulting trained model can be used as a representation of a musician's subjective preferences toward a parameter space.

2.2 Exploration Method

In addition to learning musician's preferences, reinforcement learning agents have a second aim, which is to maximise feedback received from the musician. As such, they may help musicians find the best state-action in the parameter space as they explore it.

To do so, agents rely on exploration methods that enable them to find optimal state-action trajectories in their environment. Intuitively speaking, an agent has to balance exploitation of their computational knowledge (*e.g.,* taking the best actions as defined by previous feedback to maximise future feedback) with exploration of their environment (*e.g.,* taking sub-optimal actions in terms of previous feedback, possibly leading to better actions in the future).

For the Co-Explorer, we developed a novel exploration method that builds on an intrinsic motivation technique, which pushes the agent to "explore what surprises it". Specifically, it has the agent direct its exploratory actions toward uncharted parts of the space, rather than simply making random moves, as in most reinforcement learning [1].

Thus, deep reinforcement learning agents may have a dual role: on the one hand, their learned model can be used as a representation of a musician's subjective preferences toward a parameter space; on the other hand, their exploration behaviour can be used to foster the creative process of a musician toward some parameter space.

[1] `https://github.com/Ircam-RnD/coexplorer`

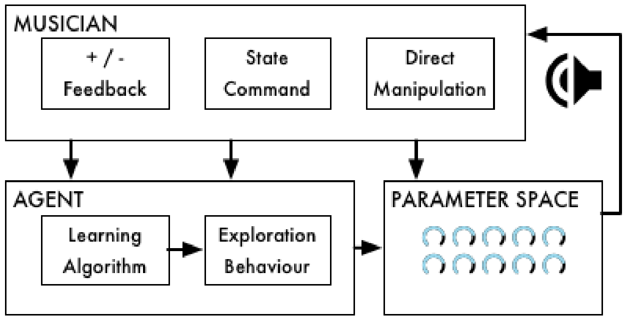


Figure 1. Co-Explorer workflow.

2.3 Interaction Modalities

Our next step consisted in designing interactions with deep reinforcement learning to let musicians experiment with both its learning and exploration abilities. We collaborated with sound designers to iteratively design and implement these interactions within the Co-Explorer (see Figure 1).

The first interaction modality is positive or negative feedback. We distinguished guiding feedback, which enables to provide feedback toward actions taken by the agent, and zone feedback, which enables to provide feedback toward states reached by the agent. While each type of feedback relies on a different implementation, they both consist of a scalar with continuous positive or negative value.

The second interaction modality is state commands. State commands enable to control the agent's trajectory more directly, that is, without relying on feedback. A first state command is changing zone, which enables to command the agent to make an abrupt jump to an unexplored parameter state. Another state command is start/stop autonomous exploration mode. In autonomous exploration mode, the agent takes actions in the parameter space at a regular time interval, whether the musician provides feedback (thus learning in real-time), or not (thus relying on its learned model and exploration behaviour).

The third and last interaction modality is direct parameter manipulation. It enables to explore the parameter space by hand, as in most sound synthesis workflows. Additionally, it enables to choose a given parameter state from which the agent would start its autonomous exploration.

3. DEEP REINFORCEMENT LEARNING AS SONIC COMPROVISATIONAL AGENT

In this section, I discuss how deep reinforcement learning may be seen as a form of sonic comprovisational agent, enabling musicians to compose a parameter sound space, then to engage in embodied improvisation by guiding the agent through sound space using positive or negative feedback. I detail possible strategies to compose such parametric sound spaces, as well as possible configurations for musical improvisation through positive or negative feedback, which let me argue for deep reinforcement learning as a form of technology for comprovisation [11].

3.1 Composing Sound Spaces

Composing a sound space consists in defining the timbral features and diversity of sounds to be produced by the learning agent. Sound spaces are context-independent, and may be fixed before interacting with the agent. As we will see in Section 4.1, they may be successively explored by the agent to create dramaturgy along a musical piece.

Technically speaking, composing sound spaces consist in linking the parameters of the model to parameters of a sound synthesis engine. Concretely, one may first choose one given synthesis engine, then curate n synthesis parameters from it, and set the numerical bounds within which the agent would lead exploration. As the environment's model is generic, one may connect the Co-Explorer to any parameter sound synthesis engines, including commercial VSTs, physically-inspired sound synthesis, descriptor-based sound synthesis, or custom Max/MSP patches. The resulting sound morphologies would continuously evolve across time, as the agent would iteratively take actions on parameters and thus reach new parameter states.

While the Co-Explorer was initially designed to explore sound spaces, the genericity of its environment model makes it theoretically applicable to other music representations. For example, the Co-Explorer was used to explore rhythmic structures, by approaching states as discrete rhythmic patterns of size n, and agent actions as activations or deactivations of beats within the pattern [12]. Other musical applications could lie in the creation a chord parameter space, and have the agent learning to modify note combinations.

3.2 Improvising Through Feedback

Beyond sonic exploration for sound design, I argue that deep reinforcement learning opens new approaches for musical improvisation due to its relying on positive or negative feedback. Below I detail how feedback may be used as a contingent element of a performance, supporting real-time instructions toward sound, symbolic communication with sound, and embodied responses toward sound, all contributing differently to the agent's learning.

3.2.1 Feedback as Instructions Toward Sound

A first musical use of feedback follows that which is technically defined by deep reinforcement learning: namely, enabling performers to provide instructions toward sound to guide the agent's learning and exploration of the space.

As a generic, scalar value, feedback may be directed toward various dimensions of sound. For example, positive or negative feedback may be used to evaluate timbral attributes of sound, so that the agent learns a model of timbre from its parameter environment. Or, feedback may be used to communicate subjective preferences toward sound, so that the agent learns a model of the composer's or performer's tastes toward sound.

In both cases, feedback-based instructions toward sound may be fixed before the performance by the composer. In this case, improvisation would be led by the agent, essentially through its exploration behaviour, while the performer would communicate accurate feedback to teach the agent to reach some goal sound fixed by the composer.

Alternatively, such instructions toward sound could be opted for in real-time by the performer. In this case, improvisation would be essentially led by the performer as they would guide agent exploration in real-time through feedback. Specifically, the performer may use feedback to convey spontaneous subjective preferences toward sound, or rely on some sonic scenario, decided before, or emerging from, improvisation, to guide agent exploration. In this case, the reaching of a goal sound may both depend on accurate feedback provided by the performer, as well as on agent learning and exploration of the parameter space.

3.2.2 Feedback as Symbolic Communication With Sound

Rather than sound-oriented instructions, feedback may be reappropriated by the performer to communicate with sound at a symbolic level. For example, a performer may use positive or negative feedback to express personal semantics or imagery toward sound, rather than to evaluate timbral features of sound. In this case, the performer might start to imagine that they are controlling sound production, even if the agent may not be able to properly learn such a high-level representation. Alternatively, a performer may consciously communicate contradictory feedback as a way to hijack the agent's learning, and thus, its trajectory in the sound space. In this case, the performer may have no pre-conceived scenario toward improvisation, except that of discovering unexpected sounds, due to the agent's struggling in interpreting the performer's feedback.

3.2.3 Feedback as Embodied Response To Sound

In addition to instructions or symbolic communication, feedback may be produced by the performer as an embodied response to sound generated by the agent. For example, a performer may produce feedback involuntarily, as errors toward instructions provided by a composer, or as an emotional response toward timbral or symbolic features of sound. Or, a performer may produce feedback to expressively accompany sounds generated by the agent, in a way similar to ancillary gestures produced by musicians with their instruments [13]. In the latter case, the performer may approach feedback as an abstract thread that connects them with the agent, thus creating space for expressive improvisation with sound, in a way similar to dance, where movements that accompany music can lead performers to feel that they have control over sound production [14].

3.3 Comprovising with Deep Reinforcement Learning

I believe that the combination of context-independent with contingent elements makes deep reinforcement learning a new technology for computer-based comprovisation. In its current formalisation, deep reinforcement learning highlights sound listening as a main feature, where it be in the composition of sound spaces, or during improvisation with the agent. Its second feature is the enabling of musical improvisation through a high-level communication channel, that is, positive or negative feedback, which can be used as either an indirect control modality toward sound generation (in the case of instructions and symbolic communication), or as a direct engagement modality with sound (in the case of symbolic communication and embodied responses). While recent, the framework was already explored by other musicians, specifically, to compose and improvise with musical gestures [15].

4. MUSICKING EXPERIMENTS

In this section, I relate my own musicking experiments led with deep reinforcement learning, made in collaboration with composer-researcher Axel Chemla–Romeu-Santos between 2019 and today. They resulted in the creation of *ægo*, a music performance for one human improviser and one learning machine. The piece was performed one time in 2019 [16]; we produced a reworking in 2022, which we will premiere at this year's TENOR music track.

ægo started by the wish to experience comprovisation with the Co-Explorer, possibly leading to the discovery of alternative music representations for deep reinforcement learning. We adopted a practice-based approach to the Co-Explorer, that I propose to describe as musicking [17], since it essentially relied on listening to, and performing with, the sounds and music produced by deep reinforcement learning, without assuming any pre-established musical form. The next sections details the compositional, improvisational, and comprovisational experiments led through *ægo*. I refer the reader to our previous paper for aesthetic and technical details on the performance itself [16].

4.1 Composing Latent Sound Spaces

A first aspect of musicking deep reinforcement learning lied in composing parameter sound spaces that the agent will navigate through. For *ægo*, we opted for latent sound spaces, that is, sound spaces created by generative deep learning, another machine learning framework that enables to produce new data that resembles existing data [18]. Latent sound spaces have interesting musical features for comprovisation. Specifically, their parameters are not necessarily interpretable as technical synthesis parameters, such as frequency, amplitude, or modulation. Rather, they should reflect perceptual variations of timbre of sound datasets used for learning. Thus, improvising in a latent sound space should generate continuous timbre variations, interpolating between recognisable timbres contained in the training dataset, while also generating ambiguous timbral artifacts typical of generative deep learning [19].

For *ægo*, we opted for two latent sound spaces built over two training datasets: synthesis sounds and acoustic instrument recordings. We stress that we consciously chose these latent spaces in terms of the training datasets they relied on to be created. Yet, we underline that we could not exactly define, nor control, the types of sounds contained in these latent spaces, due to the intrinsic generativity of deep learning [18]. As such, we opted for an experimental approach to composing sound spaces, first crafting generative models through their sound dataset, then curating the latent dimensions to be explored by Co-Explorer. This process was highly recursive, as composing latent sound spaces required improvising through gestural feedback to fully grasp their musical attributes.

4.2 Improvising Through Gestural Feedback

Indeed, a second aspect of musicking deep reinforcement learning consisted in improvising through feedback with the agent. As a performer, I opted to develop a gestural controller to communicate positive and negative feedback. Specifically, I used inertial measurements units, placed on top of velcro rings, to measure my hands' orientations. I added both angular values and scaled the resulting numerical scalar so that it goes from -1 to 1. My wish was that such a bodily interface would allow for more intuitive and creative musicking with deep reinforcement learning.

In early experiments, I was able to discover the bodily vocabulary enabled by this gestural controller to communicate feedback. The most elementary and illustrative gestures consisted in turning my hands front to communicate positive feedback, and turning them back to communicate negative feedback, by only pivoting wrists. Through improvisation, I discovered other gestures to be explored to provide instructions toward sound, as well as to symbolically communicate with sound. Asymmetric hand postures, for example, enabled to obtain neutral feedback, since the sum of the two angular values would be zero. Yet, the resulting gesture would not be neutral, and would produce expectation and tension for both the performer and the audience. I also explored somatics-based gestures, focusing on internal bodily sensations as I was listening to sound, and producing free-form aerial gestures as embodied response to sounds, resulting in varying feedback values.

All along our experiments, I witnessed myself entering in a state of heightened listening toward sound. Specifically, I observed myself oscillating between two approaches: one that was performative, where I attempted to grasp control over sound by producing precise instructions or symbolic communications, and one that was meditative, where I carefully listened to sound as if it existed by itself, detached from my very own influence, even if my body responding to it in spite of me. Both cases almost had me forgetting about the agent's learning abilities for the benefit of discovering novel sound morphologies, at times witnessing my light influence on it. In short, feedback-based improvisation pushed me to consider both optimal and non-optimal behaviours of deep reinforcement learning as relevant for music performance, while simultaneously contributing to a feeling of spiritual identification with music [16].

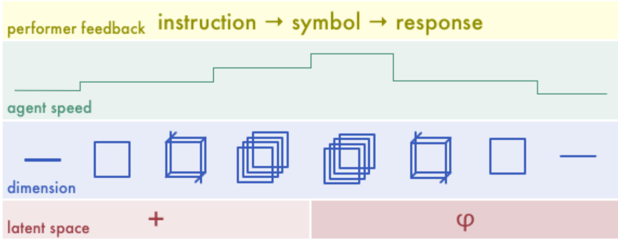


Figure 2. *ægo* musical structure (2022 reworking).

4.3 Comprovising a Musical Structure

Based on our musicking experiments, we sought to compose a musical structure for *ægo* (see Figure 2).

We chose to start the performance with the latent sound space built over synthesised sounds, since its timbral variations were less chaotic than those produced by the instrumental latent space. We also wrote series of latent dimensions for each space, going from one to eight for the first latent space, then eight to one for the second. As such, widening or narrowing timbral richness in sound spaces let us create dramaturgy across performance for both the performer and the audience.

For our premiere of *ægo* in 2019, the composer was responsible for choosing the moments where the agent would switch latent sound spaces. In our reworking, we renounced to this idea, and had the agent autonomously change zone within a sound space based on its exploration behaviour. This decision was shown to create space for improvisation for the performer, since they would discover the new sound space at time of performance only, thus dealing with even more indeterminacy from the agent during performance.

Similarly, for our premiere of *ægo* in 2019, we had set the time interval between agent's action at a fixed value, which resulted in the agent navigating sound spaces at a slow speed, thus producing a continuous drone sound. In our reworking, we wrote this time interval to produce diverse spectromorphologies along performance, alternating between continuous drone sounds through slow speeds, and glitchy sounds through higher speeds. This let us compose musical tension for both the performer and the audience.

Last but not least, we directed the performance so that the performer progressively relinquishes communication of accurate feedback to the agent, that is, going from instructions to symbolic communication and eventually to embodied response to sound. On the one hand, this choice aimed at improving audience comprehension of the performance, as they would first witness the agent being optimally guided by the performer, then progressively observe the blurring threads of influence between the agent and the performer. On the other hand, this choice enabled us to critically engage with deep generative learning, as embodied responses to sound will lead the agent to produce non-optimal behaviours within the sound spaces. Displaying such indeterminate behaviours as one defining musical attribute of deep reinforcement learning do not conform to established engineering-oriented applications; yet, from our perspective, it speaks of their material engagement with musicians and the world.

5. FROM COMPUTATIONAL FRAMEWORK TO MUSIC REPRESENTATION

In this section, I sketch contours of a music representation for deep reinforcement learning, which emerged from my musicking experiments with the Co-Explorer. I detail the epistemological tensions over learning models, the ontological shifts between performer and agent, and the aesthetics of feedback-based improvisation, produced by such a music representation for deep reinforcement learning.

5.1 Epistemological Tensions over Learning Models

As described in Sections 1 and 2, deep reinforcement learning defines a computational framework for agents that learn by interacting with their environment. Typical applications of deep reinforcement learning seek to learn an optimal behaviour in relation to the goal of a task. Engineering-oriented approaches thus seek to optimise an agent's learning by constructing some synthetic reward function that will yield the best results in terms of learning [2]. Thus, every other feedback functions can be seen as sub-optimal, or even incorrect, from this engineering perspective [1].

The music-oriented approach to deep reinforcement learning suggests that imperfect human feedback functions for engineering may in turn yield rich forms of improvisation for music research and practice. In fact, I argue that "optimal behaviour" may be a dynamic and emergent attribute of musicking and improvisation, as opposed to the static and pre-existing definition of engineering sciences. For example, one could argue that indeterminacy, as a musical feature of deep reinforcement learning, is what contributes the most to musicking, beyond agent learning or exploration behaviour. Yet, indeterminacy remains a variable that needs minimising in engineering-oriented approaches to deep reinforcement learning. Thus, goals of music and engineer practices may sometimes be opposed.

I believe that such epistemological tensions should be taken seriously by music researchers and practitioners, especially in the current growing applications of artificial intelligence to music, which often reinforce static representations of music through symbolic modelling of existing languages [4]. I suggest that musicking can be one such practice-based approach to discover material attributes of machine learning and illuminate their emerging properties. Rather than discrediting engineering-oriented approaches, I see this highlighting as an opportunity for interdisciplinary collaboration, enabling to iterate the design and implementation of learning models that are entangled with music.

5.2 Ontological Shifts Between Performer and Agent

As described in Section 4, musicking deep reinforcement learning enabled me to enter in a state of heightened listening toward sound. This heightened listening had me witness my oscillation between two different postures toward sound and the agent. On the one hand, I would aim at instrumental control over both sound and the agent, using feedback as both sonic instructions and symbolic communications. On the other hand, I would witness the existence of sound beyond myself and the agent itself, as feedback would only help me believe that I control sound.

I argue that this oscillatory phenomenon reveals an ontological displacement of the notions of performer and agent toward sound. In fact, this displacement may drastically differ from other computational frameworks for music improvisation based on machine learning [20, 21, 22]. The latters often rely on anthropomorphic representations of sound and music, such as MIDI signals, and inject these in the design of the agents. Simultaneously, the performer may also rely on their joint technical and embodied knowledge of music to produce sound with their instrument and interact with the agent. As such, the role of the performer remains clearly defined as that of a musician. The agent, on the other hand, can be described as intelligent, or even as creative, as it builds on the same anthropomorphic music representation than that of the performer, while simultaneously being equipped with a greater musical agency compared to other software for music composition.

In deep reinforcement learning, however, no anthropomorphic representation of sound or music are injected in the design of the agent. Simultaneously, the role of the performer slightly moves away from that of a musician, since they do not rely on their instrumental knowledge to interact with the learning agent, nor do they actually produce sound directly. As such, the role of the performer oscillates between that of a musician and that of a listener, while also creating space for observation of the fluid boundaries that operate between these two roles, thus fostering spiritual identification with the produced sound. In parallel, I suggest that the role of the agent may progressively move away from that of an anthropomorphically-creative agent, to that of a non-human form of intelligence that produces music by conveying temporal form to sound. I believe that such an analysis should be deepened from a musicological perspective to produce alternative discourses toward artificial intelligence: rather than seeking to imitate or replace musicians, machine learning may enable to produce rich forms of musicking that foster human creativity while heightening their listening to their environments.

5.3 Aesthetics of Feedback-based Improvisation

As described in Section 3, deep reinforcement learning enables to engage in sonic improvisation by only relying on positive or negative feedback. The resulting interactions include instructions toward sound, symbolic communication with sound, as well as embodied response to sound. In a sense, they move away from standard instrumental techniques to music, as they privilege material attributes of sound over languages usually employed to describe it; indirect influence on sound over precise control and mastery of it; but also and crucially, identification with music over actual sound production. As a result, the values encapsulated in feedback-based music improvisation may differ from values of improvisation found in more traditional written music. In fact, they may lead to ethical encounters with certain communities of music practice and research, for example debating the quality or "truthfulness" of the produced music, or criticising the entertaining aspect or "seriousness" of the designed interactions with sound.

I argue on the contrary that feedback-based music improvisation create novel embodied interactions with sound that produce as "true" music as any other approaches to music composition or performance, and as "serious" interactions with sound than other physical or computer technology for music. If required, I would situate the aesthetics produced by deep reinforcement learning in line with the experimental music movement, in the sense that they push boundaries of existing genres, definitions, or disciplines of music, through their ontological shifts of performer and agent, epistemological tensions over learning models, but also and essentially, through their concrete approach to sound, and their reliance on free improvisation and indeterminacy processes to produce music [23]. In this sense, I suggest that debates toward aesthetics produced by musicking deep reinforcement learning may not fundamentally differ from those opposing conventional and nonconformist music practices with computer technology, long before the current trend for artificial intelligence.

Furthermore, I believe that feedback-based music improvisation summon "serious" social, cultural, bodily and spiritual phenomena related to embodied interaction with sound. Feedback shares similarities with gesture, in the sense that both may be used to translate sound using embodied knowledge and somaesthetic appreciation. In fact, feedback may be closely linked to biosignals, such as muscle tension or heart beats, in the sense that both may sometimes reflect involuntary responses of a musician toward sound. In this sense, feedback as an interaction modality for sound may be shared among diverse communities of people, be they musicians or non-musicians. Going further, I would suggest that the belief of controlling sound, as fostered by feedback, could be of interest for music practices that engage with people with disabilities [24]. Rather than just a basis for entertainment, identification with music have been at the heart of musicking for centuries: I suggest that it may be actively summoned within machine learning design and engineering to empower people toward both music practice and artificial intelligence technology.

6. CONCLUSION

In this paper, I have reported an auto-reflexive analysis of my practice of designing and musicking deep reinforcement learning. I have described the computational framework of deep reinforcement learning, along with the Co-Explorer, an agent designed to support sonic exploration through positive or negative feedback. I have discussed how deep reinforcement learning may be seen as a form of sonic comprovisational agent, enabling musicians to compose a sound space, then to engage in embodied improvisation by guiding the agent through sound space using feedback. I have reported musicking experiments made with the Co-Explorer, which led to the creation of *ægo*, a music performance for one human improviser and one learning machine. This enabled me to sketch a music representation for deep reinforcement learning, attempting to make its epistemological, ontological, and aesthetic aspects explicit for practitioners and researchers in music and machine learning. I hope that the present work will contribute to further music representations and discourses on artificial intelligence within the TENOR community.

Acknowledgments

I would like to thank Axel Chemla–Romeu-Santos for collaborating in the composition and performance of *ægo*, as well as Bavo Van Kerrebroeck, Baptiste Caramiaux and Frédéric Bevilacqua for collaborating in the design and implementation of the Co-Explorer, and Raphaël Imbert for pushing me to write on my such musicking experiments with deep reinforcement learning.

7. REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[3] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *arXiv preprint arXiv:1811.12560*, 2018.

[4] N. Kotecha, "Bach2bach: Generating music using a deep reinforcement learning approach," *arXiv preprint arXiv:1812.01060*, 2018.

[5] S. M. Karbasi, H. S. Haug, M.-K. Kvalsund, M. J. Krzyzaniak, and J. Torresen, "A generative model for creating musical rhythms with deep reinforcement learning," in *2nd Conference on AI Music Creativity*, Online, 2021.

[6] P. Ramoneda, M. Miron, and X. Serra, "Piano fingering with reinforcement learning," *arXiv preprint arXiv:2111.08009*, 2021.

[7] F. Bevilacqua, R. Müller, and N. Schnell, "Mnm: a max/msp mapping toolbox," in *Proceedings of the 5th International Conference on New Interfaces for Musical Expression (NIME)*, Vancouver, CA, 2005, pp. 85–88.

[8] R. Fiebrink, D. Trueman, N. C. Britt, M. Nagai, K. Kaczmarek, M. Early, M. Daniel, A. Hege, and P. R. Cook, "Toward understanding human-computer interaction in composing the instrument," in *ICMC*. New York, USA: Citeseer, 2010.

[9] H. Scurto, "Designing with machine learning for interactive music dispositifs," Ph.D. dissertation, Sorbonne université, 2019.

[10] H. Scurto, B. V. Kerrebroeck, B. Caramiaux, and F. Bevilacqua, "Designing deep reinforcement learning for human parameter exploration," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 28, no. 1, pp. 1–35, 2021.

[11] S. Bhagwati, "Notational perspective and comprovisation," *Sound & Score. Essays on Sound, Score and Notation*, pp. 165–177, 2013.

[12] H. Scurto, B. Caramiaux, and F. Bevilacqua, "Prototyping machine learning through diffractive art practice," in *Designing Interactive Systems Conference 2021*, 2021, pp. 2013–2025.

[13] R. I. Godøy and M. Leman, *Musical gestures: Sound, movement, and meaning*. Routledge, 2010.

[14] M. Leman, *Embodied music cognition and mediation technology*. MIT press, 2007.

[15] F. G. Visi and A. Tanaka, "Interactive machine learning of musical gesture," in *Handbook of Artificial Intelligence for Music*. Springer, 2021, pp. 771–798.

[16] H. Scurto, A. Chemla *et al.*, "Machine learning for computer music multidisciplinary research: A practical case study," in *Perception, Representations, Image, Sound, Music. 14th International Symposium, CMMR 2019, Marseille, France, October 14–18, 2019, Revised Selected Papers*, vol. 12631. Springer, 2021, pp. 665–680.

[17] C. Small, *Musicking: The meanings of performing and listening*. Wesleyan University Press, 1998.

[18] P. Esling, A. Chemla-Romeu-Santos, and A. Bitton, "Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces." in *ISMIR*, Paris, Fr, 2018, pp. 175–181.

[19] D. Ghisi, "Music across music: towards a corpus-based, interactive computer-aided composition," Ph.D. dissertation, Paris 6, 2017.

[20] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov, "Omax brothers: a dynamic topology of agents for improvization learning," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, Santa Barbara, USA, 2006, pp. 125–132.

[21] J. Nika, M. Chemillier, and G. Assayag, "Improtek: introducing scenarios into human-computer music improvisation," *Computers in Entertainment (CIE)*, vol. 14, no. 2, pp. 1–27, 2017.

[22] J. Borg, "Somax 2: A real-time framework for human-machine improvisation," Internal Report-Aalborg University Copenhagen, Tech. Rep., 2019.

[23] M. Nyman, *Experimental music: Cage and beyond*. Cambridge University Press, 1999, vol. 9.

[24] S. T. Parke-Wolfe, H. Scurto, and R. Fiebrink, "Sound control: Supporting custom musical interface design for children with disabilities," in *Proceedings of the 19th International Conference on New Interfaces for Musical Expression (NIME 2019)*, Porto Alegre, BR, 2019.

MAKING GRAPHICAL SCORES MORE ACCESSIBLE: A CASE STUDY

Christina Karpodini, Tychonas Michailidis
DMT Lab
School of Computing and Digital Technology
Birmingham City University
Birmingham, UK
{christina.karpodini, tychonas.michailidis}@bcu.ac.uk

ABSTRACT

This paper explores new ways of making graphical scores more accessible for visually impaired users. Existing assistive technologies demonstrate a gap in providing accessible tools for composing and performing contemporary music with non-traditional western notation. The two case studies presented, *Blocks Sound* and *Logothetis Sound* examine the interactive relationships and affordances through tactile interaction and how this interaction can influence their experience and understanding of both graphic scores and interactive composition of users. We present the process and limitations and propose the use of haptic technology and tangible experience for making contemporary graphic scores more accessible and inclusive.

1. INTRODUCTION

According to the World Health Organisation (WHO), more than two billion people around the world are suffering from a near or distance vision impairment [1] . Technological development facilitates more and more the needs of visual impairment in their place of work, how they communicate as well as around their creative aspirations. The digitisation of the braille system into displays and editors and the use of synthesised voices that can read digital text have been a massive step in the right direction. However, other types of recent media, such as gaming, digital art, virtual and augmented reality and music notation, that rely heavily on visual feedback, have been mostly inaccessible to these users. This paper explores the possibilities of making graphic scores, often a visual art piece that combines visual aesthetics with music representation, more accessible to people with visual impairment using contemporary means of technology. The case studies presented aim to bring closer the gap of accessibility in graphic scores by examining the work of composer Anestis Logothetis (1921-1994), a pioneer in graphical scoring and the use of interactive techniques and advanced computational methods.

[1] https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

2. EXISTING TECHNOLOGY

Over the last 50 years, advances in technology have enabled new assistive technologies making digital information available to the visually impaired [1]. Text focus devices such as braille display printers, editors, and screen readers are now broadly available, and accessible to those in need [2]. However, looking at the artistic freedom of these devices, they have diminished capabilities due to interference with the required functionality for the visually impaired user. The focus is mainly on processing text or images by displaying them in a different format or through text-to-speech. There is limited opportunity to explore and experience graphic scores, something visually focused.

This paper focused on the visual aspect of the aforementioned graphic scores that bring together sound and visuals. Projects related to music and blindness focus on exploring traditional western musical semiography. Screen reader software like *Blindows* and *Jaws* braille display as well as printed sheet music in braille format, have been used extensively by those interested in learning and performing music [3]. However, despite development in technology, hardware and software, there is still a large part of compositions within 20th century that uses non-traditional methods and consequently make part of music history inaccessible. Contemporary composers describe their music in terms of sound quality at a microscopic level with tools no longer adequate by the traditional western music notation system.

Studies in Human-Computer Interaction (HCI) enable us to narrow down the problem of accessible tools for appreciating or making contemporary music. HCI studies related to music production help us to understand the relationships of the existing assisting technologies and the need for more innovative one. Haenselmann et. al [3] suggests that the evolution of electronic music developed in an uncomfortable way for people with visual impairments from analogue with direct tactile interactions, like holding and striking the strings of the guitar or playing the weighted keys of the piano, to digital devices with the majority of feedback information shifted towards the screen or into other visual means of representation.

Paul Dourish, referenced in Tanaka [4], comments on the same topic but from the prism of HCI evolution. Dourish suggests four ways of embodied–computer interaction: *electrical*, *symbolic*, *textual*, and *graphical*. In this context, the interaction shifts from being accessible to non-

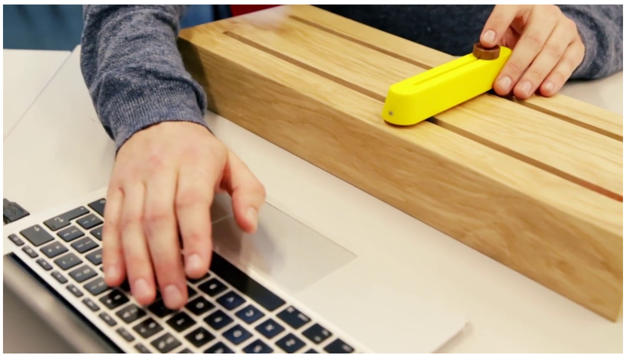


Figure 1. Haptic Wave.

accessible because of the focus on graphical/ skeuomorphic representation. Parallelisation of Dourish's four ways of interaction through the expansion of music technology are as follow:

- Electrical = Analog synthesizers
- Symbolic = Audio programming
- Textual = Live coding
- Graphical = Digital Audio Workstations

During the last 20 years, researchers have made great efforts to make contemporary and mostly digital music creation accessible. Such projects include, but not limited to, HapticWave [5] (see figure 1), ActivePaD [6], CuSE [3], Wedelmusic VIP Module [7], Soundsculpt [8] (see figure2). These academic projects often fade out rabidly without the support to become sustainable solutions. The variety of approaches in these projects, including tools for audio editing, processing and recording to a haptic musical database, is evidence of the complexity of sound as raw material and the challenges faced in transforming it into other non-visual forms.

Fine arts institutions like galleries and museums, including artists have made a significant effort to make their visual art accessible. Although these efforts provide a unique experience of the art piece, there are limitations to what the attendee can perceive and understand by just touching the artwork. Very often, a multimodal approach is needed, with brail text or audio description to accompany the piece [9]. Therefore, the development of haptic feedback technology has become an essential feature in moving research forward and creating a more holistic experience for art pieces while providing tools for creation.

Unlike other visual art pieces, graphic scores cannot be accompanied by an audio description as they are sonic art pieces. Thus, the case studies presented here have focused on exploring ways of creating Logothetis's graphical score, accessible to visually impaired users, interpreted and analysed within the concept of interactive composition. The following case studies suggest how we utilise this idea. Graphic scores are an intriguing subject for multi-modal design research that combines accessibility, interactivity, tactility, and sound.

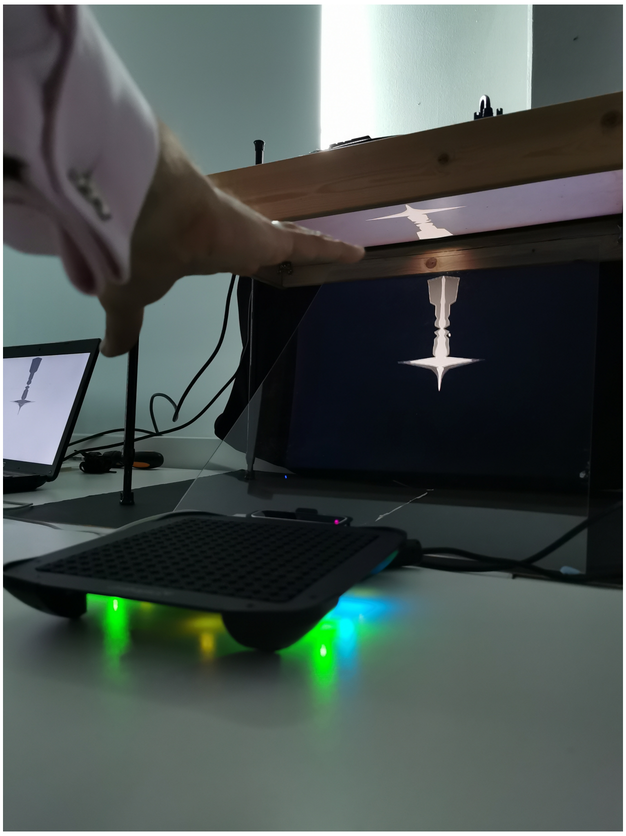


Figure 2. Soundsculpt.

3. THE SENSE OF TOUCH

Haptics has been extensively applied to systems for enhancing the experience of the user as well as to improve the music performance of particular tasks. The *Haptic Chair*, for example, by Nanayakkara et al. [10] provides a cross-modal audio-visual feedback system for making music accessible to people with hearing impairments. A study by Feierabend [11] suggests how visual experience influences the understanding of the spatial allocation of sounds in a multiple sources identification context, but not necessary with a single source. Similarly, we can understand the sounds by associating them with a shape, or graphical form such as letters and words [12]. Furthermore, the perception and understanding of music and sounds affect the different levels of visual disabilities [13, 14, 15]. The haptic feedback experience from a musical instrument is critical in understanding the musical quality of an instrument [16], and thus can be a crucial factor for developing accessible interfaces for musical expression for both visually [5] and hearing impaired [17].

Alvarez [18] suggests that the exploration of art through touch has significant advantages regarding the overall understanding and perception of the art despite the visual inability. In addition, the sense of touch can challenge how we advocate other sensory input information provided by other senses. Lastly, Christidou and Pierroux [19] suggest that our understanding can be formed and changed through the comparison process of the actual touch experience, of an object, for example, and the anticipated visual information because of "texture, materiality, shape, temperature and size".

4. ANESTIS LOGOTHETIS GRAPHIC SCORE

The evolution of music during the second half of the 20th century led to the exploration of many new ways of producing sounds but also representing them. With graphic scoring being one of these innovative methods, pioneer composers like Anestis Logothetis explored how symbols and shapes can represent sound. His approach liberated him from the conservative way of notating music and let him add sonic features to his scores like timber, flexible form, time and speciality. Logothetis, quoted in [20], describe his process as follows: "When a piece of paper is used as a space for representing a sonic event, every point and line is brought into relationship with the entire surface [. . .]. The arbitrary correspondence between surface and symbols allows for the temporal associations of sonic events and the control of their duration; while out of the convention, the positioning of musical events high or low on the paper represents high and low pitches respectively".

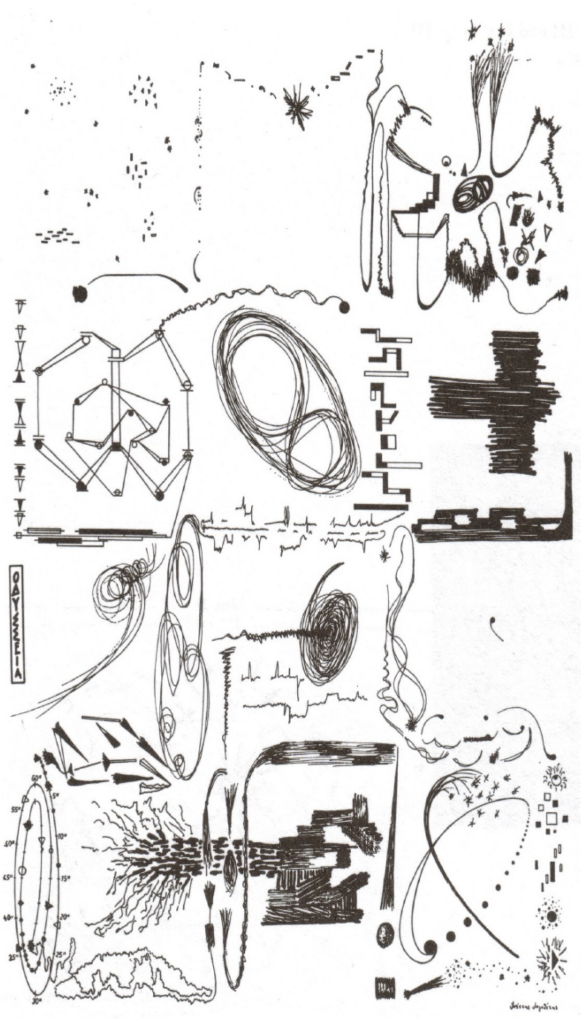


Figure 3. Graphic score *Odysee* (1963) by Logothetis.

Despite the freedom in the compositional process, his scores not only represent the musical information but also become a visual art piece that is aesthetically pleasing on its own, as seen in figure 3. Logothetis describes the physical *paper* as a space of interaction and music creation where the symbols become independent sonic entities ready to be performed. Moving forwards from this idea, we proposed that 3D tangible objects can be placed in space, ready to be manipulated and become interactive. Mongini [21] provides an overview of Logothetis aesthetic work, which provided the basis for conceptualising and designing these case studies. Furthermore, the analysis and taxonomy of symbols in Logothetis's work *Odysee* (1963) by Baveli and Georgaki [20], enable a better understanding of the selection of the individual visual structures and symbols that installation uses.

5. CASE STUDIES

The two case studies presented here are influenced by the graphic scores and the design progress of the composer. *Blocks Sound*, offers a minimalist approach for using 3D objects in space where *Logothetis Sound* has a more direct relationship and inspiration from the piece *Odysee*. In addition, it contains micro-scaled visual structures that can be used individually.

The following case studies reflect how such a graphical score can become the medium for interactive compositions for visually impaired users.

5.1 Blocks Sound

5.1.1 Introduction

Block Sound (see figure 4) installation examining how the audience responds and perceives sounds in an interactive composition. It requires participants to compose their music by positioning little LEGO-like blocks on the specified white area. The installation aims to transform the traditional musical notation into a playful tactile experience by moving and rearranging the different size blocks. Participants can interact with the block as if they are notes by simply dragging them across the board and altering the synthesis of the score. In addition, it introduces participants to musical concepts like pitch location (high and low). It makes interactive music-making tangible and accessible to people with lower or restricted vision [2] .

5.1.2 Process

The installation consisted of a lighted board, a PS3eye camera, audio speakers and LEGO-like blocks. The lighted board is placed within on a wooden box specially designed and laser cut to hide the laptop and any exposed cables. The installation is 37cm(W) x 40cm(H) x 24cm(D) excluding the speakers whose size might vary depending on the exhibitions circumstances. Blocks Sound has been developed in OpenFrameworks and Max [3] programming environments and uses a blob detection algorithm based on the

[2] *Blocks Sound*. It has been presented as an installation in various festivals such as Peckham Digital, Goldsmiths Pop-Up, Hack and Scratch (Trajectory Theatre), where participants have been asked to give feedback on their experience. https://vimeo.com/337125549

[3] www.cycling74.com

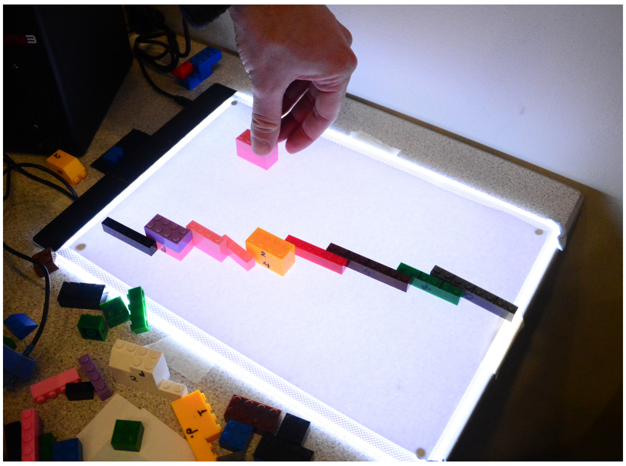


Figure 4. Blocks Sound.

ofxOpenCV [4] framework that can identify the location of the blocks within the frame of the board and translate these coordinates into individual moments of sonic events.

The camera, placed above the whiteboard, uses the algorithm, converts the incoming image into greyscale, and memorises the first frame as the default background colour. The lighted board emits white colour as the background so users can quickly identify blocks placed on top. From that point, any new object placed in the frame of this background (lighted board) is being identified as different, and its centred coordinates and contour coordinates are being saved in an array of new objects. A virtual cursor/ line moving from left to right, the same way we would read western music notation, scans the screen's pixels to find the top left corner coordinate of the identified contours.

To create a polyphonic system, the screen's height has been split into eight equal parts where only one block can be identified at the time. These eight lines are a decision taken considering the distance between the board and the camera and the number of blocks that can fit in the total height of the board. For example, if blocks are positioned on top of each other across the height of the board, they will not be able to fit more than eight blocks.

Moreover, when the cursor meets the identified object within the specified height region, it triggers a sonic event. In this way, we can have up to eight sonic events triggered simultaneously. When the virtual cursor reaches the end of the camera's window width, which is also the physical length of the whiteboard, it will loop the sequence from the start.

Following this process, the coordinates are sent to Max software programming environment via Open Sound Control (OSC) communication protocol, where they are creatively sonified.

5.1.3 Users' feedback

Seven members of the public attending the Peckham Digital [5] art exhibition were asked to provide feedback related to their experience, the interaction and the design of this

Computer vision	OpenFrameworks C++
Recognition of blocks	ofxOpenCV
Sending x,y location to sound program	OSC
Sonification of coordinates	Max

Table 1. Applications used for *Blocks Sound*.

installation. The range of participants was between 26 to 35 years, with an average age of 29.5. Out of the seven, the three were professional musicians, and the remaining four had no musical background. Regardless of their musical background and experience, all participants understood the interactions with the blocks, how they are used and what they represent. All participants answer positively to a question regarding the functionality of x and y axes and the blocks. When participants were asked how the tactile feedback experience helped them understand musical notation, six out of seven responded "Yes, significant" and the other participant replied, "Not so". Asking "how easy was it to create new musical patterns by changing the position of the bricks?", five answered "very easy" and two responded with "neither easy nor hard". Overall the feedback was positive, and the system was able to engage haptically and musically. One participant said: "really nice. It was fun creating music that way", and participant two commented, "great idea, creative and entertaining way to learn for someone!".

Although we were hoping to test the system with visually impaired attendees during the installation, we had no participants who volunteer. Future iterations of this research will include invited participants in a supervised and controlled environment. Whilst none of the participants was visually impaired, the feedback provided positively contributes to a proof of concept for tangible 3D objects applied in interactive graphic scores.

5.2 Case Study Two: Transforming Logothetis Sound

This case study builds upon the knowledge and feedback gained from case study one, *Blocks Sound*. When we observe the limitations of interaction and audio, the approach taken here was to develop more complex interactive objects with enhanced audio-tactile relationships. The case study uses composition *Odysee* (See figure 3) as the backbone of the system. *Odysee* consists of small sections bringing out freedom and flexibility in the interaction and form. The aim was to shape the 2D symbols from the score and transform them into 3D objects to create a more engaging tangible experience. The new objects need to be self-exploratory and understandable via the sense of touch aimed toward users with visual impairment.

In the first approach, we used graphic design software to extract the features of specific symbols from the 2D images and laser cut and engraved the symbols in wood (see figure 5). This process gave the signs individual tangible entities. Five moderately small blocks of wood, average 10cm(W) x 10cm(H) and wood thickness 4mm, represent five different parts of Logothetis composition. However, the results

[4] https://openframeworks.cc/documentation/ofxOpenCv/
[5] https://www.peckhamdigital.org/

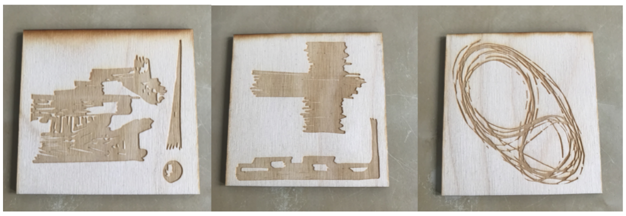


Figure 5. First attempt, wood engraving.

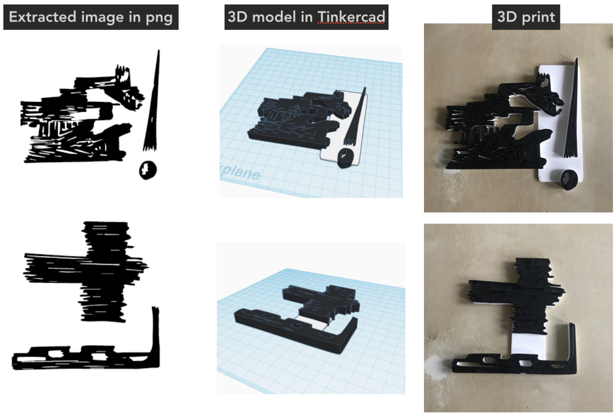


Figure 6. 3D objects.

of the wood engraving were not satisfactory in terms of the tangible feel and haptic feedback experience since the depth of the image was not deep enough >0.01mm, to provide a mental representation of the symbol.

The second step was to 3D print these specific symbols to enhance the features. Three of the graphic score symbols that were used for the laser cutting approach were imported into Tinkercad [6] software and have been extracted into 3D objects (see 6. The size of the final PLA 3D prints that have been used for this project is 110mm(W) x 100mm(H) x 8mm(D).

This approach gave more satisfactory results, and it has been used for the final version of this project. The next step of this project was to test the 3D printed symbols with computer vision algorithms. The complexity and refined detail shapes of the 3D printed symbols exclude the use of blob detection and contour detection. The approach here was to sonify all details available from the 3D object and not just the general position of the x and y-axis. Therefore the first test has been made with the darkest pixel detection algorithm where the black 3D objects have been identified instead of scanning the difference in the white background they are placed on. This algorithm, unlike the blob detection algorithm of the first case study, creates a rich set of data of recognised pixels.

Following the same technique as with *Blocks Sound*, the cursor line that moves across the window sends the coordinates of the identified pixels to a polyphonic synthesiser in Max via OSC. As a result, the synthesised sound is rich in dissonant frequencies reflecting the outlines, shapes and densities of the drawings of the composer.

Similar wooden box and electronic components found in *Blocks Sound* are also used for this case study. While the overall system of the installation remains the same, the individual 3D objects that are sonified are unique. Users can experience the tactile feedback, using their fingertips, from the 3D object and listen to the sounds generated from that object. They can also rearrange and perform with the 3D objects by placing them in a different order and angles to change light and thus the sound.

5.3 Case studies sonification process

These case studies aimed to create simple and understandable audio-haptic feedback relationships through interaction. *Blocks Sounds* has been through many sonic interactions due to the simplicity of information that the interactive blocks send to generate the music. As a result, single events in time generate endless sound synthesis ideas. The only limitation is that the pitch controls the number of blocks available on the Y-axis.

Unlike *Blocks Sound*, which has a simple and relatively linear interaction between the movement of the block and the sounds they produce, *Logothetis Sound* demands a different approach to reflect the aesthetics of the composer best. Therefore, the two sound approaches are blended and interconnected. The proposed sonification method here is similar to the *Blocks Sounds* on how the Y-axis controls the pitch of the sound and the X-axis the time. It applies physical modelling techniques to recreate orchestral sound styles similar to those available at the time of composing *Odysee* (1963) and still retain a level of interaction and engagement with the performance of the score. For a more organic approach, there is the option to use an existing recording of known performances as the source material. In addition, the system uses machine learning vision algorithms to recognise the symbols and generate the sound.

6. DISCUSSION AND CONCLUSION

The two projects *Blocks Sound* and *Logothetis Sound* support the idea of making alternative and contemporary music notation systems like graphic scores more accessible to visually impaired people. The acknowledgement of existing assistive technology, relevant projects, and the study of them through the prism of Human-Computer Interaction facilitates this research and design process. Feedback from the users of the first case study created a variety of questions regarding the affordance of this interaction and the possibilities of an extended application. The case studies can be helpful from an educational viewpoint for understanding music and sound notation and representation. In addition, the case studies are also a step forward toward understanding and creating graphic scores for visually impaired musicians. These new complex objects question the available technology and open the possibility of experimenting with different technologies. For example, machine learning, symbol identification and psychical computing can replace the current computer vision algorithm and provide more tailored audio-haptic feedback. Future iterations of these projects will concentrate on haptic interaction design and how it can become more effective and

[6] https://www.tinkercad.com/

accessible but also creative.

Acknowledgments

Thanks to Prof. Atau Tanaka for his support and guidance on the case studies and Robert Hall from Goldsmiths University of London for his help with laser cutting and the 3D printing process.

7. REFERENCES

[1] L. Kugler, "Technologies for the visually impaired," *Commun. ACM*, vol. 63, no. 12, p. 15–17, nov 2020. [Online]. Available: https://doi.org/10.1145/3427936

[2] W. Homenda, "Intelligent computing technologies in music processing for blind people," *Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications, ISDA'10*, pp. 1400–1405, 2010.

[3] T. Haenselmann, H. Lemelson, and W. Effelsberg, "A zero-vision music recording paradigm for visually impaired people," *Multimedia Tools and Applications*, vol. 60, pp. 589–607, 10 2012.

[4] A. Tanaka, *Embodied Musical Interaction: Body Physiology, Cross Modality, and Sonic Experience*. Springer Science and Business Media Deutschland GmbH, 2019.

[5] A. Tanaka and A. Parkinson, "Haptic wave: A cross-modal interface for visually impaired audio producers," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2016, p. 2150–2161.

[6] J. Mullenbach, D. Johnson, J. Colgate, and M. Peshkin, "Activepad surface haptic device," *Haptics Symposium 2012, HAPTICS 2012 - Proceedings, Vancouver, Canada*, 03 2012.

[7] A. Georgaki, S. Raptis, and S. Bakamidis, "A music interface for visually impaired people in the WEDELMUSIC environment. design and architecture," in *Proceedings of the 1st International Symposium on Music Information Retrieval (ISMIR 2000)*, Plymouth, United States, 2000. [Online]. Available: https://doi.org/10.5281/zenodo.1417291

[8] B. Di Donato, C. Dewey, and T. Michailidis, "Human-sound interaction towards human-centred cross-modal interaction with sound," in *Proceedings of the 7th International Conference on Movement and Computing*, ser. MOCO '20. New York, NY, USA: Association for Computing Machinery, 2020.

[9] L. C. Quero, J. I. Bartolomé, and J. Cho, "Accessible visual artworks for blind and visually impaired people: Comparing a multimodal approach with tactile graphics," *Electronics (Switzerland)*, vol. 10, pp. 1–19, 2 2021.

[10] S. Nanayakkara, E. Taylor, L. Wyse, and S. H. Ong, "An enhanced musical experience for the deaf: Design and evaluation of a music display and a haptic chair," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 337–346. [Online]. Available: https://doi.org/10.1145/1518701.1518756

[11] M. Feierabend, H. O. Karnath, and J. Lewald, "Auditory space perception in the blind: Horizontal sound localization in acoustically simple and complex situations," *Perception*, vol. 48, pp. 1039–1057, 11 2019.

[12] R. Bottini, M. Barilari, and O. Collignon, "Sound symbolism in sighted and blind. the role of vision and orthography in sound-shape correspondences," *Cognition*, vol. 185, pp. 62–70, 4 2019.

[13] A.-A. Darrow and J. Novak, "The Effect of Vision and Hearing Loss on Listeners' Perception of Referential Meaning in Music," *Journal of Music Therapy*, vol. 44, no. 1, pp. 57–73, 03 2007.

[14] L. Pring, K. Woolf, and V. Tadic, "Melody and pitch processing in five musical savants with congenital blindness," *Perception*, vol. 37, no. 2, pp. 290–307, 2008.

[15] B. Jones, "Spatial perception in the blind," *British Journal of Psychology*, vol. 66, pp. 461–472, 1975.

[16] C. Saitis, H. Järveläinen, and C. Fritz, *The Role of Haptic Cues in Musical Instrument Quality Perception*. Cham: Springer International Publishing, 2018, pp. 73–93. [Online]. Available: https://doi.org/10.1007/978-3-319-58316-7_5

[17] S. C. Sanayakkara, L. Wyse, S. H. Ong, and E. A. Taylor, "Enhancing musical experience for the hearing-impaired using visual and haptic displays," *Human–Computer Interaction*, vol. 28, no. 2, pp. 115–160, 2013.

[18] A. Alvarez, "Please touch: The use of tactile learning in art exhibits." From Content to Play: Family-Oriented Interactive Spaces in Art and History Museums Symposium, Paul Getty Museum, June, 2005, pp. 4–5.

[19] D. Christidou and P. Pierroux, "Art, touch and meaning making: an analysis of multisensory interpretation in the museum," *Museum Management and Curatorship*, vol. 34, pp. 96–115, 1 2019.

[20] M.-D. Baveli and A. Georgaki, "Towards a decodification of the graphical scores of Anestis Logothetis (1921-1994). The graphical space of odysee(1963)." SMC-5th Sound and Music Computing Conference, Berlin, Germany, 2008.

[21] C. Mongini, "Sign and information: On anestis logothetis' graphical notations," *Deleuzeand Contemporary Art*, p. 227, 2010.

NOTATION FOR ORGAN EXTENDED TECHNIQUES

Cláudio de Pina
CESEM
Lisbon, Portugal
claudiopina@campus.fcsh.unl.pt

ABSTRACT

The notation of extended techniques on the organ do not have a clear standard. Composers use their own notation or guiding rules to express their sonic expansion on the instrument. Since 1960, the most prolific period on organ experimentation, Ligeti, Kagel and Cage were the first to become known for using non-standard notation on the organ. From this collection of works, the ones from Ligeti are paramount. From the graphical score of *Volumina*, the long clusters of *Harmonies* to the fast torrent of notes of *Coulée.* Kagel develops further with other notations for clusters and graphical gestures in *Phantasie für Orgel*. Finally, with Cage we find a simpler way of notating long notes and stop changes. Kurt Stone has a chapter on his book, *Music Notation in the Twentieth Century*, regarding notation on the organ, but it not describes half-drawn registers, half-depressed keys and does not present anything regarding motor or other air manipulations. A clearer notation and explanation is needed for these extended techniques. In an instrument so tied to a functionality, it is pertinent nowadays to re-incorporate these techniques in the contemporary organ repertoire. A new simple notation is presented, alternative to graphical notation or lengthy performance notes. This will create an easy understandable approach.

1. INTRODUCTION

The protean Hungarian composer György Ligeti (1923-2006) caused a paradigm shift on the organ in the 1960s. His work with electronics in the WDR[1] shaped his mind to use his findings through acoustic instruments. His seminal organ work, *Volumina* (1961/62, rev. 1966) is fully written in a graphical form [1]. In this way, several clusters that need elbows, hands, and feet, are layered down with massive black and white shapes. Each sign representing a chromatic, diatonic or pentatonic cluster.

In the middle section we see a chaotic graphical form. This represents something akin to a guided improvisation. Webbed lines and dots form several patterns with a healthy guide note of how to perform them. It quickly becomes cryptic for the unprepared organist [2]. A composer that would choose this way of writing, even nowadays, would meet a hard reception on any organ composition contest. In *Volumina* a vertical line coincides with the organ compass and establish the only guiding notes. The graphical score reminds works in a same vein of Cornelius Cardew [3]. A decade later a similar graphical score is done, by Rainer Wehinter for Ligeti's electronic work *Artikulation* (1958). Nevertheless, after *Volumina*, Ligeti worked on two organ studies. The first *Harmonies* (1967), contains a fanning out of ten pitches from a tonal centre, C, decreasing in several voices towards the end. Ligeti had the idea to 'starve'[2] the wind supply to achieve an unnatural sound. These extended techniques and resulting sonority are not written on the score, but detailed on the performance notes. Also, special advises are added regarding the registration, dynamics, tempo, and even practicality considerations like octave transpositions and specific registration needed, which none are indicated on the score itself [4].

In the second study, *Coulée*, like the previous one, he also indicates exact pitches of notes. They should be played as fast as possible, to sound like a cluster. Although every note is a suggested eight-note in a continuous stream, one can assume the influence of his harpsichord piece *Continuum* (1968). At the end exists a remark on the full duration of the work, about 3' 30'' [5]. This annotation is also present in *Harmonies,* in this case 6' to 9', what can be a clear influence of Bartok.

This article will neglect Ligeti's first organ work *Ricercare per organo, 'Hommage to G. Frescobaldi'* (1953) since it does not bring anything new regarding notation. Written in the style of a chromatic *ricercare*, the original manuscript is owned by Ove Nordwall, using standard notation. A special note that this work is almost identical to the eleventh movement of *Musica Ricercata* (1951/53) for piano [6, 7]. Ligeti planned to work on further studies (*Zéro* and *Le son royal*) but they were not done, and no information remained to create a fair assumption of what path he will take [7].

The Argentine-German composer Mauricio Kagel (1931-2008) wrote several organ works with his own

[1] Westdeutcher RundFunk – West German Broadcasting, Cologne.

[2] Term used by Ligeti for decreasing the air supply.

notation. *Improvisation Ajoutée* (1962), the cycle *Rrrrrrr….* (1980/81) and *Phantasie für Orgel mit Obbligati* (1967). None have the same scope of graphical notation we found in *Volumina,* having instead a careful mix of clear notation and only a few graphical forms. *Improvisation Ajoutée* uses regular staves for the assistants, choir and stop changes, besides the manuals and pedal. Of special note, which this article reflects, is *Phantasie* were we find a more audacious approach, regarding organ techniques. A 'tablature like frame' fills the score consisting of; Register, manuals, pedal and tape-machines. The score is divided by in seconds. In the manual and pedal section we find only three lines indicating a tonal area with regular rhythms written, but without bar lines. The tape part, for two tape-machines, indicates start, stop, and dynamics. The tapes consist of previously recorded sounds in a theatrical vein (typical of Kagel); Rain, Hail, Flushing of Toilet, Breakfast, Organist Walk and a big finale of Baptisms, Marriage, and Funeral services. In general, it is heavily scripted with fast-paced changes. The clusters are regularly notated, although, without a standard stave only a general idea of the tonal range is given. We can find a direct quote to the graphical *Volumina* in the middle section with black triangles filling the score. After this middle section appears a regular notation, a *pastiche* of Fréderic Chopin's *Étude op.25* mov. VIIII, superimposed with the VI and X movements. Except for this piano quotation, all the rest would be quite cryptic for the uninitiated on vanguard techniques on organ.

In *Rrrrrrr…* we find usual notations for several manuals with the special mention to the technique of slowly releasing the keys, changing the air flux.

Regarding John Cage, his most known organ work is definitely *Organ²/ASLSP* (1987), an adaptation of *ASLSP* (1985) for piano solo (that can also be played in organ). In both titles the abbreviation stands for "As slow as possible". The inspiration for the title refers to "Soft morning City! LSP!" the first exclamations in the last paragraph of *Finnegan's Wake* by James Joyce. They are coined has one of the longest pieces in existence, since Cage did not mention the duration. This piece is still being played in the *John Cage Organ Project*[3] in Halberstadt (Germany), and it will be continuously played until 2640.

The impact of the *ASLSP* variants his unavoidable to the organ, but his other works are also pertinent, although with few remarks regarding notation. *In Some of 'The Harmony of Maine (Supply Belcher)* (1980) for one organ and six registrants (assistants that will change the organ stops) we see a clear change of scenery. Special care was given to the stops changes teamed with indeterminacy methodology and a clever *pastiche* use of known American musical themes. *In Souvenir* (1983), commissioned by the American Guild of Organists, the same thematic use of a motif can be seen, but with a clear path to the *ASLSP* variants, regarding notation, duration and clever use of manual and pedals. But there is not much new regarding notation techniques, so the idea of using weights to achieve long notes (and notate then) is the most pertinent one.

From Cage's works we can see the usefulness of weights for the long notes of *Organ²/ASLSP,* but further indications are needed for the correct use of weights, concerning, placing and removing them. Besides taking into account the preparation need by the organist. A note can be said about using pencils or *ohashi*, that depend on the disposition of the keyboard and console. The word 'weight' will be used in this article to generalize the idea of depressing a key with an object. This technique needs a carefully thinking from the composer since you need two movements, limiting the gestures of the organist, one movement to fix a note (or group of notes) and another to remove it [1].

These examples will be used to introduce extended techniques and their notations. Each delve on a new approach that result in new sounds. Taking into account all techniques created by the mentioned composers one can summarize that they use:

- Changing air pressure (by manipulating the keyboard or stops),
- Using weights for long notes (therefore liberating one or two hands for other manipulations),
- Turning on and off the motor engine (cutting the air supply).

One could easily mix these three approaches creating a *bric-à-brac.* For example, one can use different weights (that would not fully depress the keys, causing a smaller air flux) to fix certain notes, with already prepared registers half-drawn and after a few seconds turning the engine off. One could imagine a score that would have the notes to be played (or in this case, fixed) and some notation for the stops and turning off the engine. But that would not be clear to every organist. At a first glance this kind of performance would need thorough notes from the composer with a possible debate with the organist and his insight of the instrument that would be used. This task is cumbersome to each part, so a simpler notation will be presented in this article.

2. LIGETI

Ligeti used vanguard techniques that need to be further analysed. Musical notation can be defined in very different ways, quoting Cornelius Cardew and his definition that which determines what you can say and what you want to say, determines your language [3]. But not all can be translated well for an unprepared organist or composer. In *Volumina*, it is possible to perform with the graphical notation, after careful consideration of the performance notes. Although the graphical form is more helpful regarding stationary and moving clusters.

In contrast, *Harmonies* uses two treble clef staves to indicate specific pitches contained in each cluster. This style of notation, very different from *Volumina*, tries to achieve a similar sound palette, but this time with fixed notation on a score. The black or white note heads are

[3] https://universes.art/en/specials/john-cage-organ-project-halberstadt

just a marker for performance. Since the tempo value of the notes and bars are not fixed, the change of note heads aids the performer to follow the score. A special note to the use of several *fermatas* to emphasize the duration [8].

Quoting Ligeti regarding the performance notes of *Harmonies*: “Nowhere in the piece should the chord successions create an impression of meter or periodicity”. The notion of passing chords and the value that they can represent is left to the performer. Since the changes are always in minor seconds, one could assume a contra-punctual resolution of leading voices or passing notes. *Harmonies* does not follow any harmony rules. All the note changes are made to have a direct consequence on the resulting starved sound. Another quote from the performance notes: “The whole piece is soft to very soft. Pale strange, ‘vitiated’ tone colours must predominate. Denaturing the sound is achieved by ‘greatly reduced wind pressure’”.

At the end, a single note is added by the pedal, further decreasing the air flux. There is no meter signature and no time value given to the notes. Only a mark: “Rubato, sempre legatissimo”. Nevertheless, there are bar lines. The moving pitch is notated with a white head and the black notes slurred continuously on overlapping bars. Both note heads are stemless and beam less, indicating no rhythm whatsoever. The ten note chord decreases in number towards the end, remaining a three note cluster. Ligeti warns that the ‘passing notes’ are always a minor second away. The final thirteen bars use *fermata* (few of them stacked), while the texture is being thinned in the amount of voices. It can also be said, that the last thirteen bars are akin to a filtering technique, like an electronic filter would cut the sound in few parcels. In the score there is no indication of registration or assistance needed and dynamics are also absent [9].

In second study — *Coulée,* the continuum motion of eight-notes contrast to the suspended clusters of *Harmonies.* It does not appear to be a cluster or micropolyphony composition, but when we take into account the reverberant space of a large church and a carefully chosen registration we can listen to a sound result akin to a cluster. Only when we listen to several recordings we truly perceive the diverse interpretations of *Coulée*. Each hand is playing *arpeggiated* groups of 2, 3, 4 to 5 notes. No bar lines exist, merely a suggestion has a dotted line. These clusters continue to evolve until m. 100, where they contract to a single pitch, the tonal centre, Ab. This effect is maintained in a few bars with a movement similar to an octave string tremolo. The correct tempo should keep the duration within 3 and a half minutes. The stream of notes abruptly stops at the end, near the high limit of the instrument (much like *Continuum*, which follows the same composition methodology, albeit in harpsichord) [9]. The pedal notes are near inaudible and only reaffirms a cluster like a freeze convolution effect [11]. The repose and tension, unlike *Harmonies*, are implied by the lack or varying of pitches and number of notes. From this density of superimposition, a binary form can be deducted, A — Repose and B — Rhythmic tension.

3. A NEW NOTATION

It is clear that there are important aspects for the performance of Ligeti organ works outside the notation. Without the performance notes, it is impossible to perform them. The main concern of this article is proposing a clearer and concise notation for extended techniques on the organ, based on the mentioned repertoire, and without the aid of performance notes.

The sound morphology of the pipe organ does not permit the same interpretation has in other instruments. The key connected to the pipes, brings a sound with always the same envelope. The way it plays, a performer can only adapt articulation between the keys and the acoustics of the space to bring forth a degree of expression. The same applies to dynamics, the organ can only add or subtract registers in layers. Albeit some instruments have a volume pedal, that either brings more registers or open/ closes certain cabinets of pipe ranks, thus mimicking a *crescendo* [11].

The use of extended techniques breaks some of these paradigms. One can use a half-key technique for creating a small *glissando* and *crescendo*. Manipulating the stops, one can achieve a tremolo effect, or bring non-pitched sounds. Like in *Volumina*, by turning on and off the engine while some keys and registers are drawn, achieves another kind of sound, more akin to synthesis programming in a computer (an out-of-the-box comparison would be the Deep Note[4]) [11]. For the proper use of these techniques, a systematization of notation is needed to help composers and organist understand better what is happening, instead of reading performance notes. In this way, it furthers develop the interest on these techniques.

Several compound examples will be described, that are suitable for a clear perception and reading. Many of the composers that where quoted have their own notations, and some of their ideas are put into practice. One can and should adapt them to the purposes needed for the intended purpose.

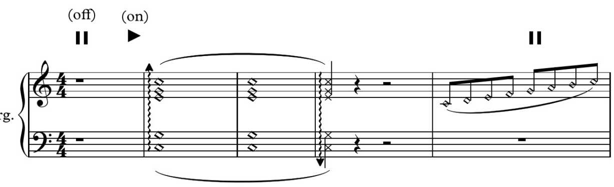


Figure 1. Turning off and on the engine. Weights noteheads.

In Figure 1 we have a clear logical sign for turning the motor on and off. The duration of the decay/rise of the sound is proportional connected to the amount of pipes that are feed by the wind. This means that many keys and/or many pulled registrations will behave differently. An introductory test is highly advised. By applying weights to the keys, with a notated rhythm, by only changing the note-head, creates a new way of interacting with

[4] A trademark of THX Ltd used previews in cinema theatres.

the keys. Using *arpeggio* to remove the weights in a certain order is a natural consequence.

Another possible way is to use words like FIX, used in *Australpnea* (2010) by Frederik Neyrink (1985) or simply use an *ossia* stave for longer values or keys on another manual [9].

In Figure 2 are examples aimed to gradually open stops. These can only be made with organs that have mechanical stops. The mechanics of stops are quite sensitive and different from each stop and organ, so once again, prior experiments are advised.

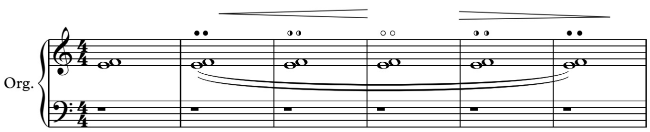


Figure 2. Gradually opening stops.

The *crescendo* and *decrescendo* lines helps the organist, assistant or composer in having a general idea of the dynamics that happen with half-drawn stops. This technique is compounded with the dot sign, gradually white, black or half-black. The use of a dot for the least amount of pressure is also correlated to the way the holes are closed in wind instruments. The white dot is the most amount of pressure on the key (or the usual amount of air to play a regular note).

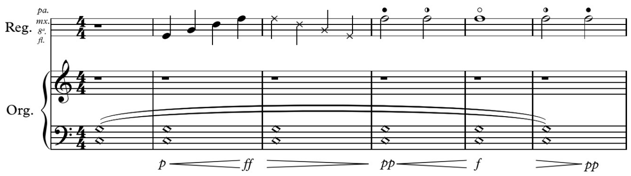


Figure 3. Stop tablature. Dynamic lines.

Using a stave (or tablature) for the stops brings another level of creativity and expression. One can notate rhythms to be 'played' by opening and closing the stops. Usual note-head for opening, cross-head for closing. When one needs to gradually open/close the stop, we use the same dot sign. In this case it is directly correlated to the way the registers work inside the organ, again, like keyholes in wind instruments. Again, dynamics marks are here to guide the organist, assistant and/or conductor concerning the general dynamic. The dynamics should follow the amount of open stops and their quality.

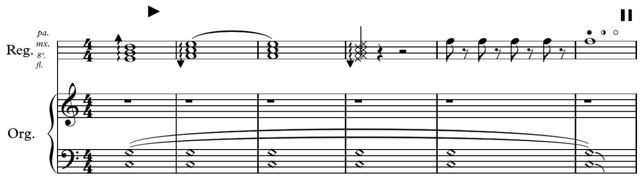



Figure 4. Applying weights. Mixed use of engine and stop manipulations.

In Figure 4 we can examine a compound use of several techniques. Using weights brings the possibility to the organist to 'play' with the stops. It also frees the hands to control the engine or prepare for the next stop change. The *glissandi* lines on the last measure are the indication of slowly removing the weights, mimicking the half-drawn stop technique mentioned before. It is similar to the slowly risen key example from *Rrrrrrr...* by Mauricio Kagel [8].

It is still possible to use a more usual way of notating stop changes with + or – signs. For a specific stop change this is clearly the most understandable. Since organs have several manuals and ranks this notation helps differentiate them from half-drawn stops. The same applies to organ with combination or feet piston that activate or deactivate several ranks, like in Figure 5.

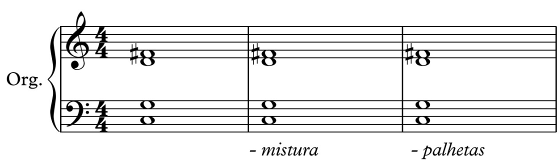


Figure 5. Stop changes.

This is the usual way of notating registrations changes in Portuguese historical organs, that is similar to other historical instruments. A subtraction or addition sign is used to call certain families of registers and is also present on modern organs. Some can even save entire stop changes that can be called upon with pistons beneath the manuals or pedal board. In these organs there are usually some feet control for families of stops, being reeds and mixtures the most common in historical instruments. Some organs can also couple or decouple manuals and also store entire performances on demand with electronic means. All of this does is already attended for in regular notation for organ. Even though there are a few exceptions, like the organ of St. Peter Station or Kassel in Germany, that can easily control the air flux, making it easier to perform extended techniques, these notations will help even on these modern instruments. This is outside the scope of this article, but nevertheless, with inventiveness, one can achieve the same sounds with a not so modern organ, being historical or not.

CONCLUSIONS

These techniques have a high amount of permutations and iterations with each other. This article presents a compound sample. A secondary objective is to make it easier to experiment with them. Since the result of this notation for extended techniques are based in the work of the quoted composers, an audition of this repertoire is advised. It is also recommended comparing new and old interpretations, like original recordings by Zsigmond Szathmáry, and recent ones by Dominik Susteck (both published by Wergo, Schott)[5].

Detaching this notation from a pure graphical form will defeat the inertia of composers and performers regarding extended techniques on the organ. This way is easier to understand and will not further intimidate young students. Besides being straight forward to all levels of organ technique. The potential of using this notation for young organists will assist both the teacher and apprentice, creating a common ground of notation that can be easily done in manuscript. New repertoire could be easily created this way, without the need of special vector graphics software or other, closing the gap of contemporary music on the usual repertoire of the instrument. These proposed signs and symbols are available in the most used notation software (Sibelius, Finale and Dorico), and if not, they are easily created. They can easily be drawn using pencil and paper, still a proven way to draft a musical idea quickly. With this notation one could change the paradigm of writing contemporary music for the organ during the undergraduate studies of young composers, who most see writing for this instrument has a daunting task.

A note could be said that even with modern organs who can produce extended techniques easily this notation will help (Orgelpark[6], St. Peter[7] and Kassel[8]). From a point of view of the composer, one should use a level of logical that can be used in any case. From the point of view of the organ teacher, one can assume the pedagogical benefits of using a simpler notation, to prepare the students to deal with certain techniques on the instrument. The same sign codification has other contemporary repertoire, with some logical signs added, can also help anyone, with sufficient musical reading proficiency to have a clear idea of what his happening and how to play.

This article proposes notations and techniques that can be used in any organ. In fact, most of these instruments are entirely mechanic in nature and the degree of control of the keys and stops are directly connected to the organ innards, providing a sense of direct connection to instrument. In some cases they indulge a deeper experimentation. Follow these guidelines has a recipe with the mind set that Ligeti purposed: "be inventive."

Acknowledgements

This work is funded by FCT - Foundation for Science and Technology, I.P., within the scope of the project *Sound expansion in the historical organ and contemporary music interpretation practices.* Furthermore, I would like to thank the CESEM, FCSH/UNL - Centre for Studies in Sociology and Aesthetics of Music, my research group GIMC – Research group of Contemporary Music and my Ph.D. advisors Dr. Isabel Pires and Dr. João Vaz.
Further acknowledgements to the Parish of Nossa Senhora da Ajuda for permission to develop these extended techniques in a historical Portuguese organ.
The scores from the composers are mentioned by kind permission of Universal Edition, Schott Edition and Peters Edition.

REFERENCES

[1] S.C. Anderson (ed) *Twentieth-Century Organ Music.* New York: Routledge, 2012.

[2] G. Ligeti, *Vorwort zur Partitur Volumina*, Frankfurt: Peters, 1967.

[3] C. Cardew, *Notation – Interpretation*. Tempo LVIII. 1961.

[4] G. Ligeti, *Vorwort zur Partitur Étude n.º 1 – Harmonies*, Mainz: Schott, 1969.

[5] G. Ligeti, *Vorwort zur Partitur Étude n.º 2 – Coulée,* Mainz: Schott, 1969.

[6] G. Ligeti, *Vorwort zur Partitur Volumina*, Frankfurt: Peters, 1967

[7] P. Albèra, C. Fourcassié, P. Michel (ed), *György Ligeti L'atelier du compositeur - écrits autobiographiques – commentaires sur se oevres.* Géneve: Éditions Contrechamps, 2013.

[8] A. Blackburn, *The Pipe Organ and Real-time digital signal processing,* doc. diss. Melbourne: Griffith University, 2011.

[9] G. Collins, *Avant-garde Techniques in the organ works of György Ligeti, A lecture recital, together with three recitals of selected works of J. Alain, J. S. Bach. W. A. Mozart, M. Reger, and others,* doc. diss. Texas: North Texas State University, 1980.

[10] G. Ligeti,"Was erwartet der Komponist der Gegenwart von der Orgel?"; *in György Ligeti, Gesammelte Schriften*, Mainz: Winterthur. 2007.

[11] H. Olson. *Music, Physics and Engineering,* New York: Dover, 1966

[5] https://en.schott-music.com/wergo/

[6] https://www.orgelpark.nl/

[7] https://www.sankt-peter-koeln.de/wp/kunst-station/ubersicht/

[8] http://www.ekkw.de/kassel-mitte/martinskirche.php

CONDITIONAL SEMANTIC MUSIC GENERATION IN A CONTEXT OF VR PROJECT "GRAPHS IN HARMONY LEARNING"

Anna Shvets
Fablab by Inetum
Paris, France
anna.shvets@inetum.com

Samer Darkazanli
iMSA
Montauban, France
darkazanli.samer@imsa.msa.fr

ABSTRACT

The article proposes a perspective on the use of generative artificial models in a context of the VR project "Graphs in harmony learning". The usage of LSTM, ConvLSTM and conditional GAN with convolutional 1D layers for semantic music generation is discussed. The efficiency of the novel data encoding scheme, along with the design patterns based on the system of graphs, are shown.

1. INTRODUCTION

1.1 Context

The article presents a new system of representation, based on a graph theory [1]. The representation methodology has been proven efficient in a multi-step pedagogical experiment in a context of hybrid learning. The experiment demonstrated a substantial increase of the quality of knowledge in a group of students, benefiting from the system application in a learning process [2, 3]. The method has also been applied in building graphic interface of the award-winning mobile and VR applications [4, 5].

1.2 Motivation

The global pandemic crisis had shown the need for research of convenient forms of distance learning to compensate the reduced interactions, detachment and isolation of individuals, which brings harmful consequences. As student surveys indicates, more than one in two students had thought of dropping the classes during the pandemic and 71% of surveyed confessed being worried about their mental health[1]. The same resource points out the lack of immersion for the videoconference format of courses, resulting in great difficulties in following the course content, which led 70% of the surveyed students to express their pessimism about their academic success. Another study[2], listing the reasons of dropping the university, mentions a lack of practice, especially for individuals who are not equipped for theoretical courses of the university's curriculum.

With the application of teaching method based on the effective representation methodology and via the interaction with the tangible elements of virtual reality, it is expected that the degree of immersion will closely approach the level of practical face-to-face lessons.

2. STATE OF THE ART

2.1 AI in Education

A meta-analysis studying the application of AI in teaching, which included 146 recent articles in the field [6], proposes a categorization according to the target group, such as learner-oriented AI, teacher-oriented AI and system-oriented AI. Following this taxonomy, the learner-oriented AI tools comprise software solutions to learn a subject, such as adaptive or personalized learning systems and intelligent tutoring systems; teacher-oriented AI tools are used to help the teacher in reducing a workload by automating tasks such as administration, assessment, feedback and plagiarism detection; system-oriented AI tools provide information to administrators and managers at the institutional level to monitor the acceptance rates in faculties or the employability trends.

The development of intelligent tutoring systems is mainly focused on visual recognition, aiming to provide a system feedback coherent with the implicit reactions of the learner [7, 8]. Another trend comprises creation of intelligent campus ecosystems, with a strong reliance on chatbot solutions [9]. The above mentioned meta-analysis reveals a lack of educational theory to support a technological choice. Current research in the field is oriented towards analysis of data patterns to build AI models or to support administrative decisions using known statistical and machine learning methods. The authors point out that there is very little evidence for the advancement of pedagogical theories related to AI-based educational technology.

The VR project specified in a present article makes a big difference compared to the state of the art by proposing the application of an innovative pedagogical strategy. This strategy is based on a new knowledge representation methodology, it exploits the possibilities of AI for the forms of

[1] https://etudiant.lefigaro.fr/article/covid-19-depuis-le-debut-de-l-epidemie-plus-d-un-jeune-sur-deux-a-envisage-d-arreter-ses-etudes_65135afa-622e-11eb-8fde-d92bf2ba0bfe/

[2] https://fr.cursus.edu/10328/les-enjeux-du-decrochage-universitaire

activity inherent to the artistic practice (analysis of the structural properties of chords in harmonic sequences), and therefore respond to the needs of educational process.

2.2 XR in Music

Another recent meta-analysis [10] brings together research on virtual reality (VR), augmented reality (AR), augmented virtuality (AV), mixed reality (MR) and extended reality (XR), applied to music. The meta-analysis covers 260 publications appeared from 1990 to 2020 which present musical research in XR, encompassing technical, artistic, perceptual and methodological fields and reveals an exponential growth of publications starting from 2015, which is explained by the increased accessibility of XR hardware and software tools.

The study proposes a definition of the term *musical XR system* by classifying the existing systems according to the ways the sound and music are used (diegetic mode of usage versus non-diegetic use), the cinematics of sound events (fixed audio playback versus sound events triggered in an interactive manner or procedurally generated), and sound spatialization (fixed stereo sound presentations versus dynamic spatial audio implementations based on user location tracking). Finally, the grouping of articles is made according to the main objective of the musical XR system (performance, education, composition, sound engineering, entertainment, perception study, development), end-user (performer, student, composer, member of the audience, studio engineer, developer), social experience (individual or multi-user experience) among others.

This grouping allows the assumption that the existing educational applications of virtual reality mainly focus on improving the practice of performance, using interfaces of traditional instruments modelled in 3D (piano, guitar, drums), neither of applications presented a system of abstraction of musical knowledge. Most of applications focus on the beginner level and a very few are designed for the expert or intermediate level musicians; almost all systems were made for students and dedicated to self-training.

In view of the state of the art in the field, the VR project in a present article is not only in accordance with the best practices in the construction of a musical XR system (spatialization of the sound source, triggering of the sound events in an interactive way, the unfolding of musical content in a diegetic way), it also fills the gap for a graphical interface independent from attachment to a specific musical instrument. It also proposes the systematization of musical knowledge in a non-verbal representation, compatible with immersive worlds.

2.3. Conditional music generation

The approaches to music generation process falls into three groups: conditional, controllable and constraint generation. Conditional generation takes one element as input to generate another element as target [11, 12], while controllable generation uses the change in input features to manipulate different aspects of the output generation [13, 14]. The third group, containing constraint generation, makes use of the template-based approach to influence a shape of the output result [15]. The research in controllability mainly explores features disentanglement, proposing systematic studies [16] and datasets [17], designed to foster further experiments in the field. The existing resources, however, are mostly gathering monophonic music examples and therefore are not suitable for harmonic sequences generation.

The research in conditional music generation presents a spectrum of generative architectures, such as LSTM, Transformer [18], GAN [19, 20], hybrid versions, such as LSTM-GAN [13] or GAN with an inception model [21]. The latter architecture exploits convolutional layers, followed by the time distribution layer that captures sequential data, which enforces the convolutional layers considering the time relationship in a similar manner as RNN layers. As a comparison to this type of architecture, a hybrid ConvLSTM architecture processes the sequential data, where each element of the sequence passes through a convolutional layer followed by LSTM layer [22]. The previous experiment has shown that this type of architecture captures well two-dimensional sequential data [23] and will be exploited in the experiment on novel encoding scheme application, presented further.

3. METHODOLOGY

The methodology consists of the system of graphs and an augmented score representation of harmonic sequences. The system of graphs is organized in horizontal and vertical triads, reflecting logical relations between chords, whereas the augmented score contains a meaningful color scheme for a visual distinction of the chord functions, along with color shades, revealing the chord structure.

3.1 The system of graphs

The system of graphs embraces functional correlations between chords, consisting of the Roman numerals, representing degrees of the scale, placed in a specific order [24]. The placement order is defined by variation potential within a multitude of diatonic progressions. The slots containing the information about chords are mapped to a specific type of information. In consequence, the slots reserved for the seventh chords cannot contain the information about triads (see in Figure 1). This limit comes from the theory of graphs [1], defining a graph as a graphical representation of frames, corresponding to a specific knowledge representation.

Figure 1. Unfilled graph structure (on the left) along with the graph filled with the information about three passing progressions, between the seventh chord of the tonic and its first inversion (on the right). The tonic seventh chords

are inside the tops of the graph and three triads are inside the edges of the graph.

The graphs are organized in horizontal triads (see Figure 2) and vertical triads (see Figure 3).

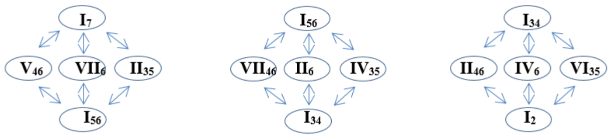


Figure 2. Horizontal triad regroups the graphs containing tops sharing the same degree. In this example, a horizontal triad embraces all possible passing progressions between the tonic seventh chord and its inversions.

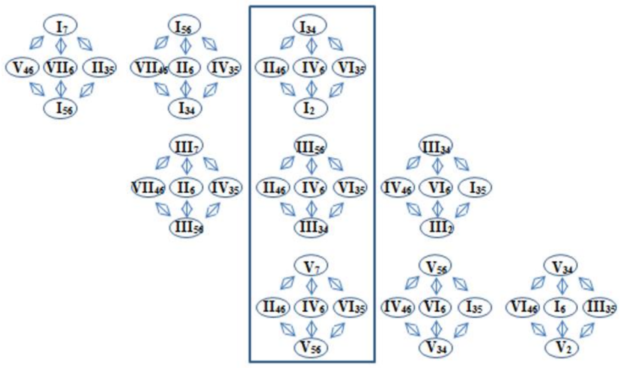


Figure 3. A vertical triad groups the graphs, sharing the same information inside edges.

A harmonic sequence is therefore represented as a path within the system, connecting the nearest instances of the chord in the system of graphs.

3.2 The color mapping

The second element of the visual representation methodology within a framework of this project consists of an augmented score, which explicitly identifies the structure of chords with their adherence to musical functions (tonic, dominant, subdominant).

The color gradient aims to visually distinguish the tones of a chord in relation to their importance in the structure of a chord (an example for the tonic seventh chord is shown in Figure 4). Such an explicit representation becomes particularly useful for an open position of chords – a range of open positions of the tonic triad and seventh chord with their inversions is shown in Figure 5.

Another visual contribution is based on a palette of colors, gathering the degrees in groups of functions – tonic (I), dominant (V, III, and VII degrees) and subdominant (IV, II and IV degrees), as shown in Figure 6.

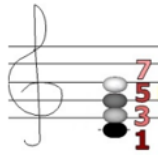


Figure 4. Color shades, that show the importance of a tone inside a chord structure.

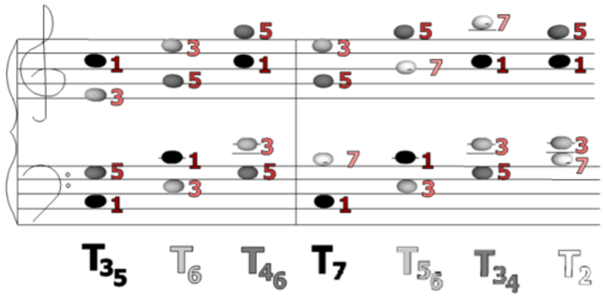


Figure 5. The chord inversions in an open position might create difficulties for students in finding the fundamental tone of the chord. The explicit shading of tones helps to lift the complexity of the graphical representation, allowing to focus the attention on the chord phonism.

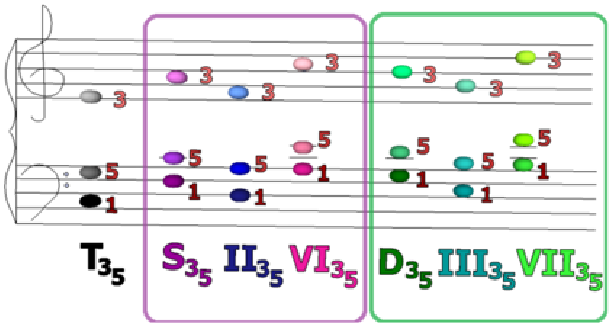


Figure 6. The color palette for the function groups: tonic (T), subdominant (S, II, VI) and dominant (D, III, VII).

The application of this representation methodology in a pedagogical process in a blended learning format was very successful [2, 3], which can be explained by the conformity of the graph structure with the possibilities of the short memory (episodic buffer) functioning, able to retain four to seven elements at a time (depends on a person). Indeed, a graph in a constructed system contains five elements, where two elements are grouped together, because they represent the same function and contain the information about 4-notes chords (the top and the bottom vertices of the graph); the remaining three elements are grouped by their structure, as they represent triads, being in a relation of the interval of a third between them.

4. VR APPLICATION

The VR project “Graphs in harmony learning” is based on the original methodology of tonal harmony representation described previously. The methodology elements receive a tangible 3D embodiment in a context of VR, enabling multimodal interaction. The VR context aims to increase a degree of immersion and therefore facilitate understanding and practice of the learning material.

The content of the application is divided into several VR rooms – the *Entry* (see Figure 7), explaining the purpose of the application, the *Temple of Knowledge* (Figure 8), which prepares the user to understand the content of the main activity, the *Study Room* (Figure 9), where the interaction with the graph system occurs. The *Test Room* and the *Practice Room* (Figures 10 and 11), propose the activities to test the acquired knowledge and put it in practice. Both latter rooms are based on AI models.

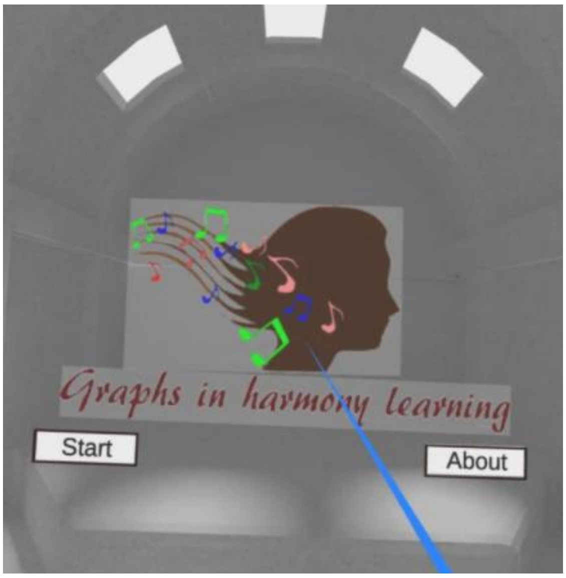


Figure 7. *Entry.*

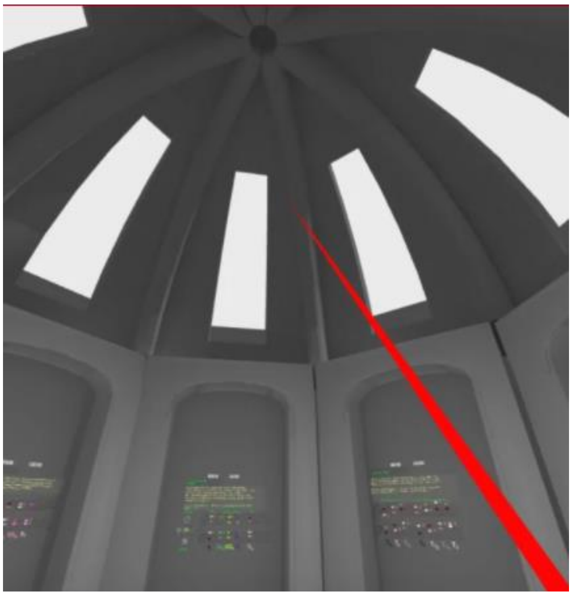


Figure 8. *Temple of Knowledge.*

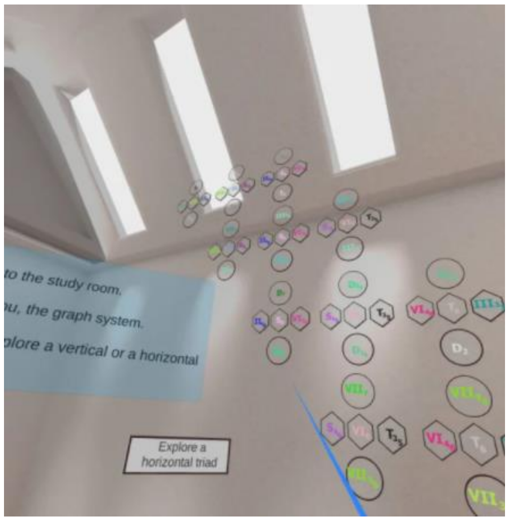


Figure 9. *Study Room.*

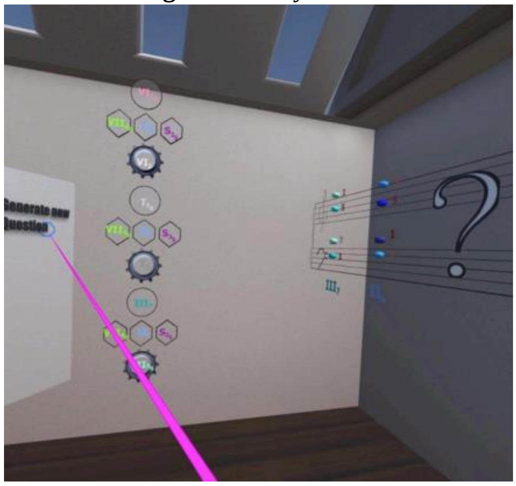


Figure 10. *Test Room.*

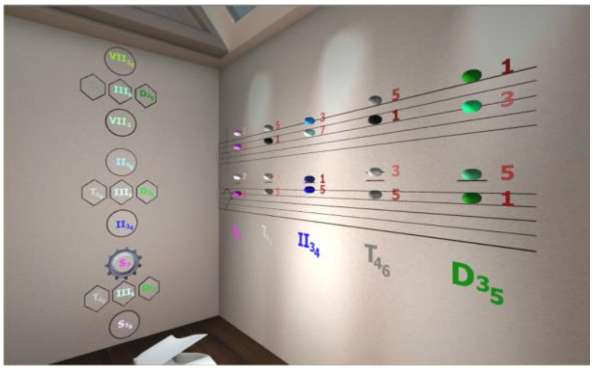


Figure 11. *Practice Room.*

The user experience begins in the *Entry* room, where the additional information about application can be seen using the *About* button; it also allows entering the *Temple of Knowledge* space using the *Start* button.

The *Temple of Knowledge* contains the explanations to provide the basics of music theory necessary for a better learning experience. This VR space contains 8 walls, 7 of which are occupied by a singular explanatory card.

The *Study Room* presents an interactive system of graphs with two modes of content exploration: *Horizontal Triad* and *Vertical Triad*. Both modes allow the user to choose either between guided and free learning experiences.

During guided learning experience, the graph element for interaction is highlighted by an animation (shown in Figure 12). If the user chooses this graph element, it triggers the augmented score appearance, the sequence audio playback and two animations: chords appearance on the staff and inside the system of graphs (both synchronized with the audio). In the guided learning experience, the interaction elements of the graph are put in a defined order and promote the discovery of all harmonic sequences from the chosen graph triad.

In case of a free learning experience, user chooses the interaction elements of the graph independently. The same mode allows to repeat a sequence as many times as needed. A free learning experience is recommended after completing guided learning experience.

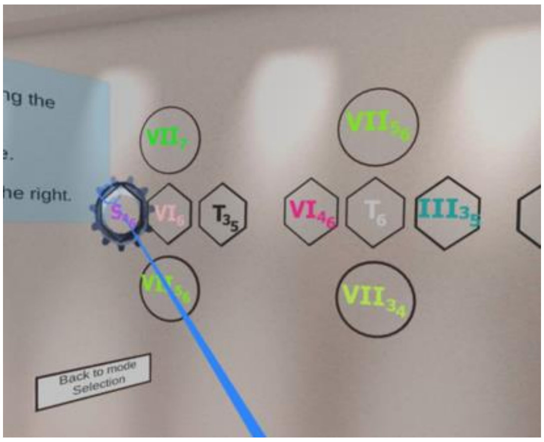


Figure 12. Highlighting edges to interact with in the *Horizontal mode*.

Horizontal Triad mode allows a horizontal graph exploration, allowing the discovery of potential for variability in passing chords (the edges in the structure of the graph). User can choose to return to a main view in order to choose another horizontal triad or switch modes at any time during the learning experience.

Vertical Triad mode gives the ability to explore the entire vertical triad, revealing the potential for harmonic variability for the seventh chords and their progressions (tops in the graph structure). The user still interacts with the graph edges to trigger the score appearance, but the choice of the edge must be made in a graph of interest inside the vertical triad. As with the horizontal triads, the user can choose to return to the main menu at any time.

The spatialization of the sound is integrated to reinforce memorization of the sequences of chords, since each chord finds its distinct spatial placement in a graph structure.

5. NOVEL DATA ENCODING

The value of the graph representation consists not only in the systematization of knowledge representation, useful in a pedagogical context, but also for the encoding of semantic music data. The main interest of such encoding lies in a possibility of training and evaluation methods application, inherent to the visual domain. This way, harmonic sequences features learning may be done with the use of 2D convolutional LSTM layers, discussed below. Another advantage of such encoding is the possibility of considerable data augmentation with the application of a small variance term during the normalization process.

To obtain the feature maps out of the arrays of harmonic sequences, the following transformation steps must be performed:

1. Chords dictionary creation.
2. Two-dimensional matrix creation using the graph system frame and replacing the chords with the values from previously created dictionary.
3. Creation of 28x28 matrices of harmonic paths in a 2D space using harmonic sequences mapped to the chords dictionary in a progressive way: one sequence of 5 chords results into 5 matrices, gradually filled in with the chord values.
4. Normalization of the feature maps with a changing normalization term (in a range between 0.01 and 0.1).

Using this data conversion strategy, it was possible to obtain 2160 data entries out of the initial handcrafted 216 harmonic sequences.

6. PREVIOUS MODELS

The main restriction of this project consists in the obligation of generating diatonic harmonic sequences in a C key only, since the mapping of 3D objects is made exclusively for this tonality. Therefore, the usage of pretrained models on multi-tonal music examples with possible alterations was not an option. Hence the need for developing and training own AI models to support the activities in a *Test Room* and *Practice Room*.

For the *Test Room* a simple long short-term memory (LSTM) deep neural network architecture with one embedding layer, two LSTM layers and two dense (linear) layers was build, using *Keras* framework. The model had 36,196 trainable parameters. The activation function of the LSTM layers was hyperbolic tangents (default *Keras* settings), the activation function of the first dense layer was ReLU (Rectified Linear Unit) and of the second dense layer – softmax. Adam optimizer with learning rate of 0.001 was applied as well. The training was done with 500 epochs with the batch size of 9 sequences. The autoregressive nature of the model allowed predicting the 3rd chord of the sequence, given two previous chords as an input. The architecture of the model is given in Figure 13.

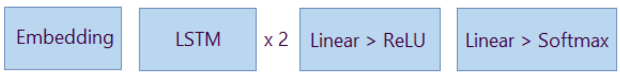


Figure 13. LSTM.architecture

For the *Practice Room*, a conditional generative adversarial (GAN) networks was developed using *Pytorch* deep learning framework. Conditional GAN model consisted of convolutional layers (transposed 1D convolution for the generator and 1D convolution for the discriminator) and was generating 4 chords given the first chord as an input. The number of input classes was equal to 7, representing 7 degrees of the diatonic scale. The architecture of the model is presented in Figure 14.

The training data for both models were partially handcrafted, partially taken from a Kaggle dataset *Classical Music*. The harmony information extraction was done with the *Music21* Python library application [25], and further tokenization and features extraction – using *Pytorch* framework tools.

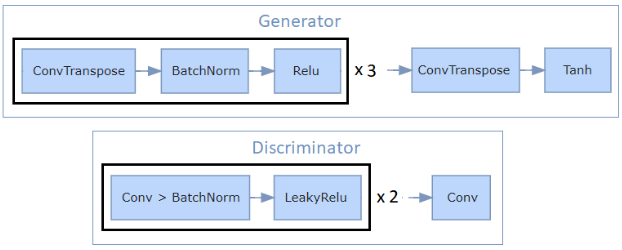


Figure 14. Conditional GAN architecture.

Trained models were stored in a cloud using *Flask* framework for performing inference. *JSON* protocol was used to exchange data between models and the VR application.

7. EXPERIMENT WITH THE NOVEL DATA ENCODING SCHEME

A new data encoding scheme have been tested in unsupervised setting, for which a ConvLSTM model was developed. The model's architecture comprised a doubled stack of convolutional 2D LSTM layers, followed by 3D batch normalization (third dimensionality was necessary to account for time steps information). The tangent activation function was used for gated mechanism inside a ConvLSTM cell. The sigmoid activation function was applied to the output of the final convolutional 2D layer, generating prediction on the input batch. The entire model architecture is shown in Figure 15.

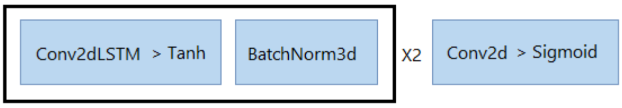


Figure 15. ConvLSTM model architecture.

Measurements of the 2D encoding scheme efficiency is therefore made by comparing the ConvLSTM model, trained on 2D data and the LSTM model, trained on the equivalent 1D data. For the sake of consistency, the previous LSTM model was rewritten in *Pytorch* framework.

The similar hyperparameters were applied to both models, such as mini-batch size (15 sequences per batch), optimizer (Adam), learning rate (1e-4), sequence length (5 chords in a sequence), number of epochs (100 epochs). The loss function for LSTM and ConvLSTM layers were cross entropy and binary cross entropy respectively.

The validation strategy for the received result was k-folds with the k value equal to 10, meaning that for each network architecture, 10 models were trained on different parts of the training and validation splits.

One-dimensional data consisted of 216 sequences of 5 chords each, where 4 chords served as an input and the 5th chord was treated as a target. In two-dimensional data split, each chord of the sequence was converted into a 28x28 matrix form, where the first 4 matrices represented the input features and the last matrix represented a target. This way, four input chords conditioned the prediction of the final chord of the sequence.

7.1 LSTM vs ConvLSTM

The result of the comparison between two models (presented in Table 1) shows that although LSTM model had smaller loss values, comparing to the ConvLSTM model, the difference in the end of training between train and validation loss for the LSTM model augments, compared to the beginning of training. Moreover, the loss of the LSTM model for the validation split becomes bigger in the end of training, which points to overfitting problem and a feeble generalization capacity. On the contrary, ConvLSTM model, having started with bigger loss values in the beginning of training, ends up with times smaller loss values. The data used for training and validation were not augmented at this stage.

Model	Train start	Val start	Train end	Val end
LSTM	0.26	0.29	0.10	0. 37
ConvLSTM	10.10	53.60	0.37	0.58

Table 1. Comparison of data dimensionality augmentation tested with LSTM and ConvLSTM models.

7.2 Data augmentation

The second stage of the experiment intended to measure the efficiency of data augmentation, made possible with the varying normalization term, using the novel encoding method. ConvLSTM model was therefore trained on datasets with and without data augmentation (2160 vs 216 data entries). The result of this stage of the experiment is shown in Table 2.

Data type	Train start	Val start	Train end	Val end
Non-augmented	10.10	53.60	0.37	0.58
Augmented	0.06	1.56	0.01	0.38

Table 2. Comparison of the data augmentation tested with ConvLSTM model.

The application of data augmentation technique has shown much better results in all four columns of the table – the first feedforward hidden representation is considerably better for the model trained on augmented data, which ameliorates the validation result at the beginning of training. In the end of training, the loss for train and validation splits substantially diminishes, meaning that the model learned well the input data shape, especially compared to the loss for the model trained on non-augmented data.

8. CONCLUSIONS

The article presented the use of a graph representation application in both ways: as knowledge representation method in a context VR application for ear training and as a novel data encoding technique applied to the generative models training. The novel encoding scheme was tested with ConvLSTM model, designed to process two-dimensional sequential information, being compared to the LSTM model with the similar hyperparameters, trained on the equivalent one-dimensional data. The results of the experiment have shown an important training result amelioration, especially when the data augmentation technique was applied. Finally, a high relevance of the design patterns used in a current VR project has been demonstrated via comparison with the recent development in the field.

7. REFERENCES

[1] M. Minsky, "A framework for representing knowledge," in P. Winston (Ed.), *The psychology of computer vision*, New York: Winston, 1974, pp. 211-277.

[2] A. Shvets and J. Jemielnik, "Visualization Strategies in the E-learning Course on Music Harmony at University – Effects and Possibilities," in Andrzej Michalski (Ed.), *Music Pedagogy. Ideals. Values. Pragmatics*, Gdansk: Athenae Gedanenses, 2016, pp. 218-229.

[3] A.S hvets, "Contemporary methods of functional harmony teaching in a high school context," in *Proceeding of Electronic Imaging & the Visual Arts (EVA Florence),* Florence, 2019, pp. 142-150.

[4] A.S hvets, "The system of graphs in music harmony: a user interface for mobile learning game development," in *Proceeding of Electronic Visualisations and the Art (EVA London),* London, 2016, pp. 193-194.

[5] A.S hvets and S. Darkazanli, "Graphs in harmony learning: AI assisted VR application," in *Proceeding of Electronic Visualisations and the Art (EVA London,* London, 2020, pp. 104-105.

[6] O. Zawacki-Richter, V. I. Marin, M. Bond and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education–where are the educators?" in *International Journal of Educational Technology in Higher Education, 16(1),* pp. 1-27.

[7] A. Lino, A. Rocha and A. Sizo, "Virtual teaching and learning environments: automatic evaluation with artificial neural networks," *Cluster Computing*, 22(3), 2019, pp. 7217-7227.

[8] B. Zhang, H. Xia and H. O. Lim, "Development of an A.I. based teaching assisting system," in *International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, Sanya, China, 2019, pp. 1-5.

[9] W. Villegas-Ch, A. Arias-Navarrete and X. Palacios Pacheco, "Proposal of an Architecture for the Integration of a Chatbot with Artificial Intelligence in a Smart Campus for the Improvement of Learning," *Sustainability 2020, 12(4):1500*, https://doi.org/10.3390/su12041500

[10] L. Turchet, R. Hamilton and A. Camci, "Music in Extended Realities," in *IEEE Access*, *9*, 2021, pp. 15810-15832.

[11] H. Liu and Y. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 722-727.

[12] Y. Yu and S. Canales, "Conditional LSTM-GAN for melody generation from lyrics," in *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 17*, 2021, pp. 1-20.

[13] Z. Wang, D. Wang, Y Zhang, Y. and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR 2020)*, Montréal, 2020, pp. 662-669.

[14] H. H. Tan and D. Herremans, "Music FaderNets: Controllable music generation based on high-level features via low-level feature modelling," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR 2020),* Montréal, 2020, pp. 109-116.

[15] S. Lattner, M. Grachten and G. Widmer, "Imposing higher-level structure in polyphonic music generation using convolutional restricted boltzmann machines and constraints," in *ArXiv*, *abs/1612.04742*, 2016.

[16] A. Pati and A. Lerch, "Is disentanglement enough? On latent representations for controllable music generation," in *ArXiv, abs/2108.01450*, 2021.

[17] A. Pati, S. Gururani and A. Lerch, "dMelodies: a music dataset for disentanglement learning" in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR 2020),* Montréal, 2020, pp. 125-133.

[18] D. Makris, K.R. Agresand and D. Herremans, "Generating lead sheets with affect: a novel conditional seq2seq framework," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, 2021, pp. 1-8.

[19] H. Liu and Y. Yang, "Lead sheet generation and arrangement by conditional generative adversarial

network," in *Proceedings of the 7th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Florida, 2018, pp. 722-727.

[20] A. Shvets and S. Darkazanli, "Conditional GAN for Diatonic Harmonic Sequences Generation in a VR Context" in *Proceeding of EVA London 2021*. British Computer Society (BCS): London, 5th–9th July 2021, pp. 97-100.

[21] S. Li and Y. Sung, "INCO-GAN: variable-length music generation method based on inception model-based conditional GAN," in *Mathematics 2021*, 9*(4):387*, https://doi.org/10.3390/math9040387

[22] S. H. I. Xingjian, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting" in *Advances in neural information processing systems*, 2015, pp. 802-810.

[23] A. Shvets, "Structural harmony method in the context of deep learning on example of music by Valentyn Sylvestrov and Philipp Glass," in *Proceeding of EVA London 2020*, London, 2019, pp. 318-320.

[24] A. Shvets and M. Desainte-Catherine, "Schemographe: Application for a new representation technique and methodology of analysis in tonal harmony" in *Proceedings of International Conference on Evolutionary and Biologically Inspired Music and Art*, Copenhagen, 2016, pp. 212-223.

[25] M. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proceedings of International Conference on Music Information Retrieval*, Utrecht, 2010, pp. 637-642

RETHINKING THE NOTATION DESIGN SPACE

Benjamin Bacon
Technische Universität Berlin
Berlin, Germany
benjamin.bacon@campus.tu-berlin.de

ABSTRACT

Previous work has demonstrated how the analysis and creation of musical notation can be seen within the context of information visualisation. In this case, graphical and musical features are broken down into primary categories which can then be linked to one another, allowing for the visualisation of notation mapping schemes. The space for mapping these elements is known as the Notation Design Space (NDS). While the NDS has the potential to be a powerful tool for analysing and creating new notations, the current model does not provide adequate support for notations which depict the actions of the performer. This paper proposes changes to the current NDS to include the mapping of sound-producing and -facilitating gestures, followed by a theoretical analysis of the similarities between notation and digital musical instrument mapping. The inclusion of musical gesture within the NDS serves a dual purpose; it allows for a more nuanced reading of prescriptive-based notation focusing more on the actions of the performer, while also aligning the development of new notations with interaction design processes.

1. INTRODUCTION

The process of designing new music notation can be seen as an area of constant innovation, experimentation, and compositional creativity. Aided by advancements in music technology, novel notation designs and modes of representation offer countless opportunities for interacting with musical concepts, ranging from animated or screen-based scores [1], live coding [2], and three-dimensional/virtual scores [3].

While there are numerous avenues for compositional exploration in developing and working with new notations, there are equally many design-based decisions to be made, each with their own trade-offs. In such cases, notation systems balance the structural complexity of their design with the cognitive bandwidth of the user, giving importance to the transparency of when and where these design-based decisions are made [4]. For example, animated notations often have a fixed reading in relation to the time domain, while interactive augmented/virtual reality notations require a certain degree of freedom-of-movement from the reader, which necessitates access to the necessary technology. Even simple graphic constructs such as line thickness and proximity can collaboratively produce additional entities and unintended visual clutter [5]. As such, any notation design choice has the potential to affect the creative pathways of a given work's development, as well as the reader's ability to engage with the score.

The complexity of designing a new notation framework helps explain why some have turned to the field of information visualisation as a resource for better understanding the notation development process. Information visualisation can be defined as the graphical representation of data and/or concepts [6]. For example, a map may employ several techniques in representing the elevation of a specific location, ranging from specific use of colour, shading, texture, and saturation. Previous work has shown that musical notation can share many similarities with cartography, allowing for detailed readings of notation based on instrumental topologies as well as bridging notation with the many information layering techniques found in map making [7]. Other works have investigated the perception of colour within notation [8], and the inclusion of information design principles for notational clarity [9]. When looking closer into human computer interaction (HCI) research, music notation has been used as an example for demonstrating cognitive analysis methods of information structures [10] as well as perceptual clarity in relating to information layering [11].

To aid the process of developing new notations, the Notation Design Space (NDS) was introduced to specifically bridge information visualisation analysis methods to this interdisciplinary practice [12]. By doing so, the NDS provides a method for exploring new notation mapping strategies, as well as analysing the designs of existing notations. But as this paper suggests, the current organisational structure of musical features in the NDS does not provide adequate support for many musical notations which depict the actions of the performer.

2. THE NOTATION DESIGN SPACE

The Notation Design Space can primarily be seen as a tool for tracing the mappings of visual channels to musical features. By viewing notation through the lens of information visualisation, the exact design features of a system can be examined. Thus, any notation mapping scheme can be broken down into its constituent parts, allowing for more helpful critique of specific design choices while also allowing for the consideration of new design opportunities through the exploitation of unused visual/musical pathways. The NDS differs from other research on information visualisation in music, as it focuses strictly on the design of notation itself (i.e. performer-centric), rather than the general visualisation of musical data for analysis or information retrieval purposes [13].

2.1 Notation Design Space Organisation

In developing the NDS, Miller et al. turned to some of the primary organizing principles of graphic design, visual perception, and musical composition. On the visual side, the NDS is broken down into various visual channels initially developed by Bertin [14] and furthered by Munzner [15], as well as Gestalt Laws for image patterns and groupings, and finally the use of text within the semantic channel. Therefore we have the following primary visual groupings seen in Figure 1 accompanied by examples.

Semantic Channel
Text
Magnitude Channel
Position on common scale
Position on unaligned scale
Length (1D)
Tilt / Angle
Area (2D size)
Depth (3D position)
Color Luminance
Color Saturation
Texture
Curvature
Volume (3D size)
Identity Channel
Spatial Region
Color Hue
Shape
Motion
Gestalt Laws
Proximity
Similarity
Enclosure
Closure
Continuity
Connection

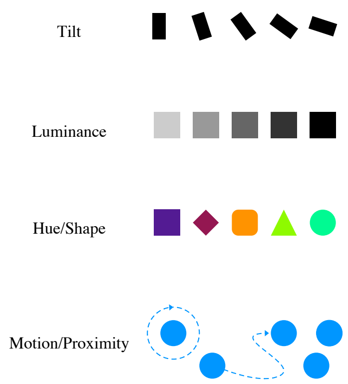


Figure 1. The original visual categories found within the Notation Design Space by [12] on the left, with specific visual examples of *tilt*, *luminance*, *hue/shape*, and *motion/proximity* on the right.

Each of the major visual meta-categories are separated by their perceptual characteristics and semantic function. The *Identity* (what or where) and *Magnitude* (how much) channels are two fundamental sensory modalities of human perception, while *Gestalt Laws* pertain to perceived patterns and spatial relationships between entities. The *Semantic Channel* relates to the real-world meaning of information and its underlying cultural references.

Within the musical fields of the NDS, the authors have generated a list of musical meta-features based on their own analysis and research. These features are broken down in accordance with their prevalence and use within traditional music notation, or as referred to by the original authors as Common Music Notation (CMN) with regards to the NDS. As such, the musical side of the NDS has four major groups seen in Figure 2.

The original design of the NDS included *duration* within the harmony meta-category due to the influence of note duration on harmonic progression. While this conclusion could be debated from a categorical standpoint, it does not affect the overall functionality of the NDS, as duration could easily be moved to the rhythm section without

Rhythm
Tempo/Beat
Meter/Time Signature
Pauses/Breaks
Harmony
Tone (Pitch/Frequency)
Note
Range
Accidental/Normal
Duration
Dynamics
Intensity/Volume
Articulation
Phrasing
Instruction
Timbre/Instrumental
Arrangement
Baseline/Clef
Finger

Figure 2. This list contains the original musical categories found within the Notation Design Space by [12].

significantly affecting the overview of a given mapping scheme. As a whole, the initial version of the NDS was designed particularly with the CMN framework in mind, which prioritises the features of traditional western musical aesthetics (e.g. quantised pitch and duration values). This will be further elaborated upon in the following sections of this paper. Due to this association, notations which seek to engage with musical concepts which go beyond the CMN structure are not as easily supported. This has led to critiques of the impact CMN has had on both the performance-based constraints and analytical capabilities of the evolving aesthetics in new music, much of which is influenced by music technology [16].

2.2 Current Limitations

In terms of its design, Common Music Notation can be seen as a visual structure for the representation of western musical concepts. The cultural foundations behind this system are strong enough that expert musicians are able to internally hear how a given piece of music sounds by reading the score itself [17]. Therefore it is not uncommon to find CMN referred to as the representation of the intended sound of a piece of music, while action-oriented methodologies focus on the sound-producing actions of the performer [18]. In its present form, the NDS contains underdeveloped room for incorporating notations which do not rely on the primary pitch and duration features of CMN. The *instruction* meta-category of the NDS is where gestural actions should be more accurately described, but the current features of *finger, baseline/clef, arrangement,* and most importantly *timbre,* unnecessarily compress the notated actions found in pieces employing prescriptive notation strategies.

Since many new compositions contain highly complex mappings of specific actions between the performer and their instrument, expanding the *instructions* meta-section could significantly increase both the analytical and creative utility of the NDS. By reviewing approaches to complex timbral interaction in contemporary music, and adopting techniques for representing gesture in digital musical instrument design, more precise analytical categories relating to music performance can be added to the NDS, thereby allowing more nuanced and specific readings of current notation design schemes and potential openings for new ones.

3. TIMBRE AND ACTION

Timbre itself has an almost notorious reputation when it comes to defining its meaning, much less representing it visually [19]. When considering the direct visual depiction of timbre to represent an intended sound, cross-modal sound-form symbols show an apparent link between auditory and visual perception [20], but there remains much to be discovered in this area. Within the context of instrumental notation, visually representing timbre can be seen as somewhat of an issue from a design perspective, as its assignment to any feature of a devised system runs the risk of embracing unintentional ambiguity. Common Music Notation is able to visually encode complex musical ideas because it is structured around historical musical criteria (e.g. quantised pitch, rhythm) which in turn possesses wide-spread cultural understanding. This makes the encoding and visual representation of newer musical criteria difficult to achieve, especially those advanced by electronically produced or altered sound. In the following section, we will examine the structure of CMN from a visual design standpoint. From this position, the functional grounding of notation from a prescriptive and descriptive perspective may shed light on techniques for representing timbre.

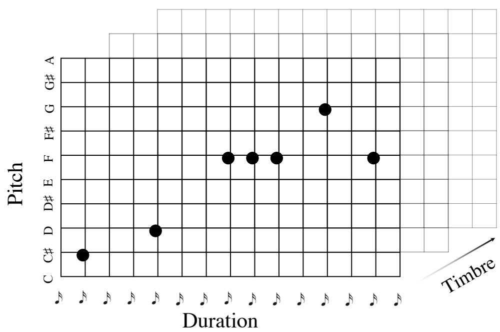


Figure 3. A schematic representation of the lattice structure of Common Music Notation as described by Trevor Wishart in [21]. Timbre is shaped by layering instrumental parts. Dynamics are not a primary component of the lattice.

The Lattice Structure

Music theorist and composer Trevor Wishart presents an analytical strategy which may help us understand the representational challenges faced by Common Music Notation [21]. Through his analysis of western notation, we find that CMN can be described by the musical concepts which it prioritises in depicting. In turn, we can reduce its structure to a lattice of specifically quantised pitch and duration values seen in Figure 3. The lattice structure was not designed to support the notation of complex shifting timbres due to its historical development. For the majority of CMN's existence, musical instruments and the sounds they produced were intrinsically linked, but this is challenged by the advent of electronic musical instruments which introduces interfaces, synthesis engines, and mapping into the compositional sphere. While electronic musical instruments can completely change their sonic characteristics with the push of a button (e.g. morphing from the sound of a clarinet to the voice), the lattice-based CMN assumes fixed instrumentation. In other words, acoustic instruments cannot completely adopt new sonic properties beyond their physical limitations.

Wishart goes on to claim that the support structure of the CMN lattice has influenced which kinds of musical practices have succeeded within the academy itself, namely, forms of music which conform to the measurable and analytical values of CMN, thus excluding more improvisatory traditions. Overall, CMN enforces a specific historical set of musical values based on western tuning, scales, harmony, and rhythm, thus rendering itself more difficult to use when exploring techniques and concepts which fall outside their scope.

3.1 Descriptive and Prescriptive Notation

Understanding the nature of what the reader is being asked to interpret can help us contextualise the functional aspects of a given music notation system. Works which explore complex timbral development as part of their compositional grammar (which usually lie outside the traditional pitch/rhythm paradigm of CMN) can access those sounds through the depiction of the gestures/actions which produce them. Notation strategies which represent instructions of this kind can be referred to as *prescriptive* notation, while notations (commonly CMN-based) which depict the intended sound of a composition can be referred to as *descriptive* notation [23].

For example, when examining the score of Peter Swendsen's *Nothing that is not there and the nothing that is* (2009) [22] in Figure 4, the prescriptive/descriptive framework allows us to unpack the graphic components of the notation. The score displays a series of actions taken upon the surface membrane of a concert bass drum, where explicit visual-graphical channels are mapped to sound producing gestures. The notation itself can be seen as a set of instructions for the performer to interpret, resulting in a realisation of the piece. In other words, the score represents the methods for producing the sounds of the piece. Furthermore, one can also observe various visual channels as seen in Figure 1 directly mapped to musical parameters. In this case, combinations of motion, luminance, and curvature indicate gestural movements as being either smooth, rough, chaotic, or symmetric in nature. For the electronic accompaniment (which includes live processing), graphical texture is mapped to articulation profile and dynamic envelope so that the performer may keep their place in the score. Lastly, text-based semantic channel information is presented above each segment, providing the performer with technical descriptions regarding mallet choice, tempo, and further phrasing instruction.

When considering prescriptive and descriptive notations,

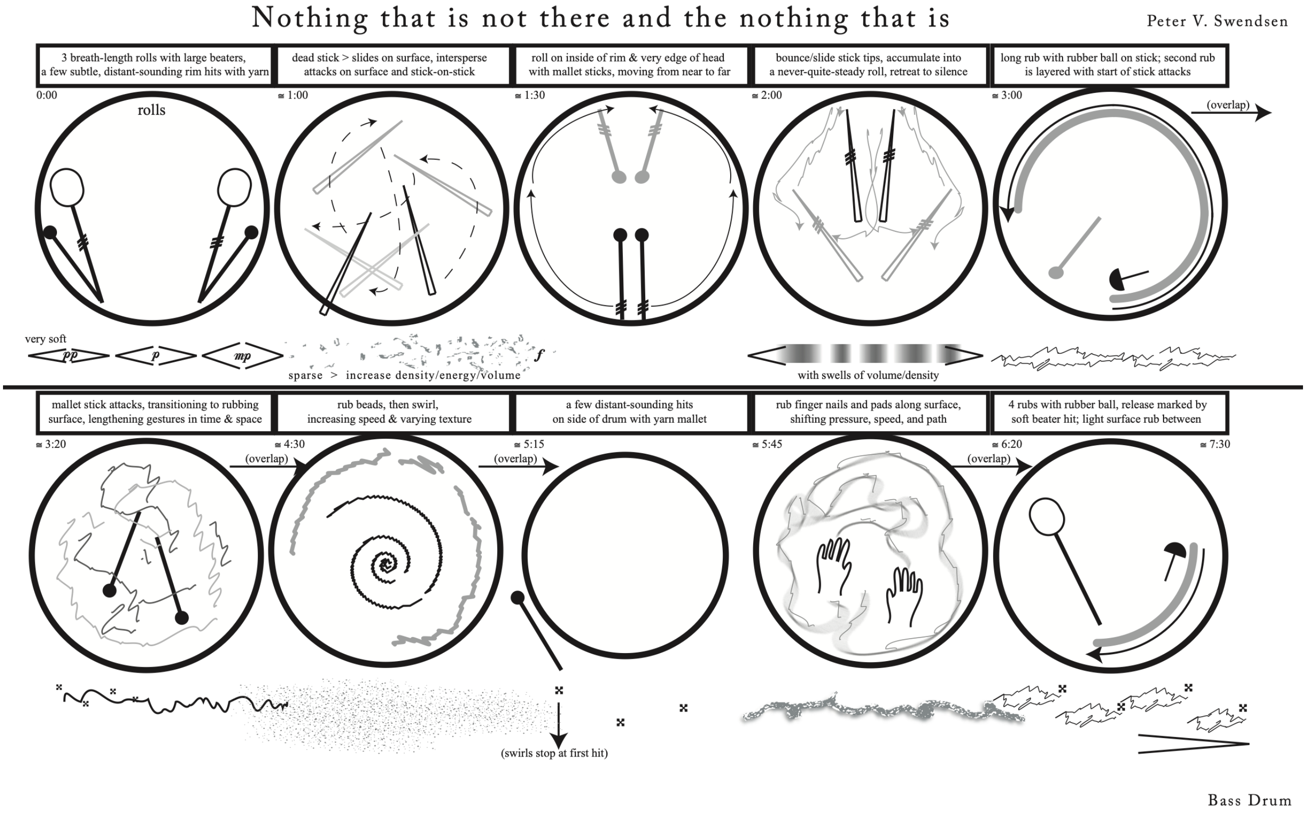


Figure 4. An excerpt from Peter Swendsen's *Nothing that is not there and the nothing that is* (2009) [22] demonstrating detailed prescriptive notation for a percussionist and electronics.

it should be stressed that this distinction is not purely dichotomous; many CMN scores contain both prescriptive and descriptive elements including fingering positions on string instruments, or mallet changes for a percussionist. Rather, the difference between prescriptive and descriptive notations helps us evaluate the economy of graphical space employed, including mappings between visual and musical features. In contemporary music, prescriptive notation has been a useful strategy for engaging with new extended techniques, music technology, and choreographed movement. These are complex forms of interaction which can stretch the utility of CMN, but ones which prescriptive-oriented designs handle well, and which stand to benefit from being incorporated into the Notation Design Space.

4. RE-EVALUATING THE NOTATION DESIGN SPACE

Given the complexity of mapping timbre to graphical features in a notation scheme, the field of instrument design and gestural analysis offers insight as to how the actions of the performer could be integrated for prescriptive designs. Digital Musical Instrument (DMI) research has provided those interested in designing interaction within music with a wide range of tools for understanding how meaningful experiences are created, mapped, and analysed [24]. The research within the fields of HCI and DMI face similar issues, especially in light of third-paradigm HCI research, which promotes the values of contextual use and creating spaces for making meaning [25]. The goals of the NDS have much in common with DMI research as well, especially in conceptualizing interaction mapping as a design process.

4.1 Performance Gestures

Digital Musical Instrument research has long been focused on developing meaningful musical interactions with technology via artificially mapped connections, lending special importance to the topic of gestural analysis. Jensenius et al. [26] offer a resource for breaking down musical performances on a gestural level, revealing the many layers of expressive and performative actions which arise out of playing a musical instrument. In their analysis, each action of the performer can be evaluated contextually, allowing one to observe gestures related to those which directly produce sound, those which facilitate the sound, and those which accompany musical expression elicited by the performer. These main categories of gestural interaction can been seen in Table 1.

With this inclusion, we have a well articulated set of gestural behaviours which tie directly to musical interaction, and thus could be adapted for performance analysis and de-

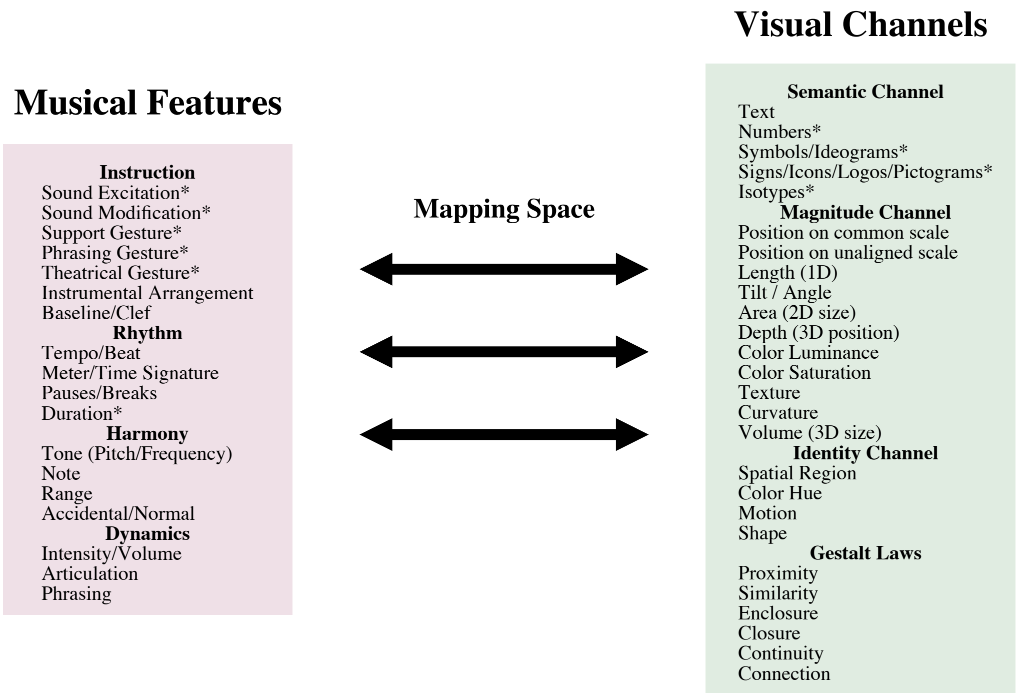


Figure 5. A new instantiation of the Notation Design Space which includes performance gestures and more semantic channel categories. These additions are marked with an *. The mapping space is where musical features are assigned to visual constructs in a given notation.

Producing	Facilitating	Communicative
Modification	Entrained	Expressive
Excitation	Support	Theatrical
	Phrasing	

Table 1. Performance gestures as seen in Jensenius et al. [26].

signing notation mapping schemes. It should be noted that not all of these gestural categories are assumed to be directly mappable in an interactive sense. Sound-producing gestures offer the most direct route to mapping, as they relate to specific aspects of controlling the sound of an instrument. Most communicative gestures are generally thought of as artefacts of emotive behaviour. Sound-facilitating gestures lie perhaps somewhere in between where phrasing and supportive gestures lend themselves to being mapped as a feature of shaping the sound of a performance, while entrained gestures (e.g. foot tapping) are less so.

Embodied cognition research may also prove to be fertile ground for the analysis of gestural mapping within the context of notation. A possible downside of visually-based analyses of gestural interaction is that they run the risk of compressing the multivariate nature of dynamic human movement in ways which could be problematic regarding the corporal agency of the performer [27]. Such notions are outside the scope of this paper, but warrant consideration for further NDS analysis.

4.2 Gesture within the Notation Design Space

By adopting gestural analysis methods from DMI/HCI research, we can integrate the relevant categories into the NDS, thus providing a more inclusive space for prescriptive notation design. By specifically adding the features of sound-producing gestures, we begin to have a much more inclusive space for notation mapping analysis. Some sound-facilitating gestures may also be possible for a given mapping, as could a theatrically-oriented communicative gesture. Therefore, a newly developed instructions segment of the NDS can be seen in Figure 5.

The inclusion of gestural categories into the instruction section of the NDS allows for the *timbre/instrument* category to be removed, as it is assumed that these gestural descriptors will offer a more precise reference of *what kind* of instrument-related action is taking place. In addition, the category of finger position can be incorporated into the *sound excitation/modification* gestural category.

With more appropriate descriptions of gesture within the NDS, compositions of prescriptive notation such as *Guero* (1969) by Helmut Lachenmann [28] or *2nd String Quartet* (2010) by Aaron Cassidy can be analysed with a higher level of transparency in their graphical mapping schemes. Cassidy's technique in this example makes explicit use of the various visual channels mapped to sound-producing or modifying actions [29]. Other pieces, such as *Shiver Lung* (2019) by Ashley Fure [30] or *Nothing that is not there and the nothing that is* (2009) by Peter Swendsen [22] in Figure 4, employ combinations of pictograms and other visual channels as a prescriptive notation due to the complexity of electronically-driven elements found within the piece.

Descriptive-based scores also stand to benefit from the inclusion of gestural action-based elements within the NDS, as these two main categories of score design are not mutually exclusive. Descriptive scores may contain within themselves prescriptive elements and vice versa [31]. A more developed NDS offers a larger analytical space for these schemes, which opens the door for new mappings. As the original authors of the NDS have stated, there remain many unexplored avenues when considering notation mapping schemes, many of which may have to do with the innate perceptual affordances and drawbacks of a graphical entity's ability to represent musical information. In this regard, a connection to perceptual graphics may offer useful insight for further study [6].

4.3 Expanding the Semantic Channel

By including gesture within the NDS, we can see how the semantic channel of the original design could also be extended to contain elements beyond just text. Many notation mapping schemes found in contemporary music employ the use of various glyphs, which can be seen more broadly as bundled collections of visual channels used to produce a visual sign [32]. Fragmenting glyphs into their various graphical parts makes understanding their contextual role within a score more difficult. Therefore, the inclusion of *numbers*, *symbols/ideograms*, *signs/icons* groups, and *isotypes* [33] into the semantic channel of the NDS could provide for a richer mapping space as well.

5. MAPPING INTERACTION AND NOTATION

Within the field of DMI design, interaction mapping is a core area of study, fostering further connections to HCI research. In research focusing on embodied cognition in music performance, scores and notation can even be seen as having instrumental properties, serving as intermediaries by which expressive musical interactions take place [34]. By further developing a process of analysis which connects notation to mapping, a framework of study exploring how particular mappings are successful in relation to their intended goals can also be developed. Through the lessons of DMI research, music notation design can be linked as a kind of parallel process to the interactive mapping of musical instruments.

5.1 Leveraging Familiarity

Borrowing interactive mappings from other contexts when designing a new instrument provides the player with an instantly familiar set of performance actions, leading to instant music making and performance feedback [35]. For instance, one can conceive of a musical instrument which uses the QWERTY keyboard as its interface. Most people would quickly be able to navigate this instrument with a well developed set of gestures stemming from their previous experiences with the interface [36]. A similar process of borrowing from familiar graphical schemes found in other contexts could be explored in notation design as well. For example, colour mapping schemes from cartography manage to successfully convey detailed information mappings to the reader. The bivariate choropleth allows for multiple complex readings of diverging or sequential values [37]. Such colour mapping schemes could be adapted and applied in the context of notation, offering a tried and true visual mapping strategy for colour which is demonstrated in Figure 6.

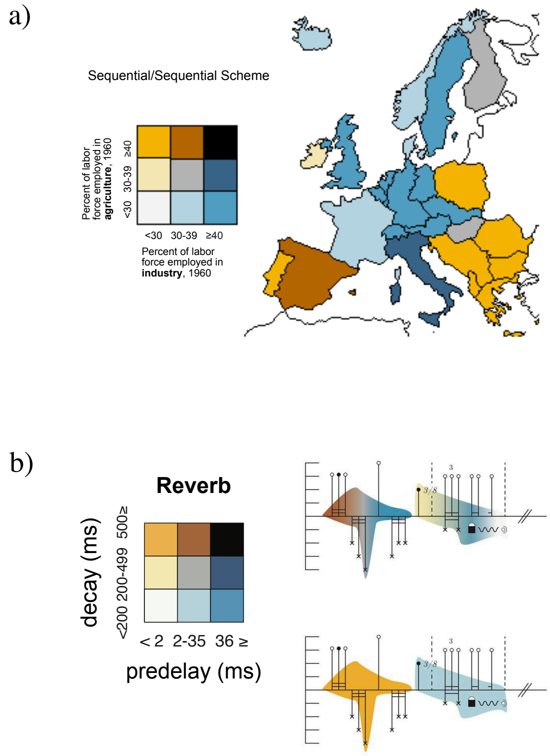


Figure 6. Two examples of bivariate choropleth colour mappings. Image *a* as seen in [37]. Image *b* by the author, employing the same colour scheme remapped to audio parameters as seen in an étude [38].

Mapping design tools from DMI research could also be reviewed for their potential use in analysing the mappings of new notations. Interfaces for tools such as Libmapper can be studied with regards to their ability to visualise mapping connections, thus aiding the work-flow in developing new designs [39]. In contemporary music, notation design is a highly interdisciplinary act guided by a wide array of influences specific to the experiences and interests of each composer. As of yet, this process does not have comparable mapping tools to those found in DMI research. When specifically considering prescriptive notation as a form of instruction, the link between interaction design and notation has the potential to spur further research in both creative and empirical contexts.

5.2 Layering and Separation

With a more developed design space for notation, the design choices of any system can be viewed with regards to their visual channels, thus leading to a more directed use of visual elements and bringing the process of notation design closer to the realm of information graphics. The layering of information is a delicate but essential process in the design of any information display [11]. How visual channels are employed can be difficult to manage, thus highlighting how the NDS could be used creatively. Through an expanded NDS, composers of both traditional and experimental styles can determine which visual channels are free and which ones are used, allowing for a critique of a given graphic layering scheme. Information layering carries with it both aesthetic and scientific implications, making the careful and considered use of any visual channel important for both the composition and the performer.

6. CONCLUSIONS

As an analysis tool, the Notation Design Space can help us understand mappings which work, those which do not, and those which are unexplored. The primary issue with its current form is the primary focus on historic musical features guided by Common Music Notation. Today, many new pieces found in contemporary music engage with diverse range of graphical elements to communicate performance gestures with the performer, thus speaking to the need for relevant analytical design tools. The expansions to the NDS suggested in this paper also aim to provide stronger links between notation design, digital musical instrument research, and information graphics. Gestural interaction can be seen as a core focus of many new works today. Therefore, offering a more inclusive design space could be beneficial both in terms of ideation and analysis.

Acknowledgments

Thanks to the kind support and feedback from my colleagues at TU Berlin. An additional thanks to Peter Swendsen for providing an excerpt of his score. This paper stems from a series of workshops conducted at the Oberlin Conservatory of Music in the Autumn of 2021.

7. REFERENCES

[1] C. Hope, A. Wyatt, and D. Thorpe, "Scoring an animated notation opera–the decibel score player and the role of the digital copyist in'speechless'," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'18)*, vol. 18, Montreal, Canada, 2018, pp. 193–200.

[2] T. Magnusson, "Algorithms as scores: Coding live music," *Leonardo Music Journal*, vol. 21, pp. 19–23, 2011.

[3] 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, pp. 33–38.

[4] A. F. Blackwell, C. Britton, A. Cox, T. R. Green, C. Gurr, G. Kadoda, M. Kutar, M. Loomes, C. L. Nehaniv, M. Petre *et al.*, "Cognitive dimensions of notations: Design tools for cognitive technology," in *International Conference on Cognitive Technology*. Springer, 2001, pp. 325–341.

[5] J. Albers, *Search Versus Re-Search*. Trinity College Press, 1969.

[6] C. Ware, *Information Visualization: Perception for Design*. Morgan Kaufmann, 2019.

[7] D. Miller, "Are scores maps? a cartographic response to goodman," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, 2017, pp. 57–67.

[8] L. Vickery, "Some approaches to representing sound with colour and shape," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'18)*, Montreal, Canada, 2018, pp. 165–173.

[9] B. Bacon, "Tufte design concepts in musical score creation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, 2015, pp. 204–210.

[10] A. Blackwell and T. Green, "Notational systems–the cognitive dimensions of notations framework," *HCI Models, Theories, and Frameworks: Toward an Interdisciplinary Science. Morgan Kaufmann*, 2003.

[11] E. R. Tufte, N. H. Goeler, and R. Benson, *Envisioning Information*. Graphics Press Cheshire, CT, 1990, vol. 2, ch. 3, pp. 37–52.

[12] M. Miller, J. Häußler, M. Kraus, D. Keim, and M. El-Assady, "Analyzing visual mappings of traditional and alternative music notation," *arXiv preprint arXiv:1810.10814*, 2018.

[13] R. Khulusi, J. Kusnick, C. Meinecke, C. Gillmann, J. Focht, and S. Jänicke, "A survey on visualizations for musical data," in *Computer Graphics Forum*, vol. 39, no. 6. Wiley Online Library, 2020, pp. 82–110.

[14] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*, W. J. Berg, Ed. Esri Press, 2011.

[15] T. Munzner, *Visualization Analysis and Design*. CRC press, 2014.

[16] C. Hope, "Electronic scores for music: The possibilities of animated notation," *Computer Music Journal*, vol. 41, no. 3, pp. 21–35, 2017.

[17] W. Brodsky, A. Henik, B.-S. Rubinstein, and M. Zorman, "Auditory imagery from musical notation in expert musicians," *Perception & Psychophysics*, vol. 65, no. 4, pp. 602–612, 2003.

[18] J. Kojs, "Notating action-based music," *Leonardo Music Journal*, vol. 21, pp. 65–72, 2011.

[19] S. McAdams and B. L. Giordano, "The perception of musical timbre," *The Oxford Handbook of Music Psychology*, pp. 72–80, 2009.

[20] K. Sathian and V. S. Ramachandran, *Multisensory Perception: From Laboratory to Clinic*. Academic Press, 2019.

[21] T. Wishart and S. Emmerson, *On Sonic Art*. Psychology Press, 1996, vol. 12, ch. 2, pp. 11–43.

[22] P. Swendsen, *Nothing that is not there and the nothing that is*. PVS, 2009.

[23] M. Kanno, "Prescriptive notation: Limits and challenges," *Contemporary Music Review*, vol. 26, no. 2, pp. 231–254, 2007.

[24] S. Holland, T. Mudd, K. Wilkie-McKenna, A. McPherson, and M. M. Wanderley, *New Directions in Music and Human-Computer Interaction*. Springer, 2019.

[25] S. Harrison, D. Tatar, and P. Sengers, "The three paradigms of hci," in *Alt. Chi. Session at the SIGCHI Conference on Human Factors in Computing Systems*, San Jose, CA, USA, 2007, pp. 1–18.

[26] A. R. Jensenius and M. M. Wanderley, "Musical gestures: Concepts and methods in research," in *Musical Gestures*. Routledge, 2010, pp. 24–47.

[27] P. Antoniadis, "The ocularcentric objectification of musical embodiment in cognitive capitalism: Covid-19 as an allegory on the multiple senses of touch," March 2022. [Online]. Available: https://pavlosantoniadis.files.wordpress.com/2021/11/embodied%5Fpavlos%5Fv3.pdf

[28] H. Lachenmann, *Guero*. Breitkopf & Härtel, 1969.

[29] A. Cassidy, "Constraint schemata, multi-axis movement modeling, and unified, multi-parametric notation for strings and voices," *Search: Journal for New Music and Culture*, vol. 10, 2013.

[30] A. Fure, *Shiver Lung 2*. Edition Peters, 2019.

[31] P. Antoniadis and A. Chemero, "Playing without mental representations: Embodied navigation and the gestcom as a case study for radical embodied cognition in piano performance," *Journal of Interdisciplinary Music Studies*, 2020.

[32] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications." in *Eurographics (State of the Art Reports)*, 2013, pp. 39–63.

[33] M. Chen and L. Floridi, "Anaanalysis of information visualisation," *Synthese*, vol. 190, no. 16, pp. 3421–3438, 2013.

[34] E. Maestri and P. Antoniadis, "Notation as instrument: From representation to enaction," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, 2015, pp. 211–218.

[35] P. Cook, "2001: Principles for designing computer music controllers," in *A NIME Reader*. Springer, 2017, pp. 1–13.

[36] C. Nash, "The 'e' in QWERTY: musical expression with old computer interfaces," in *16th International Conference on New Interfaces for Musical Expression, NIME 2016, Griffith University, Brisbane, Australia, July 11-15, 2016*. Queensland Conservatorium Griffith University, 2016, pp. 224–229. [Online]. Available: http://www.nime.org/proceedings/2016/nime2016%5Fpaper0045.pdf

[37] C. A. Brewer, "Color use guidelines for mapping," *Visualization in Modern Cartography*, pp. 123–148, 1994.

[38] B. Bacon, "Notation design lab," Online, April 2022. [Online]. Available: https://bbacon.org/Notation-Design-Lab

[39] J. Wang, J. Malloch, S. Sinclair, J. Wilansky, A. Krajeski, and M. M. Wanderley, "Webmapper: A tool for visualizing and manipulating mappings in digital musical instruments," in *14th International Symposium on Computer Music Multidisciplinary Research*, Marseille, France, 2019, pp. 823–835.

IDIOSYNCRATIC WAYS OF PRESERVING PERFORMING ARTS CREATION IN AN (DIGITAL) ARCHIVE

Filipa Magalhães
Centre for the Study of the Sociology and Aesthetics of Music (CESEM),
NOVA School of Social Sciences and Humanities (NOVA FCSH)
Lisbon, Portugal
filipa.magalhaes@campus.fcsh.unl.pt

ABSTRACT

Most collections conceived in artistic domains, whether in dance, music or theatre, as they are performances and involve heterogeneous sources such as text, image, audio-video recordings, music, scenarios, gesture, movement, among others, are difficult to describe or document in archival contexts (e.g., music theatre). Archiving these works challenges musicologists, as it requires an in-depth knowledge of their collaborative practices, in addition to a study considering an archaeological musicology, being necessary to gather the pieces of the puzzle, since the different elements/materials of the works are dispersed by various sources. Post-custodial forms of archive present some solutions, however it would be important to seek for a common core language and combine archival standards in order to allow the interoperability of information to understand these works from a holistic perspective. In this paper, I seek to broaden discussions about the issues around preserving creations in the field of performing arts in the (digital) archive, giving specific examples in different artistic spheres.

1. INTRODUCTION

The performative genre music theatre associated with such names as Luciano Berio, Luigi Nono, John Cage, Mauricio Kagel, Karlheinz Stockhausen, Heiner Goebbels, Georges Aperghis, Sylvano Bussotti, György Ligeti, Constança Capdeville, Carlos Alberto Augusto and António Sousa Dias, amongst others, has not been yet properly addressed from the archival perspective. Often composers produced documentation in close collaboration with performers, with specific notes for them, and many documents are in own their custodies. Part of music theatre documentation remains dispersed over many sources or lost, obscuring the performance of improvised hypothetical components, as well as issues of idiosyncratic or non-standard notations, coupled with obsolescence and deterioration of carriers (e.g., tapes) and a continued lack of systematisation. All this hinders study and dissemination. The importance of reflecting on the key constraints encountered in recovering music theatre works incorporating different means represents an enduring challenge for archival studies. Music theatre preservation strategies go beyond the act of active listening as proposed by Alain Bonardi from a set of interactions between listeners and musical documents, or acts of closed listening (focused listening without recourse to any other source of information) or multimodal (characterized by the use of various music-related documents, e.g., recording information and musical representations (e.g., scores, sonogram visualizations) while proceeding listening [1]. Authors Mathieu Barthet and Simon Dixon argue that the creation of software for musicologists should facilitate switching between closed and multimodal listening modes, as well as interaction with scores and lyrics serving as a reference in performance analysis, using content based on MIR techniques [2]. Preserving works in the field of performing arts requires combining conceptual practices and methodologies between musicology and archival science to preserve works holistically, being achieved through: 1) researching theoretically the works specificities and analysing their documentation considering musicological and archival science methodologies to developing a new theoretical framework; and 2) investigating practice: documenting productions as part of the notion of the post-custodial archive. Thus, re-performing music theatre implies making an archaeology of works gathering scattered documents and understanding mutual interactions [3]. Performance studies on collaborative practices [4] or performance documentation [5, 6, 7] are useful, since systematising works of a performative nature covers several methodological steps based on pre-existing documentation or interviews with composers, performers, directors, producers, studio musical assistants or other contributors of performances.

The terminology relating to artworks that include performance is varied, so some researchers are dedicated to creating documentation tools to safeguard performance-based artworks. The assessment of the existing documentation relating to performing artworks within the Tate collection led to the development of the 2018 *Strategy for the Documentation and Conservation of Performance* [8]. This strategy involved the drafting of a glossary of terms in order to standardise the information and facilitate the identification, by the Tate team, of the different types of performance that appear in each activated and/or installed collection. According to these authors, the Tate concept of performance is described as "works of art created through actions taken by the artist or by other participants, which can be live or recorded, spontaneous or with a script" [8]. This problem related to terminology affects the performing arts in general, especially dance and music from the 1950s onwards. In that sense, it seems essential to me to deepen archival theory to face challenges in representing performance, aiming to find the appropriate sustainable archival standards and/or well-defined ontologies (exploring the conceptual model RIC), which allow for information interoperability facilitating users' research.

2. POST-CUSTODIAL FORMS OF ARCHIVE

Recent experiences concerning dance and (digital) archives may also motivate the searching for alternatives to document music theatre works or other in the context of performing arts. In "Dance and the (Digital) Archive: A Survey of the Field", the authors made a kind of survey of the various online resources available for the documentation and/or archiving of dance in different manners, proposing a division into four categories of resources which they called 'online dance archives' [9]. With that survey, they intended to feed future projects and experiences of the TKB - Transmedia Knowledge-Base for the Performing Arts. They sought to understand how online dances were formed and maintained at an archival level: whether by collecting (dance collections), by accumulation (social media), by storage (personal websites), or by assemblage (research projects), having outlined four categories to encompass such "archival" practices. As previously mentioned, the TKB platform allows the construction of personal collections by the artists themselves, as well as the curation of various materials, which together create a network of relationships between the participating artists. The artists are able to import their materials and tag them according to their own idiosyncratic taxonomies, establishing ontologies and an interconnection between the various artists, thus expanding the network of connections between them [9]. In this sense, a parallel can be made with participatory archives, insofar as the participants themselves are considered as co-creators, as the archives are created based on collaborations with and for the community [10]. Members of a given community are responsible for much of the archival work itself, from their contribution to the record through assessment, archival organization, description and access, having a greater voice in the construction of the archive and creating their own metadata, while following advice from professional archivists [10]. Archives more directly engaged with the community fit into the notion of the post-custodial archive. As stated by Fernandes at al., "post-custodial forms of the archive, which often, if not always, include and depend upon digital forms of internet archival architectures, therefore represent a major challenge to the institutions whose primary mission has to do with what we call 'to collect'" [9]. TKB reflects a post-custodial approach while working as a participatory archive, as the artists decide either to upload or remove their own materials and choreographic resources, as well as deciding when and what should be archived and published. However, this approach can be problematic, not only because archiving is not regulated by archival standards for the description of metadata allowing the interoperability of information, but also because it does not facilitate the consistency of a uniform taxonomy, because each artist has their own language and decides which words to use and tag, thus also contributing to the dispersion of information. The same occurs with the various musical languages that emerged in the different forms of art that arose mainly after the 1950s, both in dance, music, and other creations in the field of performing arts, changing the way in which this variety of languages can be represented, especially in an (digital) archive. Eric Ketelaar refers to the performative turn in archival science, highlighting Diana Taylor's notion that "[P]erformances function as vital acts of transfer, transmitting social knowledge, memory, and a sense of identity through reiterated, or... twice-behaved behaviour" [11]. Archival documents only work when they are used, and there is the possibility that they can be reinvented as new performance events [12]. For example, the performance artist Marina Abramović "has long since started on a process of recovering her performances, photographing, and recording them on video, thus believing that she keeps her work alive. The artist prepares the performers for the re-performances on the grounds that her indications help them to enter her language and philosophy more easily" [13]. Abramović in the directions she conveys to the performers, passes on her own testimony believing that she contributes to the authenticity of her performances.

Methods for ensuring the authenticity and reliability of various works of art can serve as a model for how these qualities can be preserved in digital recording systems outside the arts. Ketelaar refers to the experiments undertaken by the research team InterPARES 2, which include case studies in the creative and performing arts to understand how these disciplines conceptualise authenticity, reliability and accuracy in interactive and dynamic systems in music, dance, theatre, moving images, and interactive media installation [11].

A collection supposes a random organisation of the documentation. According to Ketelaar, there are archival artists who use the category of collection, varying their materials according to what the art critic Hal Foster calls a quasi-archival logic, and presenting their documents as a quasi-archival architecture [11]. Ketelaar is referring, among others, to the archival organisation of composer Arnold Dreyblatt. Ketelaar tackles Dreyblatt's "T Projects", in which the composer tracks the movement of records and their meaning within and outside organisations, using a duplicate archive of over 4.000 documents he created from state archives in Europe and North America. Ketelaar states that these "T documents" are also used in Dreyblatt's reading projects and simulate the living environment in which records are created, stored and used. People are invited to participate in a functional but temporal "archival installation system". Thus, for Ketelaar, in installations and immersive performances such as those created by Dreyblatt and others, people and documents become an "immersive archive", demonstrating that records are created and used by people who are component and controlled by people. record keeping systems [11]. Yet according to Ketelaar, performance scholars examine archival theory to address the challenge of representing performance. Regardless of being a form of post-custodial archive, artists involved with dance, music or theatre should benefit from institutional aid, in order to systematize their collections according to the proper archival standards/models.

3. THE REPRESENTATION / NOTATION OF PERFORMING ARTS CREATION: TWO CASE STUDIES

Digital humanities research applied to the performing arts contributes to our understanding of the complex nature of works involving performances in dance, music, or theatre, as well as their collaborative creation processes. Discussing issues related to the archiving and re-performing of electroacoustic music, computer music and digitized music, as well as sharing approaches and knowledge on preserving digital media [14]. Hence, music archiving through recourse to digital technologies is significant in the preservation of musical works comprising technological means (e.g., tape) playing key roles within the works with corresponding implications for performance archival practices. Digital Annotation allows experimentation with digital technology in documenting, analysing and disseminating dance/performance. The annotation of movements or gestures from videos in an archival context requires very detailed work that ranges from annotation of the look to the raising of the arm, among many other aspects of the performance. Aiming to document a brief performance corresponding to the movements/gestures of the dancer in the music theatre work *FE...DE...RI...CO...* (1987) by the composer Constança Capdeville, I carried out an experiment resorting to the video annotator MotionNotes [13]. I was able to annotate movements and mark place notes, images, drawings, etc., writing a kind of notation for the movement part. Initially, I started taking notes from the only video recording that exists, which belongs to the Gulbenkian Archive, but after talking with the dancer João Natividade, he stated that he was crawling with his eyes closed along the edge of a grand piano and, according to him, to annotate this performative piece, one should consider the main intention, which was the feeling that he was in the middle of the abyss while moving over the edge of the grand piano. The dancer's testimony altered all of the performance documentation initially made for this sequence demonstrating how the contributions from former performers are essential for more precise annotation, while the study of video recordings fall short especially when the footage is in poor condition.

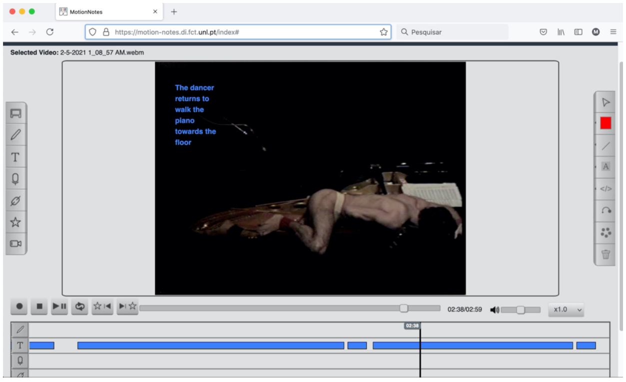


Figure 1. João Natividade's performance/gestures noted using MotionNotes.

Another tool used to aid music analysis or graphic annotations is *iAnalyse* software [15], developed by Pierre Couprie, allowing the musicologist to access a set of different files (images, sound, markers, etc.), permitting the synchronisation of the pages of a score on an audio or video file or the creation of representations from the audio signal, amongst other possibilities, helping to understand visually the collaborative creation process and serving also as a guide to listening. The result of the analysis can be exported and visualized in video format. *iAnalyse* was particularly useful for the analysis of *Double* (1982) also by composer Constança Capdeville, as it allowed the identification of elements not indicated in the score or other documents [13]. The recording of the live performance of this music theatre work from 1982, the only documentary trace that exists of the complete performance, was crucial in this process, as was the additional documentation (scripts, graphic and prescriptive score, recorded sounds on tape, images, composer's notes, and so forth). One cannot follow a score of this type in a conventional way, as we do with Mozart's scores for example, because the idea of overlapping elements permeates almost all of Capdeville's music theatre work, see Figure 2 [13]. It is a verticality created from the overlap of the various elements involved in the work that are arranged horizontally, such as a heterogeneous counterpoint, so I tried to represent such an idea in the video, therefore overlaying the scores.

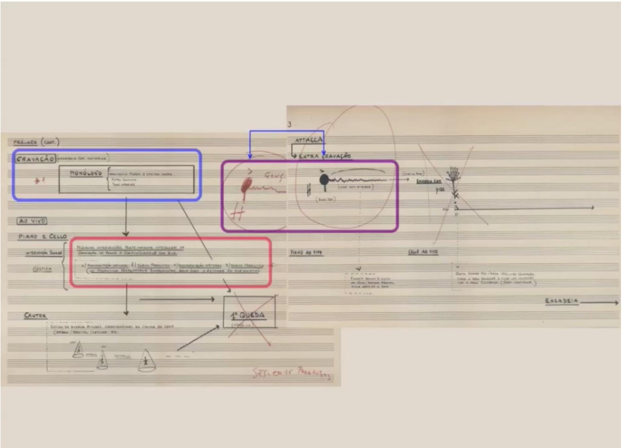


Figure 2. The Prologue section in which the monologue starts with the scores superimposed to facilitate the listening guide.

Borrowing some of the ideas from Serge Lemouton, with regard to electroacoustic, the question begs to be asked, who is responsible for ensuring the preservation of these works? Is it the composer, the performers, the musicologist or the archivist? And how do we systematically transpose this set of elements into the archive? [16]. Answering this question inevitably incorporates the development of innovative methods and tools that assist and autonomise the work of musicologists. This response still remains open and ongoing and can only ever be effective through interdisciplinary approaches from the different fields of study such as computer science, (digital) philology, performance art preservation and archival research, and with designers even making use of computer-aided design and collaborative e-platforms to document performance composition as hitherto referenced for dance or the performing arts, to assist in the creation of documentation strategies for preserving works of this type.

4. WAYS OF ARCHIVING: INSTITUTIONS VS. PERSONAL ARCHIVES

Even when institutions support artists, communication strategies may be well established, proliferating the dissemination of artists' works. However, it remains difficult to ensure sustainability without proper archival standards and/or well-defined ontologies allowing the interoperability of the information. As a representative example of the above-mentioned problems, the archive Internationales Musikinstitut Darmstadt (IMD) essentially houses materials about the history of the International Summer Courses for New Music (Darmstadt Summer Courses), whose various editions began to be held in Darmstadt from 1946 onwards. This archive is one of the main European musical collections of Post-World War II.

In the IMD archive several documents exist, from correspondence to photographs, audio and video supports, sources that document more than 70 years of history and performative practice in music. The digital archive comprises around 89,000 records, including 7,400 audio titles, 27,500 photos and contact prints, 38,000 letters, telegrams, and postcards, 13,000 administrative documents and 1,700 other text documents.[1]

The organisation of the database[2] of the IMD archive does not provide a "research path for the reconstruction of the production, documentation and conservation of pre-modern organizational information",[3] as it seeks to "confer to this documentation an organisational and organic character", that is, "the sources are treated", but not "the sources are thought of" [17]. Still regarding the IMD archive database, in Figure 3 it is shown a search by composer, in the particular case John Cage (1912-92).

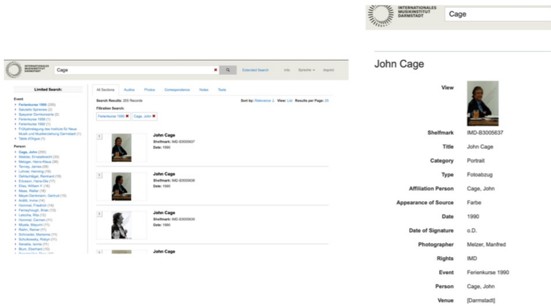


Figure 3. Search by Cage at the IMD archive database. Images extracted from it.

Although the information is indexed, or arranged in the form of an index, as an example, in the 1990 Darmstadt Summer Course, there are 255 records associated with Cage, an enormous number of images, but when clicking to access, only basic features appear such as the title, the date, the name of the photographer, and so on, without any references to a custodial history or explanations about the appearance of the documentation. Consequently, the historical context of this documentation production is not taken into account, nor are historiographical questions posed from an in-depth and scientific treatment. On the contrary, a "technical" treatment of the materials / "sources" [17] is made, which does not reflect on these sources. Hence Schmidt refers above to a notion of a vast archive, that is, almost free by the random nature of the organisation of the documentation [18].

Regarding artists who think, create, and preserve their own archives, they should be guided. Although there are exceptions in the ways of thinking and preserving personal archives, as is the case of the renowned Portuguese

[1] Source: the IMD archive available at https://internationales-musikinstitut.de/en/imd/ueber/profil/ (accessed on January 7, 2022).

[2] Source: Database of IMD Archive available at https://www.imd-archiv.de/search (accessed on January 8, 2022).

[3] "Percurso de investigação para a reconstrução da produção, documentalização e conservação da informação organizacional pré-moderna." (p. 574); "Visa conferir a esta documentação um carácter organizacional e orgânico" (p. 576); "Tratam-se as fontes", não se "pensam as fontes" (p. 552). Author's translation.

photographer Duarte Belo, the treatment of these archives should be regulated by archival standards to avoid the dispersion of information. In 2021, Manaíra Aires Athayde published an interview in which she sought to understand the methodologies used by Duarte Belo to think and rethink his vast and complex archive consisting of about 1.8 million photographs, printed and digital, in addition to notebooks, drawings and maps. Of course, this is an extraordinary case, since Duarte Belo is both the creator of a very rare archive, but also its constant interpreter, conceiving his own methodology from which he thinks and rethinks the archive. According to him, it is an exercise in constantly redesigning the archive [19]. But what about Portuguese composers? How do they deal with the archiving of their own works? In the case of music, a score is a document that includes musical notation functioning as a language or a written representation of music, which intends to be understood in a general context, but also serves to preserve music ensuring the endurance of musical works. Not every musical work includes sheet music or graphic scores, however. Mainly, from the 1950s onwards, the works start to include other resources, non-conventional media, ranging from analogue to digital formats, such as for example, magnetic tapes, electric or musical instruments, computers, etc., and this constitutes a risk to their preservation, given the obsolescence of such means. These musical works require new preservation methods, which include the production of proper complementary documentation on the musical text, encompassing information about software and hardware along with the respective composer's intentions [20]. Researcher Andreia Nogueira carried out a study that aimed to understand the preservation practices of Portuguese composers, culminating in a survey, in which , the researcher asked whether Portuguese composers took the necessary measures to safeguard their personal archives; what kind of documents did they produce and archive; if they had already been prevented from presenting any work due to technological obsolescence; and if Portuguese composers thought they were adequately documenting their creations, especially those produced in the analogue/digital era. The survey was sent to 113 composers, receiving 53 replies (45 men and 8 women). From the data obtained, Nogueira concluded that most Portuguese composers are not particularly interested in preserving their works by themselves, possibly preferring to use an archival service to perform the task. The researcher mentions that the same is the case with visual artists. The majority share this position, although they often benefit from institutional support for the preservation of their works, which cannot be said for composers. In this regard, Nogueira believes that new networks and documentation repositories should be created to help composers in this delegation of responsibilities. The researcher also argues that composers, musicians, musicologists, archivists, and conservators should work together in the preservation of the Portuguese musical legacy, especially the more experimental productions [20] involving technological resources such as those above-mentioned, but of course expanding the preservation practices applied to contemporary productions of this type in general.

5. FUTURE REFLECTIONS ON THE PRESERVATION OF PERFORMING ARTS CREATION IN AN (DIGITAL) ARCHIVE

Fundamentally, the main objective is long-term preservation and to process information it is necessary to obey the FAIR principles so that the (meta)data is easily located, accessible, interoperable and reusable in the future. Data interoperability allows information to travel from one system to another without losing its original characteristics. The central characteristic of these languages is to define which attributes of the information will be the object of that information and how they are described, in terms of vocabulary, semantics and syntax. In an organisational context formed by regulations, entities and functions, it is necessary to document/record the actions; documents, when they are integrated into a record system according to certain requirements, become records. In order to ensure the authenticity of archival documents, it is necessary to control the transmission, evaluation, custody and preservation of documents and implement and document policies and procedures, using technologies and standards. Moreover, to represent information, standards are needed and sometimes there is a need to group these standards together (e.g., ISAD (G)[4] conjugated with RISM[5] being applied to musical documents). When, in an institution, a document management system, which serves to manage the documentation between the various organisational subgroups, proves to be insufficient, the description model must be adapted to the needs. For example, the RIC[6] (Records in Context) is a conceptual model that describes relationships between entities and also the type of relationships they have with each other. RIC creates the intellectual framework to link archive resources to other cultural information, allows us to represent ontologies and also describes classes, instances, relationships, attributes, and constraints. An ontology working as a coding language adequate to each field of performing arts so that the information is indexed by subject, based on the various elements of the work, allowing a holistic view of it in archival contexts. A conceptual model based on an ontology associates' relationships (record/archive document and context). In

[4] See the link available at: https://www.ica.org/en/isadg-general-international-standard-archival-description-second-edition (accessed on January 27, 2022).

[5] Répertoire Internacional des Sources Musicales — available at http://www.rism.org.uk (accessed on January 26, 2022).

[6] See the link available at: https://www.ica.org/sites/default/files/session-7.8-ica-egad-ric-congress2016.pdf (accessed on January 27, 2022).

order to create a meta-information language model suitable for musical documents and which also serves musical creation in a more contemporary context, it is essential to create an application profile suitable for all kinds of users (musicologists, musicians, researchers, archivists, librarians and others) and understand their research needs more comprehensively. This model requires a broader and more complex interdisciplinary work, which understands and combines the methods of musicology, archival, digital libraries and computer science so that it is possible to understand and respond to the various research needs of these users.

6. CONCLUSIONS

In this paper, I attempted to discuss the different ways of thinking and organising the (digital) archive, giving some concrete examples. The archiving of artistic works or creations in the context of the performing arts from a holistic point of view requires close engagement between the practices of the various disciplines in the field of performing arts and the methods, standards or models instituted within archives, digital libraries or documentation centres to allow the interoperability of the information in a standardised way making sense for artists, archivists and users.

Acknowledgments

This paper was made possible with the support of CESEM, Centro de Estudos de Sociologia e Estética Musical da NOVA FCSH, UID/EAT/00693/2019, with the financial support of Portuguese Science and Technology Foundation (FCT) through national funds.

7. REFERENCES

[1] A. Bonardi, "IR for Contemporary Music: What the Musicologist Needs," *Proceedings of the ISMIR 2000 1st International Symposium on Music Information Retrieval*, Plymouth, Massachusetts, USA, October 23-25, 2000.

[2] M. Barthet and S. Dixon, "Ethnographic Observations of Musicologists at the British Library: Implication for Music Information Retrieval", *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011.

[3] F. Magalhães and I. Pires, "Recovering Music-Theatre Works Involving Electronic Elements: The case of Molly Bloom and FE...DE...RI...CO....," *Organised Sound*, vol. 24, no. 1, 2019, pp. 96-108. doi:10.1017/S1355771819000098.

[4] M. Blain and H.J. Minors (eds.), *Artistic Research in Performance through Collaboration*. USA/UK: Palgrave Macmillan, 2020.

[5] P. Auslander, *Liveness: Performance in a Mediatized Culture*, second edition. Abingdon, New York: Routledge, 2008.

[6] A. Dekker, G. Wijers and V. E. J. P. Van Saaze, "The Art of Documentation," *Notation / RTRSRCH*, Vol 2, no.2, 2010, pp. 22-27.

[7] H. Marçal, A. Nogueira and R. Macedo, "Materializar o intangível: a documentação da obra Luís Vaz 73 (1975), de Jorge Peixinho e Ernesto de Sousa," *Conservar Património*, Vol. 27, 2018, pp. 13-22. https://doi.org/10.14568/cp2016042.

[8] L. Lawson, A. Finbow and H. Marçal, "Developing a strategy for the conservation of performance-based artworks at Tate," *Journal of the Institute of Conservation*. DOI: 10.1080/19455224.2019.1604396

[9] C. Fernandes, S. P. Coelho and A.B. Vieira, A. B., "Dance and the (digital) archive: A survey of the field," *Dance Research*, vol. 38, no. 2, 2020, pp. 271-288. https://doi.org/10.3366/DRS.2020.0313.

[10] A. Roeschley and J. Kim, "'Something that feels like a community': the role of personal stories in building community-based participatory archives," *Arch Sci*, vol. 19, 2019, pp. 27–49.

[11] E. Ketelaar, "Archival turns and returns: Studies of the archive," in A. J. Gilliand, S. McKemmish, A. J. Lau (Eds.), *Research in the Archival Multiverse*, Clayton, Victoria: Monash University Publishing, 2017, pp. 228-268.

[12] S. Jones, D. Abbott and S. Ross, "Redefining the performing arts archive," *Archival Science*, vol. 9, 2009, pp. 165–71.

[13] F. Magalhães, "Music, performance, and preservation: insights into documentation strategies for music theatre works," *International Journal of Performance Arts and Digital Media*, vol. 17, no. 2, pp. 1-25. Doi: 10.1080/14794713.2021.1974726.

[14] M. Akkermann (ed.), *Array 2020 – Archiving*. USA: ICMA, 2020.

[15] D. Fober, J. Bresson, P. Couprie and Y. Geslin, "Les nouveaux espaces de la notation musicale: Groupe de travail AFIM," *Journées d'Informatique Musicale*, Faculté de musique de l'Université de Montréal, Montréal, Canada, 2015. hal-01160759v2.

[16] S. Lemouton, "The electroacoustic repertoire: Is there a librarian?" *Array – Archiving*, 2020, pp. 7-14. https://doi.org/10.25370/array.v20202625.

[17] M.L. Rosa, "Reconstruindo a produção, documentalização e conservação da informação organizacional pré-moderna. Perspetivas teóricas e proposta de percurso de investigação," *Boletim do Arquivo da Universidade de Coimbra*, vol. 30, 2017, pp. 547-586.

[18] D. Schmidt, "'The Darmstadt Events'. Archival Strategies, Music-Historical Work and Cultural-

Political Research Perspectives on the Development of the Digital Archive," *Archival Notes*, vol. 3, 2018, pp. 147-157.

[19] M.A Athayde, "Duarte Belo E a génese Do Arquivo," MATLIT: *Materialidades da Literatura*, vol. 9, no.1, 2021, pp. 211-35.

[20] I. Pires, F. Magalhães and A. Nogueira, "Preservation and technological obsolescence: Portuguese contemporary musical heritage in perspective," *Journal of New Music Research*, vol. 47, no. 4, 2018, pp. 355-364. doi: 10.1080/09298215.2018.1486433.

SANGEETTEX: A LATEX ENGINE FOR TRANSCRIBING AND RENDERING INDIC MUSIC

Chandan Misra
XIM University
Bhubaneswar, India
chandan@xim.edu.in

ABSTRACT

Notation system is the building block of a particular genre of music which aids in reading, composing, and performing music in a structured manner. Though computers are being used for annotating musical scores in *Staff notation*, it has not been used for Indic music to a substantial extent. Available systems and archives either employ romanization or uses image formats for representing scores which makes it less usable in terms of recreating scores as part of other documents, retrieving musical information out of the format etc. Creating music-sheets for Indic Music in computer is a inconvenient process involving placing music symbols in its proper place according to the underlying grammar and drawing the same similar to published musical texts. Document preparation tools like Latex is suitable choice for this task making it easier for developers to print quality music-sheets. In this paper, we present SangeetTEX, a Latex based music rendering engine for *Rabindra Sangeet* (*Tagore Songs* in English), a distinguished genre of Indic music and a collection of more than 2200 songs composed and written by Bengali poet and Nobel Laureate *Rabindranath Tagore*. It allows users to create beautiful music sheets while preserving the published typesetting in its original published form and provides easy exchange of musical information in text format. SangeetTEX is available at https://github.com/cmisra/SangeetTeX.

1. INTRODUCTION

Notation system for music provides means to read, compose, and perform music in a systematic manner. Staff notation is the standard music notation for traditional western classical music. Although, *Indian Classical Music* (ICM) has been orally transmitted from teachers to students, practitioners soon realized a music notation system for aiding the learning as well as performing the traditional artform. *Pt. Vishnu Narayan Bhatkhande* and *Pt. Vishnu Digambar Paluskar* was the first to create a notation systems for *North Indian Classical Music* or *Hindustani Sangeet*, termed as *Bhatkhande* [1] and *Paluskar* [2] notation system respectively. *Rabindranath Tagore*, poet and Nobel Laureate, gave birth to a unique kind of music genre called *Rabindra Sangeet*. Asia's first Nobel Laureate also proposed a new notation system, *Akarmatrik Notation System* [3] which was used to notate all of his (more than 2200) songs. There are other prominent notation systems present, *Dandamatrik* and *Sargam* [4] for *South Indian Classical Music* or *Carnatic Sangeet*.

Since, it is very important to make Indic music in computer in terms of music education and practice, the very first step towards practicing the art in a more Indic environment (contrary to Staff notation where there are always a chance to miss certain musical information during score writing). One of the starting points for creating such systems is to provide a tool to properly render the music-sheet on a piece of paper. *Swarshala* [1] [5] from *Swar Systems*, is a prominent Indic music software that allows practice, compose and learn both Hindustani and Carnatic Sangeet. However, it uses romanization for entering musical notes to the system. Therefore, the system requires a mathematical realization of the sheet architecture in a computer and respective musical fonts for transcription. One of the attempts was to render Tagore compositions [6] in a computer that leads to a mathematical framework, Swaralipi [7], that tries to make a unification of various Indic music-sheet representation while extracting common features of different notation systems and language script to make a general notation system. However, such framework when actually implemented does not generate expected results due to the rendering challenges posed by certain music symbols - *Meend* symbol of Akarmatrik notation system and vertical line to represent $t\bar{a}lbibhag$ symbol in Bhatkhande notation system. Additionally, it is not easy to place music symbols above and below other symbols which is why new notations are developed to curb the need for digitization of music-sheets. While these systems are quite capable of entering, storing and playing back music, publishable quality rendition of music-sheets is too hard to achieve and therefore it actually diminishes the very objective of creating the music-sheet in the first place.

In this paper, we try to develop a rendering engine, SangeetTEX, based on Latex document preparation tool that specifically allows music software developers to create print quality music-sheet implementing the grammar of ICM. Initially, we develop Latex commands to render scores of *Tagore Songs* following the model *Swaralipi* since it has

[1] SwarShala 4 by Swar Systems. Visit the following link for more details https://www.swarclassical.com/SwarShala/

a large corpora of songs and accompanied scores all published in 63 volumes [3], an ideal test bed for SangeetTEX. Later we try to include other varieties of classical music in SangeetTEX by implementing respective notation systems. The commands let the developer concentrate on methods for entering music symbols on the user interface and let them free from the complexities of music-sheet rendition. SangeetTEX defines algorithms that place various symbols on the music-sheet with the fonts for scores and lyrics. The choice of Latex is suitable since it can draw shapes whenever it is necessary to render certain symbols which would not be rendered properly if fonts were used. Since, the structure of the $t\bar{a}l$s are same in all form of traditional music, the same engine can be applied to other forms of notation systems.

2. A BRIEF ON SWARALIPI MODEL

The present Latex notation rendering engine is built upon a well-established musical framework, *Swaralipi* [7], which had been created to encode, arrange, display, and render Indic music symbols on a computer. The framework can arguably support all major Indic notation systems active at present in India. Although, the core of the framework consists of a row and a column model for rendering rows and columns of the music sheet respectively, it is greatly influenced by the annotating styles of Indic music and Indic language scripts. However, to understand the present work, explanation of the row and column models are sufficient and interested readers can therefore may refer to the original article cited for a comprehensive detail.

The framework can be visualized as a collection of 2D matrices of heterogeneous dimensions where each 2D matrix corresponds to a single line of the composition. Each such matrix consists of cells at each intersection of all the rows and columns of the matrix. These cells serve the containers of the music symbols. The row and column model define the number and alignment of the rows and columns for each 2D-matrix in the model. While row model takes only the lines of the composition as the input, the column model calculates the number of columns using three parameters: $t\bar{a}l$ and the implicit pattern carried with it, the *avartan* or the number of cycle, and the position of each music symbol in each 2D-matrix or line. We have maintained the same input parameters while designing SangeetTEX as described in Section 3. Below we describe the row and column model.

2.1 Row Model

The row model determines the number of rows that should be present in each 2D-matrix of the music sheet. In its simplest form, row model prescribes two rows for a line - the score and the lyrics. However, variations on this simplest form is possible with the introduction of other musical components like melody change during repetition, *Meend* (musical ornament), end of composition, $t\bar{a}l\bar{a}nk$ or beat markings etc. These variations led to the formation of two separate segments: *Primary Melody Segment* which consists of the musical components related to the primary

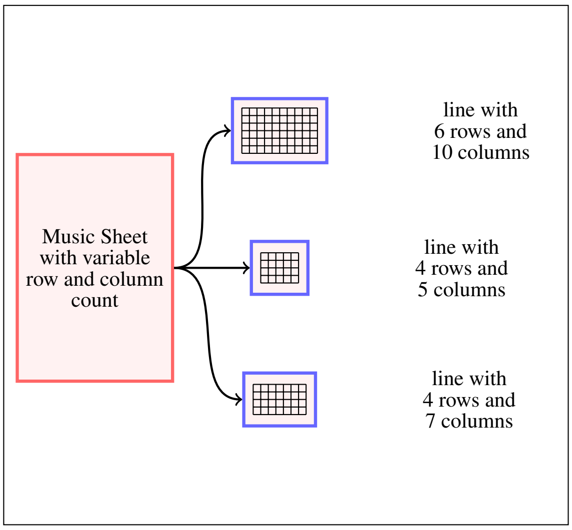


Figure 1. 2D-Matrix model of SangeetTEX.

melody line, and *Changed Melody Segment*, which consists of alternate musical components followed while repeating part of the primary melody line. Each segment mentioned above, consists of maximum four rows: $t\bar{a}l\bar{a}nk$ or beat marking row, score row, *Meend* row, and lyric row. However, not all the 2D-matrices in a composition require eight rows since, not all the lines use all the musical components.

Row model also implements the rendition of *Meend*, the ornament which slides from one note to another note of different pitch over a specified number of beats (similar to *Glissendo*). The form of the symbol makes it difficult to place in a particular cell of the 2D matrix. The model solves the problem by introducing three new symbols - *Meend Start*, *Meend Continue*, and *Meend End* and placing them in the *Meend Row* of the corresponding line. *Meend*s can differ in length depending on the number of beats it covers and can be implemented by repetitive use of the *Meend Continue* symbol.

Although the model provides a probable solution to the Meend issue, in a practical scenario, it fails to achieve proper rendition similar to published music piece. This includes alignment problem that gives users the freedom to pose variable spacing between adjacent cells of a 2D-matrix. Since, the continue symbol length is fixed, it is not possible to make proper justification of the *Meend*. This led us to draw the symbol in latex to render it in more neater and flexible way as described in Section 4.1.

2.2 Column Model

The column model takes $t\bar{a}l$ and *avartan* as the input and computes the number of columns present in each 2D-matrix and the position of the music symbols in it. Since every $t\bar{a}l$ is accompanied by its beat pattern, the column count can be easily computed as explained in the following example:

Figure 2 shows a 2D-matrix (bottom part of the figure) created from a single line (top part of the figure) having *Shashthi* $t\bar{a}l$ with two unequal measures having 2 beats in the first measure and 4 beats in the last, 2 *avartan*s and the lyric line is written in Bengali script, have been transformed into the architecture having 2 rows and 17 columns.

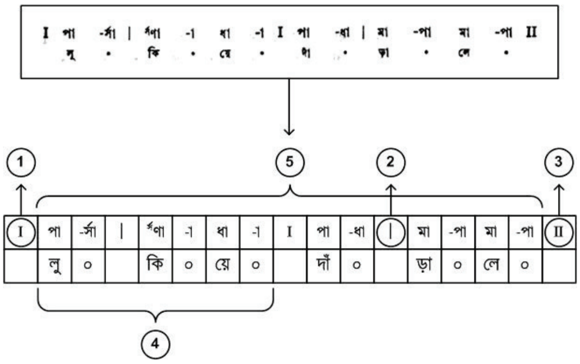


Figure 2. Determination of column number of a line of Rabindrasangeet score in the present architecture.

The calculation depends on three symbols as shown in Figure 2:

1. Symbol 1 or (I): The beginning of the $t\bar{a}l$ occupies a cell before the first beat of the $t\bar{a}l$. It is repeated once every cycle of the $t\bar{a}l$ or *avartan*.

2. Symbol 2 or (|): The $t\bar{a}lbibhag$ symbol which comes between adjacent measures of the $t\bar{a}l$ and occupies a cell by itself. Since there are only two measures per cycle of this $t\bar{a}l$, symbol 2 is used only once per cycle and can be generalized using the following equation:

$$n = (a \times m) \quad (1)$$

where n is the number of columns needed for symbol 2, a is the number of *avartan*, and m is the number of measures of the $t\bar{a}l$.

3. Symbol 3 or (II): The end of a musical phrase after which the first phrase (called the *Aasthayee*) must be sung; it also occupies a cell by itself.

Putting it altogether gives us the total column count t as:

$$t = (b \times a) + (a - 1) + n + 2 \quad (2)$$

where b is the total number of beats in the $t\bar{a}l$. The column count 17 in the above example can be easily verified from the equation.

3. STRUCTURAL ANATOMY OF SANGEETTEX

Since we are interested in creating a new document format i.e. music-sheet for score transcription and not in adding more functionalities to an existing type of document (Latex's default document class *article* for example), a *class* rather than a *package* [2] [8], have been considered a better choice for our project.

[2] Latex2E for Class and Package Writer. Visit https://www.latex-project.org/help/documentation/clsguide.pdf

The foremost aspect of consideration to creating the structure of the music-sheet format is the symbols, which in this case belong to two classes, to be transcribed and rendered through Latex compiler. Two classes of symbols, the scores and the lyrics, are rendered using two different fonts.For lyrics, a Unicode Bangla font, named *Bangla* [3] [9] has been used and the scores use a custom font, named *Swarabitan* which was designed particularly to be used for writing scores for *Rabindra Sangeet* music-sheet. The use of Unicode font for the transcription in the TeX class file made us to select the compiler as *XeTeX* as a Unicode input and font aware engine. Another variant, LuaLaTeX engine, can also be used for this purpose.

The structure of the music-sheet is implemented in a single class file, named *sangeet* class file, and is primarily made up of two commands, namely *scoreLine* and *scoreLyricLine*, which correspond to creating a score line without and with a lyric line respectively. The scoreLine command takes three parameters: the name of the $t\bar{a}l$, the number of *avartan*, and the note sequence to form a single line of score. The *scoreLyricLine* command has an extra parameter, the lyric syllables, to be added beneath every note in the score line along with the above mentioned parameters. It is important to mention that the third parameter, note sequence, in both the commands do not include $t\bar{a}lbibhag$, *start* and *end of song* symbols.

The primary objective of the *scoreLine* command is to calculate the number of cells required to accommodate the music symbols and their respective positions on that particular line. This would obviously exclude the *Meend* symbol, since it would require different rendering altogether, to be discussed in Section 4.1. The total number of cells in a line is to be computed using equation 2 which defines the number of columns (column model) for a line in a music composition, while the position of each symbol requires more involved calculations which is discussed in Section 3.2.

3.1 Computation of Number of Columns

As already mentioned, the computation of number of columns requires two input parameters: $t\bar{a}l$ and avartan. These parameters have been given in both the classes *scoreLine* and *scoreLyricLine*. However, $t\bar{a}l$ name does not contribute to the actual calculation and therefore, require other $t\bar{a}l$ related parameters like *maatra* or measures, total beat count of a particular $t\bar{a}l$. These values are implicit to a specific $t\bar{a}l$ but must be given explicitly to obtain the count. Maatra and total beat count can be obtained from the beat pattern of the $t\bar{a}l$ and a comprehensive list of beat pattern of all $t\bar{a}l$s used in *Rabindra Sangeet* is added in the swarabitan class file. This list is included in a new latex environment named *sangeet* and the value of *maatra* and beat count are also initialized there to be used as a global variable later in both the classes.

Finally, the column count is calculated using equation 2 as follows:

[3] Bangla, Unicode Bangla Font. Download at https://www.omicronlab.com/bangla-fonts.html

$$
\begin{aligned}
colcount &= ((avartan \times (maatra - 1)) \\
&\quad + (avartan \times beatcount) + (avartan - 1) \\
&\quad + 2 \\
&= (a \times (m-1)) + (a \times b) + (a-1) + 2
\end{aligned} \tag{3}
$$

3.2 Evaluation of the Positions of the Music Symbols

Since, both the latex commands takes only notes as the third input parameter (which is also intuitive since musical notes are the only symbols which are variable for different compositions), it is a bit tricky to determine the positions of the notes and other music symbols in any 2D-matrix. Typically, two essential symbols are present in a musical line: The beginning or end of the *tāl* (I) and *tālbibhaga* (|) as discussed in section 2.2. As previously seen, the presence of these two symbols prohibits notes to occupy consecutive positions in the 2D-matrix. Fortunately, the underlying systematic pattern of *tāl* resolves this indexing problem and therefore, implemented here with careful computation. Below we briefly explains the steps of the calculation (with a running example as given in Figure 2):

1. The key input and the tunable parameters for the computation are: *beat count* (b), *maatra* (m), *avartan* (a), beat pattern of *tāl* ($bPat$).

2. We use 1-based indexing meaning that the symbols occupy cells that start from column one till column equal to the total number of columns (given in equation 2) in that 2D-matrix. For example, in the running example the column index starts from 1 and ends at 17.

3. We create a sequence of numbers that starts from 1, increments by 1, and stops at the column count. For the running example it gives the following sequence: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$

4. We identify the indices where we place (I) symbol and put the indices in a sequence. It is placed at the beginning, at the end, and between cycles of the beats (*avartan*). If the line consists of single *avartan*, then there will be only two places where it is to be placed. The index for additional occurrences of the symbol is given by Algorithm 1. For the running example, the sequence contains three entries $\{1, 9, 17\}$.

5. We identify the indices where we place the *tāl* Bibhag Symbol (|) and insert them into a sequence. The indices to place the symbol is tricky since various *tāl* consists of unequal beat pattern as given in Table. The following algorithm (Algorithm 2) uses the number of beats in the first Bibhag or measure to calculate the indices:

   In the running example, the *tāl* bibhag sequence contains two entries $\{5, 13\}$.

6. Since, we are done with the positions of the two primary symbols, we can extract the indices of the positions of the notes by first combining the indices of step 4 and 5, and subtract it from the column count sequence created in step 3. In the running example, this would lead to the sequence:
   $\{2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16\}$.

Algorithm 1: Algorithm to find the positions of the start symbol (I)

```
Data: m, a, b, t
Result: Sequence seq that contains the indices of the
        symbol
insert 1 to seq
if a ≠ 1 then
    index ← 1
    for i ← 1 to (a − 1) do
        index ← index + b + (m − 1) + 1
        insert index to seq
    end
end
insert t to seq
return seq
```

Algorithm 2: Algorithm to find the positions of the *tāl* Bibhag symbol (|)

```
Data: m, a, b, t, bPat
Result: Sequence seq that contains the indices of the
        symbol
/* Calculate the number of times tāl bibhag
   symbol appears                                  */
talBibhag ← a × (m − 1)
for i ← 1 to talBibhag do
    insert i to talSeq
end
/* get the beat count in the first Bibhag */
first ← bPat[1] N ← length(talSeq)
for i ← 1 to N do
    temp ← 2 × (talSeq[i] − 1) + 1
    index ← first+(b×(talSeq[i]−1)+temp)+1
    insert index into seq
end
return seq
```

3.3 Evaluation of the indices for Meend

Before drawing the meend between notes, it is required to know the indices of the start and end notes. To annotate the start and finish of the meend in any line of composition, we place a backtick character (`) character just before the corresponding characters. Once we get the start and end characters we can draw the meend from start to end according to the method described in section 4.

4. RENDERING OF THE MUSIC SYMBOLS

We use tikz library to render the music sheet and once the positions of the music symbols have been evaluated, it is easy to place them using tikz. Each line of the music piece or 2D-matrix is rendered as a separate figure and each cell of such matrix is realized by a tikz node. A tikz node is

Notation Properties	English Alphabet	Rendition
Pure Notes or *Śuddh Svar*	s r g m p q n	স র গ ম প ধ ন
Flat Notes or *Komal Svar*	v t d u	ঋ জ্ঞ দ ণ
Sharp Notes or *Tīvr Svar*	k	ক্ষা
Middle Octave or *Suddh Saptak*	s r g m p q n	স র গ ম প ধ ন
Upper Octave or *Tār Saptak*	sf, rf, gf, mf, pf, qf, nf	র্স র্র র্গ র্ম র্প র্ধ র্ন
Lower Octave or *Mandra Saptak*	sh, rh, gh, mh, ph, qh, nh	স্ র্ গ্ ম্ প্ ধ্ ন্
Whole Note or *Purna Maatra*	sa ra ga ma pa qa na	সা রা গা মা পা ধা না
Half Note or *Ardh Maatra*	si ri gi mi pi qi ni	সঃরঃগঃমঃপঃধঃনঃ
Quarter Note or *Siki Maatra*	sI rI gI mI pI qI nI	স॰ র॰ গ॰ ম॰ প॰ ধ॰ ন॰
Meend	w y and x	

Table 1. Notation symbols in Akarmatrik notation system and corresponding English alphabets and symbols in SangeetTEX.

a named placeholder which can contain text inside it. The name of a node uniquely identifies it in a figure. To render a single line we create t square tikz nodes and set the name of the node with indices 1 to t. Since, we already have the indices of various symbols, we insert the symbols as text into respective nodes. For completeness, we remove the border from each node so that it looks line a printed composition.

4.1 Rendering of Meend Symbol

For the rendition of the *meend* symbol we use the *draw* tool of tikz and create a bezier curve. The bezier curve is defined by four points: two endpoints (start and end) and two control points that determine how curve it is. We place the first two endpoints at the bottom midpoint of the start and end note of the *meend*. However, in our case, the curve operation is accomplished using angles instead of control points. We provide angles at which the curve should leave the start point and reach the target point. Moreover, we create a *cycle* by drawing another curve just above already drawn curve that reaches the start point from the end point. At last we fill the area created with color so that it looks exactly same as meend.

We have considered another important feature of *meend* symbol is that it is stretchable depending on how far start and end notes are situated. There would be a rendering issue if we try to give same angles to a wide meend as a narrow meend. This would make the wide meend fat in the middle and would provide wrong rendition. We try to adjust the angles proportional to the length of the meend so that it preserves the same height of the symbol.

4.2 Notation used in SangeetTEX

SangeetTEX uses *Swarabitan* font that chooses several English alphabets or the combinations of them to represent each symbol of the Akarmatrik Notation System. The English alphabets are chosen intelligently so that most of the notes resemble phonetically to the music symbol. The notation system uses same letter for both *Sparsh Svar* and *Svar*, but with a smaller size in case of *Sparsh Svar*. In these cases, the captial and smaller version of the same alphabet is used. A comprehensive list for the notes and corresponding English alphabets are given in Table 1. Since, we render the *Meend* in a separate way, we don't use the *Meend* keys (w, y, and x) to represent *Meend* in the commands.

5. DEMONSTRATION OF THE ENGINE

In this section, we try to show snippets of the Latex code and the corresponding rendition of the music sheet. We will also provide the meta data of the song, so that the reader can validate the transcription as required.

In the first example (Table 2), we give several rendition of the compositions with some of the prominent $t\bar{a}l$s of Indian Classical Music. The simplicity of the engine lies in the fact that it only requires the name of the $t\bar{a}l$ and number of *avartan* of the composition and everything else has been taken care of by the rendering engine. The user can then place the musical symbols according to the notation presented in Table 1. The user only has to take care while entering the music symbols and check the number of symbols is equal to the total number of beats in the $t\bar{a}l$.

The second example (Table 3) demonstrates the rendition of the variations of the *Meend* symbol in different context and therefore proved to be a better rendition engine than published music sheet. We have also included `\scoreLyricLine` command in the table to render score with lyrics in a composition.

<table>
<tr><th>$t\bar{a}l(pattern)$ and [avartan]</th><th>Latex Statement and Rendition</th><th>[Line Index][Parjaay][Song Index]</th></tr>
<tr><td>Dadra (3-3) [2]</td><td>\scoreNewPhrase{dadra}{2}{Mga,ma,-ua,Nda,pa,-da,Dma,-a,-pa,mpa,-dua,dpa}
II মগা মা -ণা | নদা পা -দা I দমা -া -পা | মপা -দণা দপা I</td><td>[1][Puja][215]</td></tr>
<tr><td>Iktāl (3-3-3-3) [1]</td><td>\scoreline{iktaal}{1}{ma,-pa,pa,pa,pa,pqa,ma,-pa,pna,na,sfna,sfa}
I মা -পা পা | পা পা পধা | মা -পা পনা | না র্সনা র্সা I</td><td>[3][Anusthanic][1]</td></tr>
<tr><td>Teentāl (4-4-4-4) [1]</td><td>\scoreNewPhrase{teentaal}{1}{sa,ga,rga,rgmpa,mpma,ga,ga,ga,
Gma,ma,ma,mpma,Gma,ga,ga,gra}
II সা গা রগা রগমপা | মপমা গা গা গা | গমা মা মা মপমা | গমা গা গা গরা I</td><td>[1][Prem O Prakriti][24]</td></tr>
<tr><td>Jhaanp (2-3) [3]</td><td>\scoreNewPhrase{jhap}{3}{sa,-a,sa,-a,sa,sa,-a,
-ra,-a,-na,sa,-ra,rpa,-a,-a}
II সা -া | সা -া সা I সা -া | -রা -া -না I সা -রা | রপা -া -া I</td><td>[1][Prakriti][25]</td></tr>
<tr><td>Dhamaar (3-2-2-3-4) [1]</td><td>\scoreline{dhamaar}{1}{pha,pha,-a,pa,-a,pa,-a,pa,pa,-a,na,-Qna,sfa,-a}
I প্া প্া -া | পা -া | পা -া | পা পা -া | না -ধনা র্সা -া I</td><td>[3][Puja O Prarthana][54]</td></tr>
<tr><td>Kahaarba (4-4) [2]</td><td>\scoreline{kaharba}{1}{-sa,-pa,-pa,-ma,}
II সর্জ্জা -া -া -া | র্জ্জা -া র্সা -া I র্সর্জ্জা জ্জা -সা র্সা | ণর্সা -া ণা -দা I</td><td>[1][Swadesh][12]</td></tr>
<tr><td>Shashthi (2-4) [2]</td><td>\scoreline{shashthi}{2}{SFna,-rfa,rfa,-sfa,sfa,-na,na,-a,Npa,-ka,qpa,-a}
I র্সনা -র্রা | র্রা -র্সা র্সা -না I না -া | নপা -ক্ষা ধপা -া I</td><td>[4][Prakriti][23]</td></tr>
<tr><td>Teora (3-2-2) [2]</td><td>\scoreNewPhrase{teora}{2}{sfa,-a,sfa,sfa,-a,sfa,sfa,
SFna,-a,qa,Qna,-a,qa,pa}
II র্সা -া র্সা | র্সা -া | র্সা র্সা I র্সনা -া ধা | ধনা -া | ধা পা I</td><td>[1][Swadesh][17]</td></tr>
<tr><td>Jhampak (3-2) [3]</td><td>\scoreNewPhrase{jhampak}{3}{\{sfa,Msfa,-da,na,sfa,vfa,
vfa,vfa,sfa,-a,na,sfa,-a,-a,-a}
II {র্সা মর্সা -দা | না র্সা I র্ঋা র্ঋা র্ঋা | র্সা -া I না র্সা -া | -া -া I</td><td>[3][Puja][166]</td></tr>
</table>

Table 2. Rendition of SangeetTEX on different $t\bar{a}l$s having different Maatras along with the reference of the source for validation

<table>
<tr><th>Property</th><th>Latex Statement and Rendition</th><th>[Line Index][Parjaay][Song Index]</th></tr>
<tr><td>Score and Lyric</td><td>\scoreLyricNewPhrase{dadra}{2}{sa,sa,-ra,ra,ra,-a,ra,-a,-ga,Gra,-ma,ga}{অ,নে,ক,দি,নে,র,আ,o,o,মা,র,যে}
II সা সা -রা | রা রা -া I রা -া -গা | গরা -মা গা I
অ নে ক দি নে র আ o o মা র যে</td><td>[1][Prem][21]</td></tr>
<tr><td>Meend</td><td>\scoreline{dadra}{2}{na,-a,-a,sfa,-a,-vfa,‘na,‘sfa,‘-a,‘-da,-a,-a}
I না -া -া | র্সা -া -র্ঋা I ‘না ‘র্সা ‘-া | ‘-দা -া -া I</td><td>[10][Prem][43]</td></tr>
</table>

Table 3. Rendition of SangeetTEX on different $t\bar{a}l$s having different Maatras along with the reference of the source for validation

6. CONCLUSIONS

In this paper, we develop a music-sheet rendering engine, SangeetTEX based on Latex document processing application for the rendition of *Rabindra Sangeet* scored on *Akarmatrik Notation System*. SangeetTEX is implemented on the theoretical 2D-matrix model *Swaralipi* with the instructions to structure rows and columns of the every line of the composition. We give details of the class file along with the commands that are responsible for the rendition of score, lyric, and variations of score in the compositions. We give algorithms to calculate the number of cells for each 2D-matrix and the positions for each of the music symbols for easy rendering. We give special emphasis on the rendition of *Meend* symbol since it is one of the primary reasons to adopt such a document preparation tool to prepare print quality music-sheet. Finally, we validate the engine by annotate compositions with different $t\bar{a}l$s and score variations.

Acknowledgments

The *Swarabitan* font as been designed and developed by the author while working in a research project funded by Society for Natural Language Technology Research, Department of Information Technology and Electronics, Govt. of West Bengal, India. The font is freely downloadable from the SNLTR website [4] [10].

7. REFERENCES

[1] V. N. Bhatkhande, *Hindustani sangeet paddhati: bhatkhande kramik pustak*. Sangeet Karyalaya, 1990.

[2] D. Courtney, "North indian musical notation: An overview," https://chandrakantha.com/music-and-dance/i-class-music/n-indian-notation/, accessed: 2022-01-30.

[3] R. Tagore, *Swarabitan*. Visva-Bharati, 1949.

[4] S. Mammen, I. Krishnamurthi, A. J. Varma, and G. Sujatha, "isargam: music notation representation for indian carnatic music," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2016, no. 1, pp. 1–12, 2016.

[5] S. Systems, "Swarshala 4," https://www.swarclassical.com/SwarShala/, accessed: 2022-01-30.

[6] C. Misra, B. Bhattacharya, and A. Basu, "A new framework to preserve tagore songs," *World Digital Libraries-An international journal*, vol. 3, no. 1, pp. 63–72, 2010.

[7] C. Misra, T. Chakraborty, A. Basu, and B. Bhattacharya, "Swaralipi: A framework for transcribing and rendering indic music sheet," 2016.

[8] T. L. Project, *LaTeX2e for class and package writers*, 2006.

[9] "Bengali fonts - bangla fonts - free download," https://www.omicronlab.com/bangla-fonts.html, accessed: 2022-01-30.

[10] "Society for natural language and technology research," https://nltr.itewb.gov.in/, accessed: 2022-01-30.

[4] Swarabitan font can be downloaded by visiting the site `https://nltr.itewb.gov.in`

SYMBOLIST RE-IMAGINED: BIDIRECTIONAL GRAPHIC-SEMANTIC MAPPING FOR MEDIA NOTATION AUTHORING AND PERFORMANCE

Rama Gottfried
Hochschule für Musik und Theater
Hamburg, Germany
rama.gottfried@hfmt-hamburg.de

ABSTRACT

SYMBOLIST is an in-development application for experimental notation, which aims to provide an un-opinionated authoring environment for the design and performance of symbolic notation. By following an information visualization rather than prescribed musical orientation, the application is thought of as an open play-space, with tools for experimentation and thinking visually about relationships between representation and interpretation in media performance. In the paper we begin with an overview of the project's background, iterations and relationship to the DRAWSOCKET project, and introduce a redesign of the system, centered on a new framework for custom symbol definitions for bidirectional mapping and user interaction. In conclusion we discuss future development directions and evaluation of the project.

1. BACKGROUND

The origins of the SYMBOLIST project can be traced back to 2011, through the development of a composition practice in Adobe Illustrator using a plugin called Scriptographer, [1] which allowed users to create new drawing tools in Javascript which could then be used in Illustrator as an interactive brush. As a composer working with experimental instrumental techniques and spatial notation, Scriptographer was perfect for my composition needs at the time, since you could design a notation for a given technique or musical expression and then code it as an interactive graphic function, which could then be manipulated graphically in Illustrator. With the access to the mouse movement, interactions could be used to compose different elements of the notation, much in the same way that mouse interaction is used in programs like Processing. [2]

For example, a note with a spatially indicated duration typically is written as a note-head of some shape with a line extending out from it to show its duration. Using Scriptographer, you could create an interaction for composing note-duration symbols, where clicking down on the Illustrator canvas places the note-head, and then using the mouse drag, determine the end of the duration line, ending at the location of the mouse up event.

Further, you could also group elements to structure hierarchies of objects which would then appear in Illustrator as grouped objects, visible as nested folders in the layers menu.

Around the same time, I was studying at UC Berkeley's Center for New Music and Audio Technologies (CNMAT) developing approaches to instrument design using OpenSoundControl (OSC) [1] for data structuring. One day, while working with Scriptographer, I had saved a score in Illustrator as Scalable Vector Graphics (SVG) format [3] and accidentally opened the SVG file in a text editor. In the SVG file I noticed that all of the graphic objects were there in a human readable format, and closely resembled the kind of nested objects that we were working on at CNMAT in the Odot library[2]. This gave me the idea that I might be able to translate the SVG information into OSC, then "perform" the OSC score in much the same way as you would a stream of OSC coming from a sensor-based instrument—in a similar spirit to Daphne Oram's "Oramics" [3] and Xenakis' UPIC system [4], but with a greater focus on symbolic interpretation.

Soon after, while composing for a high-resolution spatial audio rendering system, I found that I was lacking a way to compose spatial movements graphically, in a way that would connect with instrumental notation practice. After some experiments using Blender [4] to draw 3D curves which could then be parsed via Python and sent out over OSC, I was dissatisfied by the perceptual differences between common practice notation and the kinds of 3D representation I was able to create in Blender—both had their merits, but what was missing was a compositional frame that connected the two representation paradigms into a unified notation system. Due to time constraints, for this piece I fell back on using automation controls in Ableton Live [5] and sending OSC to control the movements using Max for Live. [6] This was practical, but this kind of automation approach has the limitation of forcing the composer to separate each data parameter into separate streams of data (e.g. x, y, z), whereas in a symbolic representation multiple attributes can be indicated in unified graphic representation. Following these experiences [5] I later returned

[1] https://scriptographer.org/
[2] https://processing.org/

[3] https://www.w3.org/TR/SVG11
[4] https://www.blender.org/
[5] https://www.ableton.com/en/
[6] https://cycling74.com/

to the SVG-OSC transcoding idea, and developed a first working model which was presented at the 2015 TENOR conference [6].

In the meantime, the Scriptographer project was abandoned by its developers after Adobe drastically changed their plugin API in version CS6. The new Adobe API was different enough that it would require a significant amount of work, and even then the mouse interaction tools that were used in Scriptographer were no longer accessible, so the authors decided to stop development on the project and later went on to create Paper.js, [7] which has some similarities with Scriptographer, but is more closely related to Processing since it no longer is bound to the Illustrator application environment.

1.1 Symbolist JUCE

After preliminary tests using the SVG-OSC transcoding for score playback, it was becoming increasingly complicated to parse complex hierarchies of symbols in order to format them into OSC streams, so in 2017 I began work in collaboration with OpenMusic [7] developer Jean Bresson, through an Ircam-ZKM Musical Research Residency towards the goal of creating a system that could replace the Scriptographer/Illustrator approach that I had developed so far.

The first version of SYMBOLIST was created in 2018 as a standalone JUCE [8] application, which provided the basic tools for drawing vector graphics and query system that allowed SYMBOLIST to be used as a lookup table for OSC stream playback [8].

Working in JUCE seemed practical since it has a wide user base and is used for audio plugins as well as Max and Ableton Live applications. There were some minor complications, due to JUCE's incomplete SVG support, [9] however, generally, we were able to create a working first prototype of the system.

1.2 Clefs and Bidirectional Mapping

A conceptual turning point in the project occurred towards the end of the residency as we were thinking about how to simplify the process of using the SVG data for controlling digital processes.

In the first version of SYMBOLIST, as in the original SVG-OSC implementation, the graphic data needed to be interpreted by the application that received the data. Using Odot I would parse the graphic OSC information coming in from SYMBOLIST, and then map the data to other processes, for instance synthesis parameters or coordinates for spatial rendering.

The process of interpretation requires that the parsing-mapping algorithm knows the context of the graphic objects. For example, that a circle is a note-head and not a rhythmic dot, and so on. Like the axes of a graphic plot of information, contextual musical symbols (meter, staff-lines, clefs) indicate to the reader how they should interpret the notes and rhythmic symbols written on the staff.

The following phase of SYMBOLIST development continued as I began work at the Hamburg University of Music and Theater, working with Georg Hajdu in the Innovative Hochschule project. Continuing from the idea of the *clef* as a graphic symbol that contextualizes the notation on a musical staff, I began work on developing a system for SYMBOLIST that would allow users to define interpretive symbols for hierarchical context parsing. Since the data is already in hierarchical format inside the application's data structure, it was logical that SYMBOLIST could interpret the graphics internally, and then stream OSC post parsing and mapping, rather than streaming the raw graphic information which required complex parsing of the graphic hierarchies to contextualize the symbol references, as in traditional common practice notation.

Following this line of thought it became clear that what SYMBOLIST really needed was a system for *bidirectional mapping*, where graphic data is interpreted as symbolic data with semantic meaning, and inversely, that the user should also be able to send data in its semantic representation to SYMBOLIST, where it would then be mapped to its graphic representation.

In 2018 I began implementing this idea into the JUCE version of SYMBOLIST, but soon needed to switch tracks to focus on a different notation issue for a project at the Innovative Hochschule, developing a platform for realtime networked score display.

1.3 Drawsocket and Symbolist JS

At the end of 2018, and first half of 2019 we developed a drawsocket, a server/client framework for realtime dynamic notation using web browsers for graphic rendering [9, 10]. Based in node.js and standard web technologies of HTML, SVG, CSS, and Javascript, DRAWSOCKET is essentially an OSC wrapper [11] for web browsers, [10] providing a homogeneous message API for the creation and realtime manipulation of browser elements.

Returning to SYMBOLIST after the first DRAWSOCKET concerts, I began wondering if the framework of SYMBOLIST needed to be redesigned from the perspective of bidirectional mapping, since I realized that this was most likely going to be the most important part of the graphic-data relationship that the system is developing towards.

SVG is very well supported in modern browsers, and having just worked with node.js and browser technologies for DRAWSOCKET, it seemed that the flexibility of Javascript could be a convenient option for users to create custom symbol definitions (as in Scriptographer). I decided to see how fast it would be to implement a proof of concept using Electron.js, [11] a cross-platform desktop application development using a node.js server and Chrome as a front-end, used for applications like Skype, WhatsApp, Visual Studio Code, Slack, WordPress, and others.

[7] http://paperjs.org/
[8] https://juce.com/
[9] https://forum.juce.com/t/complex-svg-files-fail-to-load-properly/26917/16

[10] And by extension, provides access to other media via WebAudio, WebGL, etc.
[11] https://www.electronjs.org/

Using DRAWSOCKET as a frontend, and node.js for the backend I found that I was able to get up and running very quickly with Electron, and decided to continue SYMBOLIST development in this direction, bringing the experience gained from the JUCE version to the creation of a deeper structure for the creation of symbol interpretations that integrate into the graphic manipulation of symbolic data.

2. APPLICATION STRUCTURE

The new implementation of SYMBOLIST is organized as a server-client model (Figure 2), comprising of:

- The main SYMBOLIST application server, running in node.js, which serves the main display page and manages messages between the client and server via WebSocket [12] connection. Within the main server, a child process called the "io-controller" handles input and output from external sources via OSC over a UDP socket, and maintains the "*score*," a database of hierarchical score elements, stored in SYMBOLIST "*semantic representation*" format (Section 4).
- The "*editor*," a browser-based user interface client, which displays the graphic representation of the data, and allows the user to edit and create new data objects through graphic interaction. The "ui-controller" runs in the browser and handles interaction via a library of definition scripts, which specify mappings to and from data and graphics formats, as well as other tools and interactions.

Since the system is now based on a web-server model, we are able to also use Max's *node.script* object to run the server from within Max as an alternative to the Electron desktop app.

3. GRAPHICAL AUTHORING AND INTERACTION

The SYMBOLIST graphic user interface (Figure 1) is designed around units of symbolic objects and their contextual containers. Graphic "*symbol*" objects are placed in "*container*" symbols, which function as a context frame that can be used to interpret the meaning of the symbol.

In order to maintain an open and un-opinionated approach, the SYMBOLIST framework does not specify how containers and symbols should look, act, or respond when you interact with them. Rather, the interaction and meanings of the symbols are defined in a library of custom object "*definitions*," which specify meaning through mapping semantic data to-and-from its graphic representation.

Symbol definitions provide a mechanism to design custom-tailored composition environments for particular authoring situations, stored as Javascript libraries, which can be shared between users, and used as templates. Leveraging web-browser technologies like JS, HTML, CSS, SVG, etc., there are many ways to customize the layout in SYMBOLIST, and potentially, the definition libraries could completely transform the editor layout to serve a particular use-case scenario.

3.1 Interface Components

The main graphic components of the default SYMBOLIST graphic editor are:

- *Score view*: the top-level view of the application window, containing the main view and side bar. Sliders are provided to offset the view of the document, as well as basic zoom functionality.
- *Palette*: a set of buttons in a side toolbar displaying icons of symbols that have been defined for the current selected container context.
- *Tools*: a set of buttons that open high-level tools, that can be used for operations like algorithmic generation of new symbols, or applying transformations to existing elements (e.g. alignment of multiple objects, setting distributing objects, or other operations).
- *Inspector*: a contextual menu for editing the semantic data of a selected symbol, which on update is mapped to the graphic representation and sent to the server to update the main score database.

3.2 User Experience

On entering the application, the editor loads a score or configuration file from the default load folder, which sets the top-level page setup and palette options. A typical sequence of creating a score might be as follows:

1. The user opens a workspace with one or more default container symbols displayed on the screen, for example an empty rectangle, which is like a piece of paper.
2. Clicking on the "paper" container rectangle *selects* it, and then the user sets it as the new *context* by pressing the [s] key.
3. After setting the context, the palette toolbar is populated with icons of symbols that are defined with the selected container context type.
4. Clicking on one of the palette toolbar symbol icons puts the interface into "*palette mode*," where the mouse interaction is now designed for use with this specific symbol type.
5. Holding the Command(Mac)/Control(Win) key enters "*creation mode*," which by convention draws a temporary preview of the symbol (how it will appear when you click), and displays the corresponding semantic representation data as textual feedback.
6. After clicking, the symbol is placed in the container.
7. Clicking and dragging a symbol graphically modifies its semantic data in reference to the container context. In the case of common practice notation this would be how you would change the time and pitch information of a symbol. The interaction results depend on the "*selection mode*" specified in the symbol definition.

[12] https://websockets.spec.whatwg.org

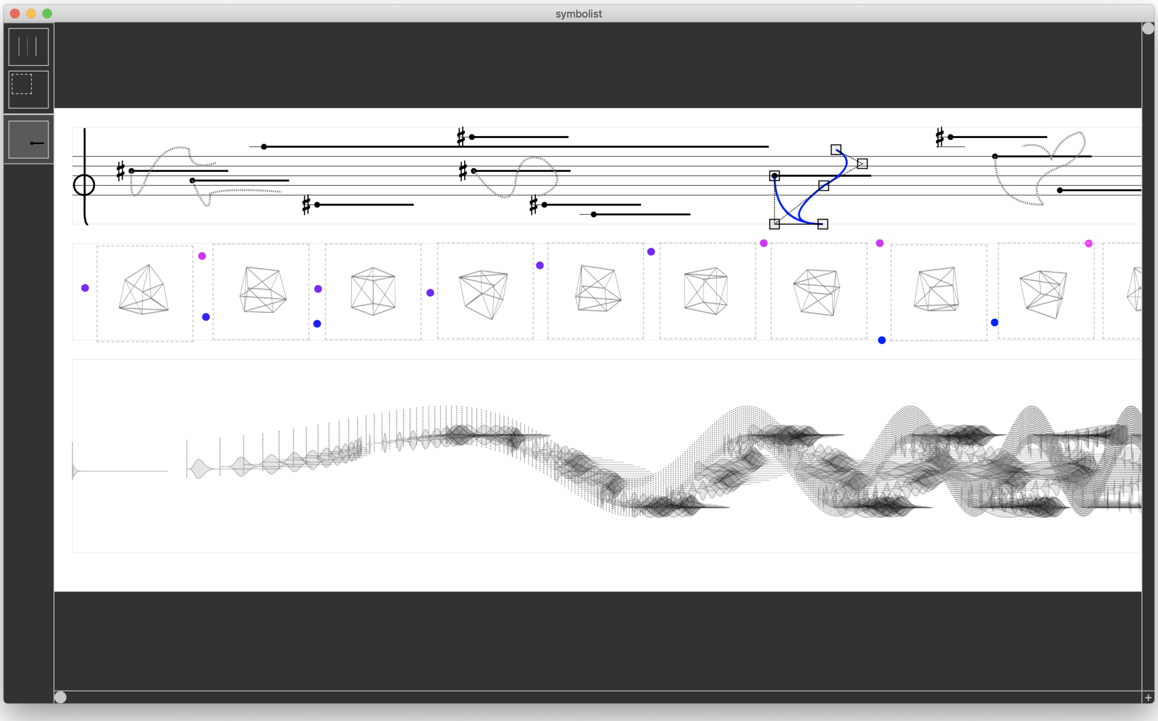


Figure 1. SYMBOLIST screenshot, showing some different types of staves, and editing capabilities.

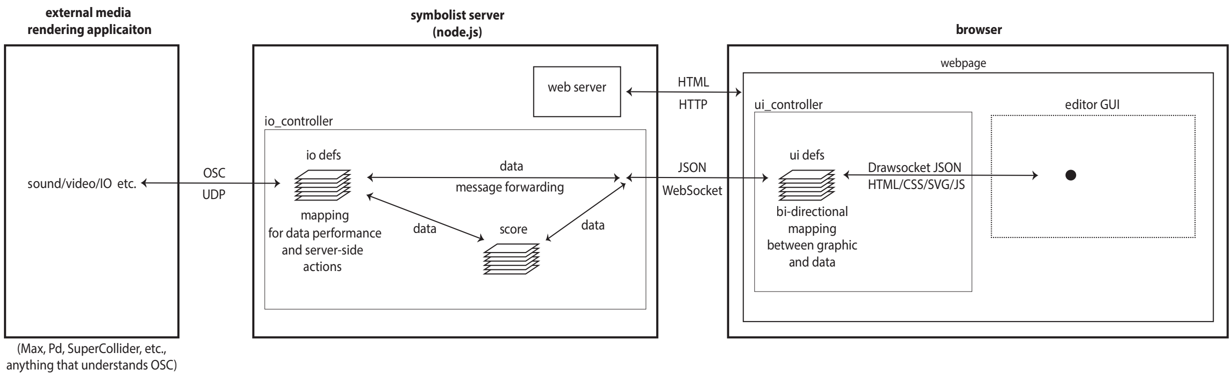


Figure 2. SYMBOLIST architecture.

8. User can also modify the semantic parameters as text by selecting the symbol and hitting the [i] key, which brings up the inspector window, where you can edit the data directly.

9. Pressing the [e] key enters "*edit mode*" for the selected symbol, useful for editing of internal attributes that are less relative to the container symbol. For example, you would enter edit mode to adjust bezier curve anchor points.

4. DATA REPRESENTATION

At the heart of SYMBOLIST are two parallel forms of information expression: *semantic* and *graphic* representation (Figure 3).

Semantic data specifies the various attributes of information about a symbolic object in terms of the object's meaning to the author. For example, the meaningful attributes of a *note* object might be information about pitch and duration, or a *point* object might contain x, y, and z values corresponding to the point's location in 3D space. In SYMBOLIST the semantic representation is thought of as the main holder of information, which can be grouped into hierarchies to represent scores or other types of data structures.

The *graphic* representation is a symbolic visual expression of the semantic data, designed relative to the context defined by the author.

The aim of SYMBOLIST is to provide an agnostic environment for developing, and composing with, new symbolic representations of semantic data for use in multimedia composition practice; and so, the central design consideration of this new implementation is to build a flexible framework for specifying a wide range of mapping relationships between semantic and graphic representations.

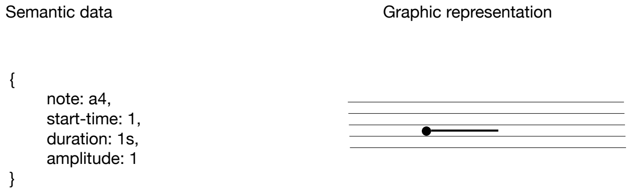


Figure 3. *Semantic* vs *graphic* representation of the same information. Note: Figures 3-6 use pseudocode for brevity (see Sections 6 and 7 for syntax details).

5. MAPPING

Between each of these representation contexts there is a layer of mapping, with the *semantic data* serving as the primary representation type.

Semantic data to graphic representation mapping (Figure 4) is used for the creation of graphic symbols from a stream of input, for example from generative processes, textural authoring, or computer assisted composition systems [7, 12, 13, 14, 15].

Graphic representation to semantic data mapping (Figure 5) is used in order to create or edit data based on graphic information. This is the typical "graphical user interface" situation, where the data is accessible through its visual representation.

Semantic data to performance media mapping (Figure 6) is the use of the data as a sequence of events that can be played in time (or used to control other processes not necessarily in time).

Note that in SYMBOLIST mapping between *performance media* and *graphic representation* is achieved through first mapping to semantic data. See section 8 for further discussion.

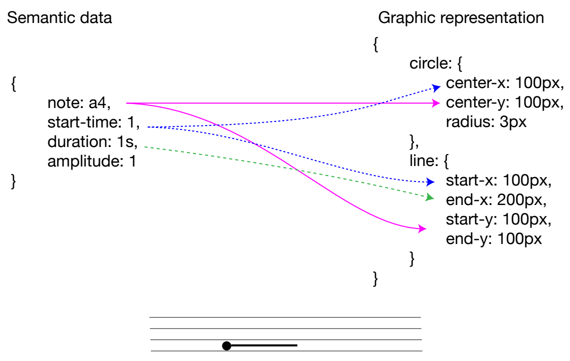


Figure 4. *Semantic* data mapped to create a *graphic* representation from input data.

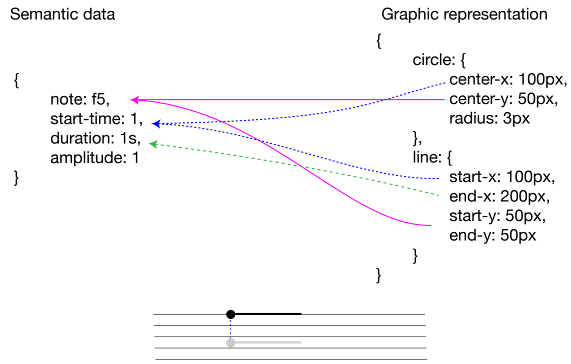


Figure 5. If edited graphically, the updated graphic data is then mapped back to *semantic data* representation.

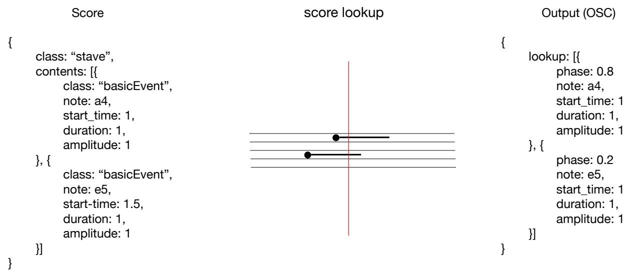


Figure 6. Using the lookup method defined by the symbol class, the *semantic data* can be used to perform external instruments via Open Sound Control.

6. SEMANTIC REPRESENTATION

Within the SYMBOLIST application semantic data is stored as Javascript objects and read/written in JSON [13] format (transcoded to-and-from OSC for inter-application communication).

The main attributes used in SYMBOLIST semantic data objects are:

- *id*: a unique identifier name (required).
- *class*: a reference to the definition of the object type in the user-definition library (required).
- *contents*: an array of child objects that a parent container object might hold (required for container symbols).

Semantic data objects may include any number of other attributes [14] (*pitch*, *amplitude*, etc.). For example a simple semantic object might look like:

```
{
  "id" : "foo",
  "class" : "legs",
  "action" : "jump",
  "start_time" : 0.1
}
```

Here we see an object with the *id* "foo," which is of *class* "legs," that has an attribute *action* associated with it and a start time.

[13] https://www.json.org/json-en.html
[14] The term *attribute* is used here interchangeably with properties, parameters, aspects, etc.

6.1 Containers

Symbols may also contain other symbols. Container symbols function to frame their contents, providing reference and context like a plot graph frame, which provides a perspective and scaling for interpreting the set of data points displayed in the graph. Similarly, when a semantic data object contains other objects, the children are stored as an array in the object's *contents* field. For example, for an imaginary class "timeline," which holds two types of leg actions, we might write:

```
{
  "id" : "bar",
  "class" : "timeline",
  "duration" : 1,
  "contents" : [{
    "id" : "foo-1",
    "class" : "legs",
    "action" : "jump",
    "start_time" : 0.1
  },{
    "id" : "foo-2",
    "class" : "legs",
    "action" : "sit",
    "start_time" : 0.2
  }]
}
```

6.2 Score Files

Since the data elements are stored as JS objects, it is easy to import/export SYMBOLIST scores as JSON files.

When the application loads, it reads a default initialization file in the form of a SYMBOLIST score. The current default initialization config file looks like this:

```
{
  "about" : "some metatdata",
  "id" : "Score",
  "class" : "RootSymbol",
  "contents": {
    "id" : "trio",
    "class" : "SystemContainer",
    "x": 200,
    "y": 100,
    "duration": 20,
    "time": 0,
    "contents" : [{
      "id" : "oboe",
      "class" : "FiveLineStave",
      "height" : 100,
      "duration": 20,
      "time": 0,
      "contents" : []
    },
    {
      "id" : "bassoon",
      "class" : "PartStave",
      "height" : 100,
      "time": 0,
      "duration": 20,
      "contents" : []
    },
    {
      "id" : "synth",
      "class" : "PartStave",
      "height" : 200,
      "time": 0,
      "duration": 20,
      "contents" : []
    }]
  }
}
```

In this example, we can see there is a "RootSymbol," which contains a "SystemContainer," which in turn contains two "PartStave" symbols and one "FiveLineStave" symbols.

7. BROWSER NOTATION

SYMBOLIST uses SVG to draw graphic symbol representation, utilizing DRAWSOCKET as a convenience wrapper to provide shorthand methods for the creation and manipulation of browser window elements.

7.1 SVG / HTML Format

The SYMBOLIST format for a *symbol* in its browser rendered notation, is a set of three group elements (<g> in SVG, or <div> in HTML) marked by *class* tags, which follow the defined symbol class name.

For a symbol class type "SymbolClassName," the SVG template would be:

```
<g id="foo" class="SymbolClassName symbol">
  <g class="SymbolClassName display"></g>
  <g class="SymbolClassName contents"></g>
</g>
```

The "*symbol*" class marks the top-level symbol group containing the "*display*" and "*contents*" groups. The "*display*" group holds all of the symbol's display information and the "*contents*" group contains any potential child elements. For simplicity all graphic *symbol* elements include both the *display* and *contents* elements as placeholders.

7.2 Storing Semantic Data in Dataset Attributes

Since SYMBOLIST is constantly mapping back and forth between semantic data and its graphic representation, we are making use of the HTML *dataset* feature [15] to store the semantic metadata inside the top-level *symbol* element.

For example, using our imaginary "legs" actions above, we include the *action* and *start_time* parameters, written as dataset attributes by using the prefix "*data-*": [16]

```
<g id="bar" class="Timeline symbol"
        data-duration="1">
 <g class="Timeline display"></g>
 <g class="Timeline contents">
   <g id="foo-1" class="Legs symbol"
          data-action="jump"
          data-start_time="0.1">
     <g class="Legs display"></g>
     <g class="Legs contents"></g>
   </g>
   <g id="foo-2" class="Legs symbol"
          data-action="sit"
          data-start_time="0.2">
     <g class="Legs display"></g>
     <g class="Legs contents"></g>
   </g>
 </g>
</g>
```

8. SYMBOL DEFINITIONS

Symbols are defined as Javascript classes, which are stored and recalled when symbol actions are performed. For each user interaction, the ui- and io-controllers look up the symbols involved in the interaction by class name, and use their definition to execute the symbol's reaction.

Definitions specify the bidirectional mapping between *semantic* and *graphic* representations and responses to OSC "*lookup*" queries which can be used to perform the score.

[15] https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/dataset

[16] Note that according to the HTML dataset specifications, all names will be converted to lowercase, this can create issues in some cases, so best practice is to use all lowercase for attribute names.

There are two types of definition scripts:

- *ui-definitions* run in the ui-controller and perform user interactions based on the different interaction modes, and applies bidirectional mapping between semantic and graphic representations.
- *io-definitions* run in the io-controller and are used to assist in the lookup and OSC *performance* mappings of the semantic data to media like sound synthesis, video, etc., or to perform server-side score manipulations.

In each controller context there are certain methods and variables that need to be defined in order for the class to function properly in the SYMBOLIST ecosystem. Users may also call custom io- and ui- methods from external applications via OSC (described further in Sections 11 and 12).

For convenience, there is a base class template that can be used to handle most common interaction situations, which may be overridden by sub-classes for custom handlers. A set of helper functions are provided in global objects called "*ui_api*" and "*io_api*" which can be used in definitions for many essential operations. Eventually it is planned to create a graphical tool in the editor to help define symbol definitions, but this is not yet implemented.

9. UI DEFINITIONS

At the time of writing, the variables and methods defined in the symbol class used in the the ui-controller when handing user actions are:

- *class*: the unique name of the symbol, used to store and lookup the symbol definition.
- *palette*: an array of class names of other symbols that can be used within a container symbol, which are drawn in the palette toolbar when the user selects the symbol as a new context.
- *getPaletteIcon*: called when drawing the palette icon, returns DRAWSOCKET drawing commands.
- *paletteSelected*: called when the user clicks on the symbol icon, used to trigger custom UI. When the symbol is selected in the palette, the definition should enable its mouse handers.
- *getInfoDisplay*: called when creating the inspector window; returns drawing commands for the inspector contextual menu.
- *fromData*: called to map data from semantic to graphic representation.
- *selected*: called on selection and deselection, for optional custom UI handling.
- *drag*: called when the user drags selected symbols; by default the *ui_api translate* function is used to set the symbol's SVG translation matrix to preview the new location.
- *applyTransformToData*: called on mouse-up if selected objects have changed, and applies the transform matrix to the SVG attribute values.
- *currentContext*: called when the user enters or exits a container symbol (hitting the [s] key, [esc] to exit).
- *editMode*: called when entering and exiting edit mode.

9.1 Data and View Parameters

Looking at Figures 4 and 5 we can see that in some cases the relationship between a semantic property and its graphic representation is not a one-to-one mapping. For instance in Figure 4 the *note* property needs to be mapped to a pixel position that is used both for the center point of a graphic circle (note-head) as well as the starting point for a line (duration indication). In reverse, Figure 5 shows how when the user moves a symbol *graphically*, the new pixel positions need to be translated back to its semantic representation to update the score.

To manage the bidirectional mapping between semantic and display representation, the template base class uses an intermediate stage called "*view-parameters*". The idea is that the *view-parameters* contain the bare-minimum number of variables needed to draw the symbol.

For example, in Figure 4 the graphic representation requires a y position relative to the pitch's location in the staff, an x position relative to the start time, and a *width* value relative to the duration of the event. After first mapping from the semantic attributes *note*, *start-time* and *duration* to view-parameters x, y, and *width*, the drawing method can then use the view-parameters x, y, and *width* values to draw its two graphic objects from a single set of values.

The ui template class uses two functions to define data-view mappings: *dataToViewParams* which receives the semantic data and returns the view-parameter object, and *viewParamsToData* which performs the opposite mapping. Just as the view-parameters provide a minimal set of variables needed to draw multiple graphic graphic objects from the semantic representation, the *viewParamsToData* function uses the same view-parameters to map back to semantic data. For example, in Figure 5 the mapping only needs either the center point of the note-head or the start-x position of the line to determine the *start-time* parameter.

The template class also two additional data/view parameter translation methods to coordinate child objects with parent containers: *childDataToViewParams* and *childViewParamsToData*. For example, in Figures 4 a note-head circle is drawn from its *note* parameter, whose position is relative to the parent five-line staff object. Inside the *note*'s *dataToViewParams* and *viewParamsToData* methods, it will need to "ask" its parent objects where to place itself by calling the parent's *childDataToViewParams* and *childViewParamsToData* functions. In deeply hierarchical container structures, it is possible that the parent may need to ask *its* parent for some data as well, and so the parent querying system can provide a way to maintain separation of concerns between different aspects of the notation.

10. IO DEFINITIONS

Running in the server, the io-controller's job is to handle OSC communication with external applications, reading and writing files, and maintaining the score database. Current default io-definition variables and methods used by the io-controller are:

- *class*: the unique name of the symbol, used to store and lookup the symbol definition.
- *comparator*: a comparator function used in container symbols to sort child symbols. For example, if a given container uses a *time* value for sorting, when a new child node is added, the comparator function helps the container insert the child element at the correct location in the *contents* array. [17]
- *lookup*: called via OSC to look up events at a given value specified by the container (e.g. typically time); returns the query results to the caller via OSC. By default the output is an array of all active data objects at the lookup point, along with the relative phase position within each element, useful for controlling amplitude envelopes etc. Figure 6).
- *getFormattedLookup*: called via OSC to request a complete list of events for external sequencing, formatted in the symbol definition to apply to the external syntax requirements.

Note that the *lookup* and *getFormattedLookup* methods receive the complete OSC bundle that is sent in, and also have access to the entire score database, and so it is also possible to define multiple ways of looking up (and performing) the score data at the same time; for example multidimensional nearest neighbor lookup, or polytemporal sequencing.

11. CREATING SYMBOLS FROM OSC INPUT

As an illustration of how data is processed through the SYMBOLIST architecture, we can follow the sequence of events in the case of *semantic-to-graphic* mapping; for example when algorithmically generating score data, using an outside process to create new symbols via OSC messages.

By convention, SYMBOLIST uses the DRAWSOCKET message syntax for OSC and JSON interprocess communication, where a "*key*" address is used a keyword to signal which routine should interpret the message, and the "*val*" address contains an object payload (or array of objects) to be processes.

11.1 Symbol Creation from an External Process

The io-controller has a small collection of built-in processes that can be called via OSC, the most important of which is the function to add new data elements to the score and graphic display, accessible using the *data* keyword.

For example, here is an OSC bundle using the *data* key:

```
{
  /key : "data",
  /val : {
    /class : "FiveLineStaveEvent",
    /id : "foo"
    /container : "oboe",
    /time : 0.13622,
    /ratio : "7/4",
    /duration : 0.1,
    /amp : 1
  }
}
```

The "data" keyword message has the following required and optional attributes:

- *class*: the class name of the object to create (required) .
- *container*: the *id* of the container symbol class to put the object in (required).
- *id*: a unique id to use for the data object (optional); if not specified a unique string will be generated .
- Other required or optional parameters will depend on the symbol definition.

Upon receiving an OSC message with the *key* "data," the object payload stored by *val* is added to the model, and then relayed to the ui-controller.

Data-to-View Mapping in the ui-controller

Received by the ui-controller, the semantic data then is mapped to graphic data, by looking up the symbol's *class* definition and calling the ui-definition's *fromData* method, which maps from the data representation to the graphic drawing commands.

As discussed above (in Section 9.1), when using the symbol template base-class, the *fromData* method will usually call the symbol's internal *dataToViewParams* which performs the mapping from semantic to a minimal set of graphic values which are then used to draw the graphics, by sending drawing commands to DRAWSOCKET accessed through the *ui_api*, including the HTML dataset storage, as described above (in Section 7).

A typical drawing command would look something like:

```
ui_api.drawsocketInput({
  key: "svg",
  val: {
    class: "NoteLine symbol",
    id: uniqueID,
    parent: containerID,
    ...newView,
    ...ui_api.dataToHTML(dataObj)
  }
})
```

Here, we use the JS spread operator "..." to merge the *newView* variable, holding DRAWSOCKET format SVG data organized in three <g> group containers (as described in Section 7), and the HTML dataset information, encoded via the *dataToHTML* helper function into the *val* object with the associated "svg" DRAWSOCKET keyword. The object is then sent to DRAWSOCKET via the *drawsocket-Input* helper function to be added to the browser screen.

12. CUSTOM USER METHODS

Users may also create their own custom methods in either ui or io-definition classes and call them from an outside

[17] Pre-sorting increases the efficiency for later lookup queries.

process via OSC (or from other symbol definitions), using the "call" keyword. [18]

Using DRAWSOCKET syntax, the "call" system requires two parameters in the *val* object to lookup and execute the method:

- *class*: name of the class to lookup.
- *method* name of class method to call.

However, *all* parameters in the *val* object will be passed to the function, which can be used as a variable length argument when calling the method.

Custom class methods can be used to apply operations to the score or ui, for example a method for transposing all pitches on the "Staff" named "oboe" might look like this:

```
{
  /key : "call",
  /val : {
    /class : "Staff",
    /method : "transpose",
    /id : "oboe"
    /steps : 12
  }
}
```

On receiving this OSC bundle, the io-controller will lookup the class "Staff" and attempt to call its method "transpose," passing the entire *val* object to the symbol method as an argument. The user-defined "transpose" function might then do something like lookup the "oboe" staff in the model, and then iterate all of its contents, offsetting the "note" values by the number of steps specified in the method arguments.

13. CONCLUSIONS

With the new symbol class definition system in place, initial tests seem promising, and support the hope that this new experimental SYMBOLIST implementation will be able to handle a wide variety of score and symbol structures by providing mechanisms for users to compose bidirectional mappings between semantic and graphic representation. The system has the potential to address many applications in digital media compositional practice, and may someday evolve into a fully-functional authoring environment for computer performable symbolic notation.

In order to further evaluate the robustness of the system, the next steps will be to go through the process of developing complete definition libraries for working with different types of notation systems. As a test case, we have been working on a set of definitions for common practice notation, which is planned for presentation at the 2023 TENOR conference, along with other experimental approaches.

One challenge that may need to be addressed is the ease or difficulty of creating new symbol definitions. At the moment the system is based in Javascript, which means that the user must program the definitions with textural code. However, since SYMBOLIST is a graphically oriented authoring environment, it would be convenient if there was a way to create new symbol definitions graphically within the main editor application. To address this issue, further research is planned to develop a GUI for symbol definitions, and other tools to help streamline the process of specifying bidirectional mappings. For example, most mathematical operations have an inverse operation, so perhaps there could be a GUI interface that provides tools to define both mapping directions simultaneously.

The Electron framework is currently working well for cross-platform app development, however some issues came up after the Electron version 12 update, which introduced new security measures including context isolation, [19] and increased limitation in using the *require* function used in SYMBOLIST to import user libraries. In order to comply with new Electron web safety measures, SYMBOLIST now uses Webpack [20] to bundle the symbol definitions into a static library file, which is loaded on startup—previously users were able to dynamically load symbol definitions at run time, which seems like a more natural user experience. The new security measures are less than ideal for dynamic updating, however, on the positive side, since SYMBOLIST is now browser-based and uses the same system as DRAWSOCKET for dynamic graphic rendering, SYMBOLIST could now be easily used in networked situations. For example for synchronized score display, or use with multitouch tablets (iPad etc.). In cases where SYMBOLIST is exposed to the internet, the new web-security measures may be important. More testing is needed to determine which features are the most critical, balancing usability with websecurity.

For playback/sequencing of SYMBOLIST scores, users can currently send either *lookup* or *lookupFormatted* messages to the io-controller, which will then respond with data that can be used to perform the score in another software like Max, Pd, SuperCollider, etc. The lookup system is currently implemented in Javascript, which is not the fastest or most temporally precise method of playing back the score. As a starting point we are testing a new Max external called *o.lookup∼*, which accepts a list of x and y coordinate points and reads through the sequence of points via a sample-rate phase input. This system works quite well for single data sequences (i.e. value of y at point x), however for more robust playback, it might be worthwhile to develop a more complete database lookup system in C/C++, which could provide optimized getter methods for data playback. For example, this might take the form of a Max external that can read a SYMBOLIST score and provide optimized access for playback, and instrument track selection. It could also be imagined that a score could be exported to playback in a DAW like Ableton Live, and to form connections to other OSC sequencing applications like IRCAM's Antescofo expression language[16].

Other development directions that may be interesting to pursue would be to integrate other frameworks into the application. Some first steps for 3D graphics have begun with the introduction of the three.js [21] library, visible in the rotated cubes in Figure 1, however more work is needed to provide tools for manipulating 3D graphics. In the area of notation for spatial movement, there are plans to continue

[18] SYMBOLIST will pass the same call request to both definitions, so if both have a function of the same name they will both be called.

[19] https://www.electronjs.org/docs/latest/tutorial/context-isolation

[20] https://webpack.js.org/

[21] https://threejs.org/

development of trajectories (visible in the curves attached to note events in Figure 1), and to connect SYMBOLIST with the ICST's Spatialization Symbolic Music Notation (SSMN)[17].

In the audio domain it could be interesting to develop tools for development of signal processing graphs that could be interpreted and performed in other applications, for example generating Faust [22] or Max/Gen∼ DSP code.

There are many possibilities for the future development of SYMBOLIST, and so far it seems that the framework is providing a solid ground for the creation of new authoring environments. In a way, SYMBOLIST is a meta-environment, an application that aims to ease the process of creating new authoring environments. Like the creation of a new instrument, the challenge then is to work through the difficulties of creating the instrument, so that it can be learned and used for the creation of new kinds of art.

14. REFERENCES

[1] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193–200, 2005.

[2] 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.

[3] P. Manning, "The oramics machine: From vision to reality," *Organised Sound*, vol. 17, no. 2, pp. 137–147, 2012.

[4] G. Marino, M.-H. Serra, and J.-M. Raczinski, "The upic system: Origins and innovations," *Perspectives of New Music*, pp. 258–269, 1993.

[5] R. Gottfried, "Studies on the Compositional Use of Space," IRCAM, Paris, France, Tech. Rep., 2013.

[6] ——, "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.

[7] 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 – OpenSource Software Competition*, Scottsdale, AZ, USA, 2011, pp. 743–746.

[8] R. Gottfried and J. Bresson, "Symbolist: An open authoring environment for end-user symbolic notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'18)*, 2018.

[9] R. Gottfried and G. Hajdu, "Drawsocket: A browser based system for networked score display," in *Proceedings of the International Conference on Technologies for Music Notation and Representation–TENOR 2019*. Melbourne, Australia: Monash University, 2019, pp. 15–25.

[10] G. Hajdu, "Quintet. net: An environment for composing and performing music on the internet," *Leonardo*, vol. 38, no. 1, pp. 23–30, 2005.

[11] A. Freed, D. DeFilippo, R. Gottfried, J. MacCallum, J. Lubow, D. Razo, and D. Wessel, "o.io: a Unified Communications Framework for Music, Intermedia and Cloud Interaction." in *Proceedings of the International Computer Music Conference (ICMC'14)*, Athens, Greece, 2014.

[12] 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.

[13] 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.

[14] T. Baca, J. W. Oberholtzer, J. Treviño, and V. Adán, "Abjad: An open-source software system for formalized score control," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation-TENOR2015. Paris: Institut de Recherche en Musicologie*, 2015, pp. 162–169.

[15] 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.

[16] J.-L. Giavitto, J.-M. Echeveste, A. Cont, and P. Cuvillier, "Time, timelines and temporal scopes in the antescofo dsl v1.0," in *Proceedings of the International Computer Music Conference (ICMC)*, Shanghai, China, 2017.

[17] E. Ellberger, G. T. Perez, J. Schuett, G. Zoia, and L. Cavaliero, *Spatialization Symbolic Music Notation at ICST*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2014.

[22] https://faust.grame.fr/

TEXTURAL COMPOSITION IN 3D ENVIRONMENT THROUGH SWARM ALGORITHM

Wander Vieira Rodrigues
NICS/Unicamp
Campinas, Brasil
wanvrod@gmail.com

Walber de Macedo Rodrigues
CIn/UFPE
Recife, Brasil
wmr@cin.ufpe.br

ABSTRACT

The presented project uses algorithms based on bird flock behavior to generate sound masses in a 3D environment through "Cave" and sound spatialization in 8.1 Surround. The behavior of animal groups in agglomeration makes us lose the individual's perception in isolation to force us into the whole. Part of this process is identical to listening and composing sound masses in the textural music 20th and 21st centuries. Thus, we want to apply the swarm behavior algorithm to generate sound masses in real time, allowing contemplation, creation of musical material, and interaction through digital immersion.

1. INTRODUCTION

This project deals with creating an environment for projecting images in three dimensions with musical interaction in 8.1 surround. In this environment, the user can act with digital swarms of flocking birds, associating each individual to sounds produced by a human agent immersed in a 3D chamber to generate sound masses in real time.

Textural Composition in 3D Environment through Swarm Algorithm starts from the following steps: 1 - immersion in digital space; 2 - generation of digital birds; 3 - sound capture, interaction, and overlay; 4 - sound and visual results. The developed environment is available at GitHub[1]

To develop this concept, we need to understand that the behavior of agglomeration of animals has called humanity's attention since ancient times, be it herds, fish shoal, swarms, or flocks of birds. From a visual point of view, the behavior pattern of these beings makes us lose focus on the individual animal and start paying attention to a larger body that is the agglomeration. This topic has been the focus of current researchers who strive to model these behaviors due to the complexity generated and efficiency because of the factors resulting from this process, such as low collision rates between individuals, protection from predators, and long-distance travel.

[1] https://github.com/walbermr/TEXTURAL-COMPOSITION-IN-3D-ENVIRONMENT-THROUGH-SWARM-ALGORITHM

[1] argues that a swarm is a large number of homogenous, simple agents interacting locally among themselves and their environment, with no central control to allow a global behavior to emerge.

[2] takes the topic further by focusing on the individual: A swarm particle is a stochastic optimization approach modeled on the social behavior of bird flocks. The PSO (Particle Swarm Optimization):

> Particle swarm optimization (PSO) is a stochastic optimization approach modeled on the social behavior of bird flocks. PSO is a population-based search procedure where the individuals, referred to as particles are grouped into a swarm. Each particle in the swarm represents a candidate solution to the optimization problem. Each particle through the multi-dimensional search space in a PSO system, adjusting its position according to its own experience and neighboring particles. [2, p. 49]

Commonly associated with the concept of Computational Intelligence or Artificial Intelligence, the application of these algorithms, according to [2] again, brings satisfactory results in telecommunications network optimization, graph coloring, quadratic allocation problem, clustering optimization, and structural optimization, or as also advocated by [1]:

> Swarm-based algorithms have recently emerged as a family of nature-inspired, population-based algorithms capable of producing low-cost, fast, and robust solutions to several complex problems. [1, p. 2]

This project is inspired by Multimodal generative installations, and the creation of new Art form based on interactive narratives, the one created and developed by [3]. However, unlike the Generative Multimodal Installation, which uses compositional processes and generative designs linked to a scope of mediated perception, our project uses sound samples produced through improvisation by an immersed human agent feeding the system based on the representation of a bird flock modeled algorithmically.

Practically speaking, the moment the individual is immersed in the Digital Interactive Chamber, he will be guided to produce sounds of his choice in a pre-established or random way, using a musical instrument, voice, body percussion sounds, electronic oscillators. He will be free to choose and not need any formal musical experience. Then the Interactive Chamber system will then start extracting sound samples at a predetermined time and associating or allocating them to each swarm element or virtual bird. After these birds begin to swarm and form a flock, the result will be a mass of sounds that will move around in the sound space in 8.1 surround, projecting the swarm's behavior into the digital environment.

To better understand our goal, we must understand computer modeling and sound mass concepts, Section 2, and Section 3, respectively.

2. SOUND MASSES

Sound Mass is a concept associated with 20th and 21st century Textural Music. [4]. His research dealt with the SM approaching two perspectives: 1 - that linked to orchestral music and 2 - that linked to electronic music.

The Sound Mass is not only a musical resource to compose a work in contemporary style, but it is also a musical aesthetic. From a certain point of view, it is linked to the concept of *canone* in traditional music, resulting in the creation of patterns of sounds that can be called Textural Sound Blocks, as described in [4].

[5] discusses the canon by identifying it as an element of an original musical cell copied through isomorphism or the transformation of preserved information. The author further states that this property reveals itself in a recursive process, in which infinite structures can be defined as being the "*expansion of node after node*".

[6] already starts from the point of view that Musical Texture is based on the idea of sound clouds and argues that the sequence combination of these clouds in the mesostructure induces a process of statistical evolution.

He also states that a synchronous cloud generates metric rhythmic sequences at a grain density rate of 2 to 20 occurrences per second. When the initial density is not the same as the final density, but the variation of this rate is continuously controlled and increasing/decreasing, such a procedure generates an acceleration or deceleration effect. Otherwise, the initial and final density variation produces an irregular acceleration or deceleration for an asynchronous cloud. The cloud must be at a density above 100 grains per second to create continuous sounds.

[7] emphasizes these two distinct ways of using granular synthesis: synchronous and asynchronous, and in granular synthesis, the spacing of the sound grains radically modifies the generated texture. In the synchronous process, the grains are separated in the same time interval involving a particular linear relationship, while in asynchronous granular synthesis, the grains are inserted into the texture without a strictly linear relationship.

> This parameter specifies the temporal distribution of the grains, either synchronous or asynchronous. Synchronous generation means that one grain follows another in series, with the distance between the grain attacks determined by the density parameter. [...] Asynchronous clouds scatter grains at random time points within the specified boundaries of the cloud, at the specified density. Asynchronous clouds produce random and explosive effects. [6]

In a nutshell, we propose the shift of the analytical perspective from the score and orchestra to the sensory and cognitive process of the Sound Mass in the Textural Music *cf.* [8, 9, 10, 11].

Looking at our project through this lens, this means that for each bird that will make up our digital flock, we will have a speck and that the accumulation of birds will generate a cloud. Thus, we can determine a set of sound samples obeying asynchronous patterns. However, we still have another argument that starts from the sounds found in nature and is part of our sound universe. From a perspective that arises from the sounds found in nature that are responsible for our hearing development, [4] also defends:

> We can, then, relate the perception of individuality and sound grouping to the auditory experience when listening to the rain. The isolated sound of a drop of water hitting a surface, floor, or puddle becomes clear and recognizable as a single sound source. In this way, we can identify the sound and pinpoint its direction in space. However, as the number of drops hitting these surfaces increases (randomly), we stop perceiving them as drops and begin to relate them to the sound phenomenon of rain (which we can consider a sound mass). [4, p. 40]

To prove this point, the author [4] created an experiment associated with the phenomenon of rain that involves stochastic processes and gave a clear direction on the principles governing the formation and concept of Sound Masses.

By applying these concepts in our Digital Interactive Chamber, we will create the conditions for these principles and mechanisms linked to the Sound Mass Music, or Textural Music, to be incorporated from the behavior of flocks of birds since it has implicit models of stochastic processes.

We intend our interactive system to allow the individual immersed in the Cave to interact with birds flying through computer simulation and by supplying each bird with sound information using a musical instrument, its body percussion sounds, synthesized sounds, voice, or any other sound source, through prearranged or improvised ideas in real-time. Each bird will then be responsible for capturing a sound, or groups of sounds, with a time frame ranging from 2 to 5 seconds. After the sounds are captured, each individual bird will fly and seek a similar one to form a virtual flock. Consequently, the swarm's movement in space will create a mass of digital birds and several providing Sound Masses that will behave similarly to the spatially and dynamically arranged flock.

3. MATHEMATIC MODEL

Our objective is to create an immersive three-dimensional graphical environment through Unit3D. After the participant is positioned in the center of the Cave, the system will automatically perceive his presence and guide him in the interaction to generate sounds related to the bird swarm. As the system is gradually fed, there will be a waiting time for each sound produced to start the displacement of birds, implying that the environment will be fed by capturing sounds. The captured sounds will be incorporated into a particular bird of the flock, generating movement, and flocking simultaneously and gradually agglomerating based on the mathematical and computational modeling implemented. The established principle expects the captured sounds to have the same behavior as the song or call for each bird in the flock in space, keeping in mind that the captured sound samples will have variable durations and occur at the participant's pleasure. From the musical point of view, which is different from the sound mass compositions in G. Ligeti, W. Lutoslawsky, and others, instead of a static Sound Mass predicted and projected in the score, a dynamic Sound Mass is created in real time.

The user immersed in the Cave will be able to guide the swarm and generate more sounds based on the mass he or she has created or even interact with other people in the same environment. The Cave will work as a form of improvisation for experienced musicians or participants with some musical experience. Whereas, for amateurs, it will work as a playful experience that can provide a musical moment different from the one traditionally offered.

For the algorithm representing the flock of birds, we will have, as described by [12], the following mathematical model.

A *flock* B of birds can be represented by a set containing n birds b_i.

$$B = [b_i, i = 0,1,\dots,n-1] \quad (1)$$

(1) shows the definition of a flock.

Each bird has two randomly generated attributes and three related forces. The attributes are the position p and speed $\vec{v}$. Its forces are separation, cohesion, and alignment.

To visually recognize and discriminate the elements around the bird (isolated element) in the simulation, consider the birds' field of view. In this model, it is assumed that all birds have the same field of vision, and for this reason, the field is not a particular attribute of each individual bird. Since the two attributes of the birds (position and speed) vary during the simulation, this field of view will be represented by a sphere of radius e and the set of birds seen by another bird that is V_i, defined in Equation (2).

$$V_i = \{b_j \in B; \forall b_j : |p_i - p_j| < e, j = 0,1,\dots,n-1; j \neq i\} \quad (2)$$

As Equation (2) shows, we have a set of birds within the bird's field of view.

Three main movements in a flock of birds can be observed: 1- separation, 2 - cohesion, and 3 - alignment. When combined and applied to each bird, these forces should guide them so that the flock does not disperse.

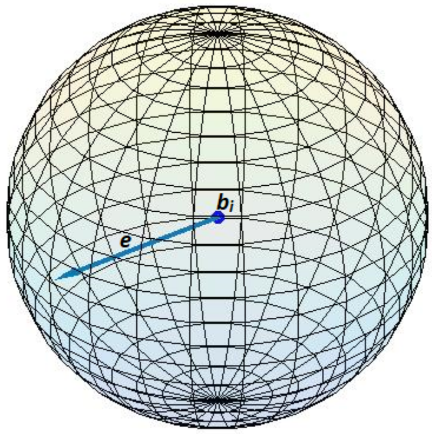


Figure 1: Field of View of the Bird b_i with radius$e$. Designed using the software MATLAB.

The separation force is essential because it prevents collisions between birds in the group. With its field of vision, each bird can identify other birds in the surroundings and estimate the movement needed to avoid a collision. This movement is called the separation movement, defined by the vector $\vec{s_i}$, and is the inverse direction of the sum of the vectors between the bird positions b_i and the birds b_j around them. Equation (3) defines the separation motion vector.

$$\vec{s_i} = -\sum_{\forall b_j \in V_i} p_i - p_j \quad (3)$$

The cohesive force $\vec{k_i}$ is responsible for the non-dispersion of the flock during the separation movement. It is defined as the vector between the bird b_i position, and the center of density c_i of all the other *m* birds visible by b_i. Equation (4) defines the center of density of visible birds, and Equation (5) shows the vector referring to the cohesion movement.

$$c_i = \sum_{\forall b_j \in V_i} \frac{p_j}{m} \quad (4)$$

$$\vec{k_i} = c_i - p_i \quad (5)$$

The force of alignment $\vec{m_i}$, defined in Equation (6), is what ensures that all the birds in the flock maintain similar speeds. This is observed in nature through the tendency of bird flocks to synchronize their speeds and stay close for extended periods. For a bird b_i, the alignment force is calculated using the average speed of the other birds within its field of view.

$$\vec{m_i} = \sum_{\forall b_j \in V_i} \frac{\vec{v_j}}{m} \quad (6)$$

Combining these three forces ensures that the flock maintains its unity, with close speeds between birds and a low number of collisions between birds. This combination results in the updated speed $\vec{v_i}'$ of a bird. Each force has a related coefficient that represents the share of that force in the final velocity. The updated velocity is defined in the Equation (7), shows here the vector referring to the updated speed of the bird.

$$\vec{v_i}' = \vec{v_i} + S \cdot \vec{s_i} + K \cdot \vec{k_i} + M \cdot \vec{m_i} \quad (7)$$

The next position p'_i that a bird b_i will assume is calculated by adding your current position to your speed multiplied by a time interval Δt. Equation (8) shows the point referring to the updated position of the bird.

$$p'_i = p_i + \Delta t \vec{v_i} \quad (8)$$

Every sound sample on our system must be a periodic signal, represented by a function x in the time domain t, represented by the function $x(t)$. Signal periodicity is defined as any point t in time to have a repetition period T_0, causing $x(t) = x(t + kT_0)$ where k is any integer.

4. MATHEMATIC STATISTIC

The swarm design has three main features, 1) acquisition of a sound sample that is repeated intermittently, 2) several samples are played together, and 3) the samples are arranged in *boids* that behave according to a bird swarm algorithm giving spatialization and providing clutter to the sounds.

Taking a sound sample y. any a period of T_1 seconds, where T_1 is an integer, will project on the swarm a sample repeated for an infinite time or interruption of the environment's execution. Thus, the sound signal will be set for the entire time dimension every time T_1 seconds in which it repeats. This observation fits the definition of a periodic signal, so the sample is periodic at kT_1seconds, where the k is an integer number.

In the theory of Signals and Systems the sum of two periodic signals, with periods T_0 and T_1, is also a periodic signal since the ratio $\frac{T_1}{T_0}$ be a rational number $\frac{M}{N}$ where M N are non-divisible integers.

As new sound samples are inserted into the swarm, the existing periodic sounds are added together to create a different sound signal that is also periodic. This other unreleased signal is periodic since the periods T_1 ad T_0 are integers, and division between two integers does not yield rational numbers. The period of this new sign is defined as $\frac{T_1}{T_0} = \frac{M}{N} \rightarrow MT_1 = NT_0$, that is to say, $T_2 = MT_1 = NT_0$. Each new signal added will always respect this rule, generating new periodic signals.

5. THE LISTENING PERSPECTIVE

The Project in question here deals with sound not as a pure matter dissociated from an artistic context but as a sound construction molded in the function of materials that carry minimalist traits and are allocated, superimposed, and spatialized by the algorithm of agglomeration of animals, specifically birds. At the same time, there is no intention to use or sculpt musical structures as traditionally established. Our idea is built on the concept that it is not the sound that generates the beauty we associate with a landscape of birds in flight but the condition that each bird producing sounds and agglomerating creates a sound and visual phenomenon that has acoustic impact and generates the beauty, or any reaction, that has always caught the attention of humankind.

[13] launches the perspective of sound agglomeration from the point of view of rain and the nuances between the sonorities implicit in it, as well as the different ways we perceive and interpret it (In [4], we took a little far in depth the issue of similarity between sound structures such as rain and sound mass). Furthermore, both rain and other phenomena of nature are responsible for forming our listening. [14] has a similar approach when dealing with the relationship between sound perception, natural phenomena, and the relationship with the development of our listening. He states:

> The first reason comes from the observation and discovery of rare or everyday sound phenomena that nature or society offer us. Thus, for example, during summer in the countryside the song of the cicadas, which rocked humanity and its poets, imposes itself and invites us to join the structure of this event, if only out of childlike curiosity "how it's made" that every adult keeps for the happiness of humanity. Then comes the desire to reconstitute a similar event, no longer with cicadas but with other sound means, with orchestral instruments or with machines. And this desire goes so far as wanting to modulate, according to invention, the sound event inspired by the song of thousands of cicadas. How to get there? By appealing to the internal logic of this natural phenomenon [14].

Furthermore, it seeks a mathematical understanding of the phenomenon:

> This logic is that which also governs the movements of the molecules of a gas and physics has already given its answer in the "kinetic theory of gases", which is a stochastic probabilistic theory. If the observation of the song of cicadas had been made by a mathematician composer before the creation of the "kinetic theory of gases", it is certain that the study of the song of cicadas would have led him to the discovery of similar laws. Once the abstract structure of the mass event made up of thousands of elements has been defined, it can be used to condition a mass of pizzicati or any occasional bowing of the classical string orchestra." [14].

[13] speaks of the lo-fi soundscape due to sound congestion or saturation resulting from the emergence of new sounds. He states:

> The Industrial Revolution introduced a multitude of new sounds with unhappy consequences for many of the natural and human sounds which they tended to obscure; and this development was extended into a second phase when the Electric Revolution added new effects of its own and introduced devices for packaging sounds and

> transmitting them schizophrenically across time and space to live amplified or multiplied existences [13].

And continues:

> Today the world suffers from an overpopulation of sounds; there is so much acoustic information that little of it can emerge with clarity. In the ultimate lo-fi soundscape, the signal-to-noise ratio is one-to-one, and it is no longer possible to know what, if anything, is being listened to. This, in summary, is the transformation of the soundscape [13].

Nowadays we are facing a panorama that offers the possibility of a better understanding of what gave rise to the consolidation of an aesthetic that breaks with traditional concepts of music and focuses greater attention on sonority.

Based on this perspective, we intend to offer the participants of our immersive project a sound experience that will be built based on the interaction with bird flocking algorithms. In our system, the sound material produced will result from the resources available to the human agent, be it an entirely non-musically literate person or even an excellent and experienced musician. The experience derived from this digital medium will allow a different vision of what the immersed human agent is used to observing from his own experience in his daily life and sound experiences, composing sound materials with conceptual support in the masses of sounds coming from the music of the 20th and 21st centuries.

6. IMPLEMENTATION OF THE SYSTEM

The project presented here was initially developed until the implementation phase and the first tests.

Using the Cave set up at the NICS/Unicamp lab at Campinas/São Paulo, we were able to launch the projection of the Boids and perform experiments with sounds produced by agents immersed in the digital environment. In this initial phase, we did not build the birds' bodies. We used spheres as representatives of each bird of the flock, shown in Figure 2:

Figure 2: photo taken from a *boid,* created in the system developed for sound mass composition.

Before deployment, the system was previously executed on a computer using three screens, then projected onto a larger screen to generate the 3-D environment.

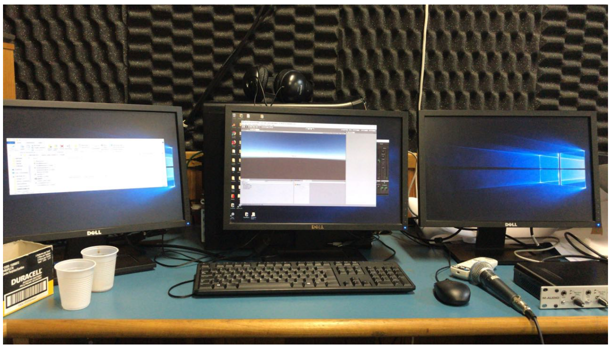


Figure 3: Screen computer where the 3D plot was generated.

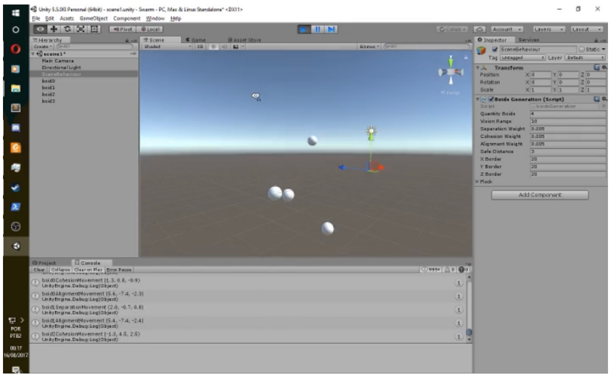


Figure 4: Projection screen in Unity 3D.

Figures 2, 3 and 4 present the system running on the computer screen. Figures 5 and 6 show the Cave.

Figure 5: Projection Screen of the digital Cave.

Next, we have the projection of the swarm on the Cave:

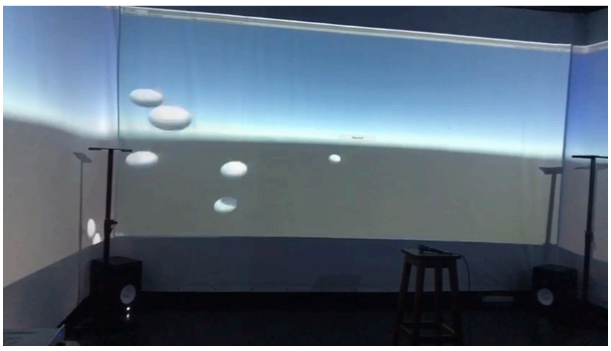


Figure 6: The Cave working with images and sounds with surround 8.1.

In the presented projection we can observe a group of 8 elements that had their sound representation agglomerated into images and sounds within the principles of sound mass.

7. CONCLUSION

The exposed project obtained tangible results from both points of view: visual and sound experience. We worked with sounds generated with body percussion sounds, clapping, hissing, and musical instruments such as the xylophone during this project phase.

The process of sound mass generation, that is, the loss of the perception of the sound unit for the agglomeration in an extrapolation of understanding in a larger dimension, was perceived as more sound elements were generated and composed of the swarm.

The unfolding of this project involves:

- The composition of melodies by collecting melodic, harmonic, and sound masses.
- The experiment may give rise to an ecological approach to birds for students.
- Analysis of the sound extrapolations resulting from the interaction in the digital environment.
- Music/Sound Installation to be presented in artistic presentations.
- The neuroscientific study of the agglomeration process of sound and image. This study is currently in progress but without publications and looking for partnerships.

In this way, we intend to advance with the research and seek partnerships to disseminate and improve the idea.

8. REFERENCES

[1] H. Ahmed and J. Glasgow, *Swarm Intelligence: Concepts, Models and Applications*. School of Computing Queen's University, 2012.

[2] A. P. Engelbrecht, *Computational Intelligence: An Introduction*. John Wiley & Sons, 2007.

[3] J. Manzolli. Multimodal generative installations and the creation of new Art form based on interactivity narratives. *XVIII Generative Art Conference*, [Online] Available: https://www.gentiveart.com/ga2015_WEB/Multimodal-Installation_Manzolli.pdf

[4] W. R. Vieira, Composição Textural Assistida por Computador. PhD Thesis, State University of Campinas (Unicamp), Campinas, 2016.

[5] D. R. Hofstadter, *Gödel, Escher, Bach: an Eternal Golden Braid*. Baic Books, 1999.

[6], C. Roads, *Microsound*. MIT Press, 2001.

[7] T. Opie. Sound in a NutShell: Granular Synthesis, An overview of granular synthesis and the techniques involved. [OnLine] Available: http://www.granularsynthesis.com/hthesis/hthesis.html.

[8] R. Bader, Berechnung Fraktaler Strukturen in den Etüden für Klavier von György Ligeti, *Musikwissenschaftliches Institut Hamburg*. 2003.

[9] M. Chemillier, György Lgeti et la Logique des Textures. *Analyse Musicale*, n. 38, 2001, pp. 75-85.

[10] J. Clendinning, Jean Piper. The Pattern-Meccanico Compositions of György Ligeti. *Perspectives of New Music*. v. 31. 1993, pp.192-234.

[11] A. M. Roig-francoli, Harmonic and Formal Processes in Ligeti's Net-Structure Compositions. *Music Theory Spectrum,* v. 12, n. 2, 1995, pp. 242-257.

[12] W. R. Craig, Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics 21 (4)*, 1987. pp. 25-34.

[13] M. Schafer, *Our Sonic Environment and the Soundscape: The Tuning of the World.* Destiny, 1994.

[14] I. Xenakis, *Formalized music: Thought and mathematics in composition*. Pendragon Press, 1992.

A FEW THOUGHTS ON POLYMORPHISM IN DIGITAL SCORES

Georg Hajdu, Konstanina Orlandatou, James Tsz Him Cheung
Hochschule für Musik und Theater
Hamburg, Germany
{georg.hajdu,konstantina.orlandatou,tsz.him.cheung}@hfmt-hamburg.de

ABSTRACT

This paper is motivated by the phenomenon of polymorphism in graphic notation, a notion introduced to the discourse of graphical composition by Greek composer Anestis Logothetis. It refers to the reading of a graphic score in which alternative paths can be taken by a performer. The reading can either be synthetic /global or analytical/local with intermediary levels. We are contrasting Logothetis' concept of polymorphism with analogous phenomena in molecular biology and look at the paradigm shift leveraged by digital technologies where machine and hybrid readings ought to be taken into consideration. Examples of current practices are given for live, extended reality and hybrid scenarios. The paper finishes with an outlook on how AI might eventually become another game changer.

1. INTRODUCTION

We would like start by quoting Logothetis: "What fundamentally differentiates graphic notation from traditional notation is the afore mentioned polymorphism, which clearly enables all performers to retain their subjective reaction times. The composer takes into consideration the divergences of the different performers in composing and expects a certain degree of surprise through the new formalization of musical form in every performance" [1].

The application of the term polymorphism to graphic notation is attributed to Anestis Logothetis (1921-1988) [2]. Inspired by works of Cage and Brown, he established a graphic notation system which was to "was to broaden the musical script/code and not to provide a score with illustrative elements" [3]. To this aim Logothetis distinguished three categories of symbols: 1. pitch symbols, 2. association factors and 3. action signals, a concept put forth in his main essay *Zeichen als Aggregatzustand der Musik* [1].

Much of his work is based on the idea that different readings (German *Lesarten*) of the same text are conceivable. We can see this at a basic level in his text compositions in which Logothetis creates subtexts by overlapping and slicing words into syllables and phonemes. In *Sisyphos – Stein* [4] for instance, he used a passage from Plato's dialog Kratylos for a reading that includes multiple languages as well as the semiotics of shapes (see Figure 1). As Hartmut Krones points out in the introduction of his book [1], the word ἄνθρωπος (man) can also be interpreted as ἄ = ah! (exclamation), θρω = droh! (threat) and πο = po! (the human behind in German) where ω also graphically represents the shape of the body part in question. Logothetis thus used a fragmentation technique to allow for alternative readings, and thereby establishing polyphormism on the level of the rendering of literary texts.

Vickery, co-author of the Decibel ScorePlayer, developed a similar concept which he refers to as *rhizomatic*. Based on the ideas by Deleuze and Guatarri, he explored "the development of rhizomatic musical scores that are arranged cartographically with nodal points allowing for alternate pathways to be traversed" [5].

2. THE MORPHOME

The English translation of title of Logothetis' essay is "Signs as Aggregate State of Music". As Logothetis clearly makes a reference to physics and the natural sciences, one wonders whether this also has a discursive aspect directed at his countryman Xenakis who since the 1950s was fascinated by translating the statistics of molecular (Brownian) motion into music and described much of that in his seminal book *Formalized Music* [6].

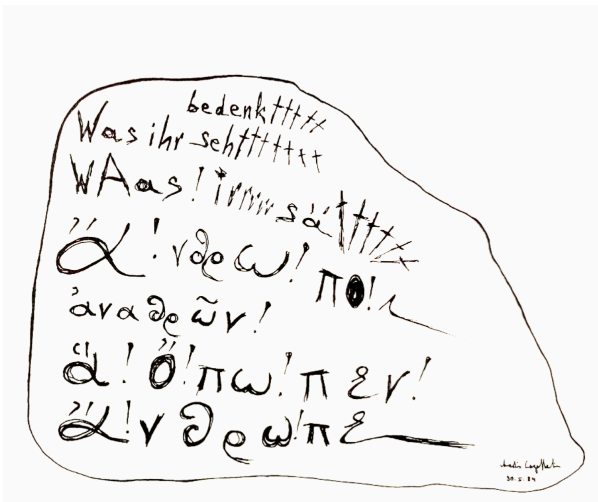


Figure 1. Score of Sisyphos – Stein by Anestis Logothetis.

But Logothetis might have as well turned to the then budding field of genetics as he could have discovered both there: signs and a molecular structure that enables life via transcription and translation of genetic code. In genetics, the analogy goes even deeper. We find a precise sequence of nucleotides (the basic building blocks of DNA and RNA) akin to his pitch notation, a molecular machinery operating on sequences analogous to his action signals and finally a complex system modulating the expression of code comparable to association factors. We also encounter polymorphism. For instance, for the major histocompatibility complex (MHC) more than 32000 variants, called alleles, are known[1]. It resembles also a phenomenon called *gene overlap* where a sequence can be read in various ways[2] (Figure 2).

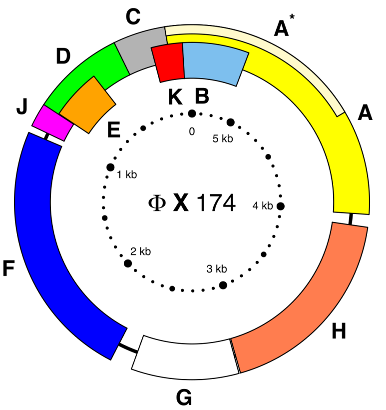


Figure 2. The genetic code of the bacteriophage ϕX 174 exhibiting gene overlap. © Emmanuel Douzery

Extreme cases can be found in out-of-phase overlap where alternative readings can be shifted by one or two nucleotides, leading to a different gene product, or in sense-antisense gene overlap where the usually silent antisense DNA (the complementary DNA strand that doesn't get translated into proteins) also encodes a protein [6].

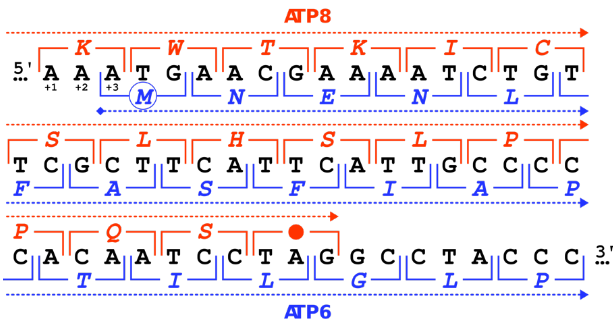


Figure 3. Out-of-phase overlap in DNA transcription resembling Logothetis concept of polymorphism. © Emmanuel Douzery

Reading in two directions is also a requirement in Logothetis' composition Dynapolis [8] which actually is inspired by the layout of a city (Figure 4) yet bears similarities to the aforementioned cellular mechanisms of transcribing and translating genetic code. In analogy to its definition in biology (mapping and classification all the morphological features of species), we are proposing *morphome* as a term referring to *the totality of all possible readings of a score.*

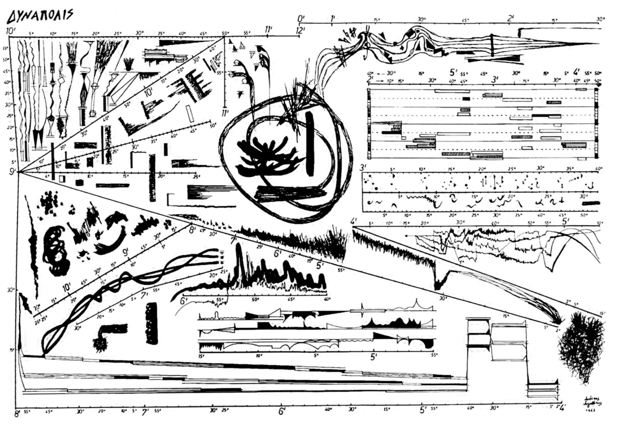


Figure 4. Score of Dynapolis by Anestis Logothetis.

As DNA serves as a sequence of codons which carry meaning, the field of semiotics has been expanded to also include genetics. While it's outside the scope of this presentation we would like to refer to the article "The Linguistics of DNA" by David B. Searls[3].

3. HUMAN SCORE READING

In the context of creating his classification, Logothetis extensively studied the nature of reading graphic notation [1]:

"It was the time when I was intensively occupied with problems of musical recordings and realized that graphic elements can be grasped in three ways if one wants to use them for musical purposes: They can symbolize a thing by signifying it. Then they can evoke associations, and finally they can signal commands."

As with any form of reading as a cognitive activity, the process of decoding of text (I use text in a broader sense) is hierarchical. Different brain regions (nodes) are in charge of the decoding of it [9]. This division of work allows for the anticipation of meaning and providing robustness in case of error. We all know *exmaples wehre chactres r swtichd nd omittd* without affecting the intelligibility of the sentence. Such processing of hierarchically arranged information of has been captured by Schenkerian analysis and is also at work when reading Logothetis' scores.

In analogy to gestalt perception of tones which are referred to as analytical (resolution of individual partials) and synthetic (focus on the fundamental) I shall call the reading focussing on the graphics as a whole (and primarily being used to create associations for an

[1] https://en.wikipedia.org/wiki/Gene_polymorphism

[2] Gene overlap was first discovered in the bacteriophage (a virus targeting bacteria) ϕX 174. Its genome being a little larger than 5000 nucleotides codes for 11 genes (A through K), eight of which overlap with other genes by at least one nucleotide.

[3] https://www.jstor.org/stable/29774782?seq=13#metadata_info_tab_contents

improvisatory approach) *global* and *synthetic*, whereas the fine-grain resolution of a pitch symbol or association factor is rather *local* and *analytical*. In this context, it may be useful to distinguish between the interpretation of a score (synthetic) and execution of it (analytic).

Baveli and Georgaki have developed a taxonomy that reflects this hierarchy which also includes an intermediary stage representing a synthetic reading on a local level, as in the case of action signs which aren't fixed on a temporal level [3].

Local resolution (in terms of speed and accuracy) of course isn't a quality per se and depends on familiarity with the terrain in question as hinted in the following distorted image of a passage from Scriabin's piano piece *Vers la flame* op. 72 (Figure 5).

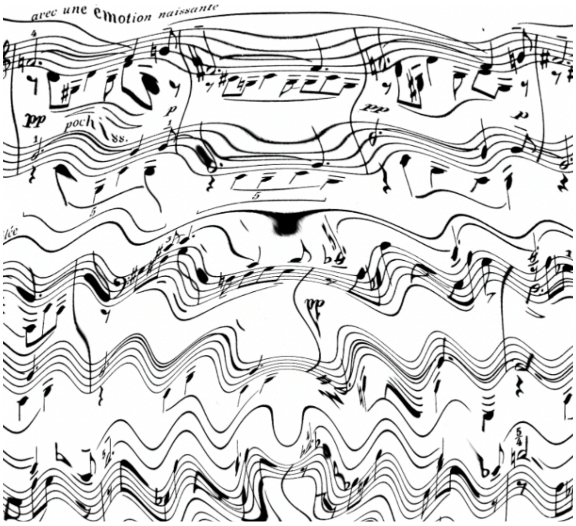


Figure 5. A page from a piano piece by A. Scriabin with distortion applied spurring a different reading and agency in comparison to the original score.

Due to impaired local resolution caused by the distortion, performers are more likely to fall back into an associative behavior and focus on the interpretation of the contorted lines rather than trying to accurately execute the musical events encoded in the score.

This friction of local vs. global, analytical vs. synthetic is what makes Logothetis' approach to graphic notation so enticing: On the one hand, we encounter action signals which carry an immediate meaning appealing to the embodied cognition of the interpreter and on the other hand, we find precise quantifiable pitch notation which can be further subjected to music-theoretical taxonomies.

4. MACHINE AND HYBRID SCORE READING

Zeichen als Aggregatzustand der Musik was published in 1974 (perhaps as a counter draft - in German "Gegenentwurf" - to Xenakis' Formalized Music) when Logothetis was already 53 years old and computer composition was still in its infancy. Xenakis, an engineer by training, was at the avant-garde of computer-based composition and instigated the development of the UPIC system, completed in 1977. It allowed composers to draw partial tracks on a digitizing tablet, to be sonically rendered by the associated software and hardware. But not all composers of this generation were willing or in the position of following on this path as it also meant to leave the safe boundaries of paper, pencil and (if at all) pocket-calculator and to embark on a journey with potentially questionable results. In his essay *Die Geschenke meiner Umgebung anhand der Frage "was denn nun Musik sei"* (the gift from my environment vis-à-vis the question "what's music after all"), Logothetis expresses his dissatisfaction, a bit circuitously though, with the way musical structures were represented graphically by software at the time [10].

"Many of these devices today register their sound derivatives in various graphic ways, including that of the 5-line staff, and print what is registered, but this leads to an obscuring of the compositional notation concept. The text appearance, which invites to produce sounds, becomes an inventory through the recording of already produced sounds, and the notation activating the musical practice becomes a programmed automatism which can also be triggered randomly by pressing keys and does not presuppose any compositional intellectual work. The sound image is no longer necessary for composing, at most for control purposes. This state of affairs can lead to great perceptual complications in the evaluation of musical recordings and necessitates the emergence of new competencies."

He was most likely referring to piano roll and standard notation representation of MIDI events afforded by 1980s software such as *Notator* by C-LAB or *Cubase* by Steinberg and bemoaning the cognitive divide between the text appearance (*Schriftbild* in German), its reading as a programmed automatism and the sonic outcome, while at the same time mandating the development of new competencies to overcome this very divide. In computer music during the last decades of the 20th century, *reading* has become somewhat synonymous with *sequencing* where in most cases, a linear score either in standard Western or piano roll notation is played back while a play head is moving across the screen, or the score itself moves under a static play head. Here, every note represents a MIDI command to be executed at a particular time defined by its position in the score and a few accessory elements. Logothetis passed away before computers and software applications became powerful enough to offer anything close—in the graphic domain—to what he had already achieved with traditional means. And obviouosly, using a computer only makes sense if there is an added value. This value can be found in the notion of *computer reading* and this ought to be substantially more than just reading a graphic score off a screen created with a mouse on a screen instead of being drawn on paper with a pencil or using a piano roll or 5-line representation of MIDI events. One of the first applications that broke with this paradigm is IanniX [11], a non-linear sequencer whose name is an unequivocal reference to Xenakis and his UPIC system [12].

To achieve the kind of polymorphism that Logothetis would have expected from a digital score, we also need to first define how the score reading is supposed to take place when performed by the machine alone. Here we can differentiate between three levels concerning the encoding of a score:

- bit map
- vector graphics
- graphical representation of an underlying musical data set

In the first case, pixels are commonly mapped to time and pitch such in a left-to-right linear reading such as in some pieces by Clarence Barlow and the partial tracks created with the UPIC (see section 6 for a brief discussion of more complex mapping facilitated by AI). The second case requires a semantic mapping between graphic commands and music events and forms the basis of the Symbolist software by Rama Gottfried where left-to-right linear reading is also standard [12]. The third case is typical for applications such as InScore [13] where readings can be either linear or non-linear but can also be achieved by gradually moving up the three levels by applying pre-defined rules while progressing from bit map to the musical data set.

A computer-based system needs to imbue a graphic element with some meaning pertaining to the environment in which the reading is to take place and turned into machine agency. A beautiful example for a polymorphic score with is *Cube with Ribbons* by Simon Katan which has a cursor travelling down a line to which various events are attached like ribbons until it encounters a junction allowing it to take an alternative path (Figure 6).

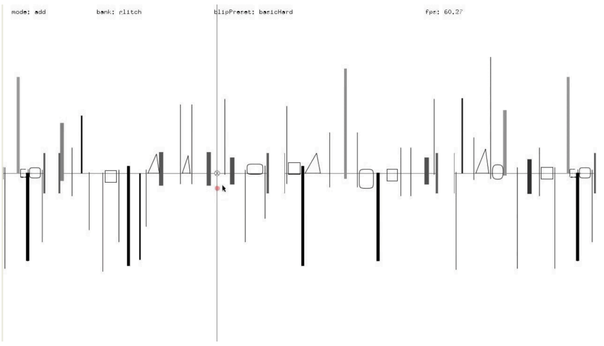


Figure 6. Screenshot of *Cube with Ribbons* by Simon Katan.

As one can see in the video on Vimeo[4], user intervention can change the likelihood for such things to happen. Like in a score by Logethetis, such an approach implies that certain constellations may never sound or at least not in a particular rendering of the score. The score, therefore, is a field of possibilities rather than just an unambiguous linear text. We have already referred to this as the *morphome*. Such fields can be either hard coded in the score in terms of alternative routes to be chosen by the performer or the basis for real-time notation in which the computer makes those choices, ideally in response to the performer [15].

If we factor in human reading, we end up with a hybrid scenario where the digital score is executed by the machine and interpreted by the human at the same time. A hybrid of a simultaneous human and computer reading can therefore be achieved, for instance, when the computer deals with the necessary local and analytical reading of details such as the exact pitch or sample to be played, while the human user can focus on the global and synthetic interpretations concerning the more associative aspects of the piece, about when and how these signs should be triggered as interface elements.

5. UMIS: UNIFIED MUSICAL INSTRUMENT SURFACES

The Decibel ScorePlayer [16] is a piece of software which the typical right-to-left linear scrolling paradigm but also has a mode that allows non-linear reading by employing moving shapes to highlight areas of a score. A performer can thus *react* directly to the score, either scrolled or presented page-wise (which is preferable presentation mode when the screen needs to be touched). However, when *interaction* is required, a graphic element needs to function as a GUI element. For instance, in MaxScore, a package for standard Western, microtonal and graphic notation co-authored by Hajdu and Didkovsky, any element created with its Picster drawing tool can be associated with an expression to be executed and interpreted by the Max and/or Ableton Live host environments [17].

An expression thus forms a sign in which the element takes on the role of the signifier and the expression that of the signified. This can either be a linear relationship where a curve gets interpreted as a trajectory and translated into a breakpoint function or an abstract one where the element becomes a graphical representation of some parameter settings. In MaxScore, these signs can then be executed according to their temporal order or serve as interface elements to be manipulated by an interpreter at his or her own will. Elements can be left out and non-linear, rhizomatic routes taken.

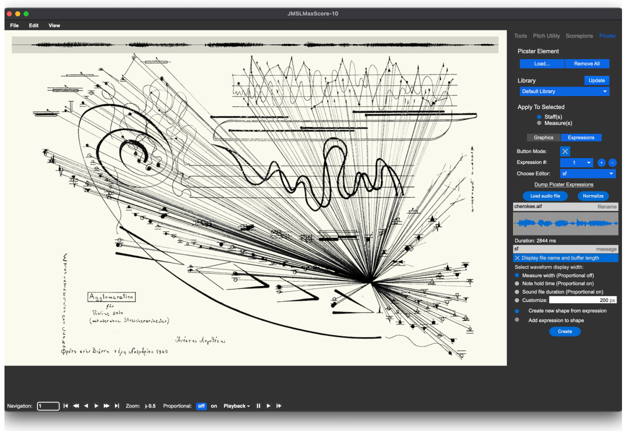


Figure 7. A score by Logothetis rendered in the MaxScore editor with an additional waveform added as

[4] https://vimeo.com/36888504#_=_

an example of how an existing graphic score can be supplemented with additional expressions.

A score hence functions as a controller in addition to being a graphic gestalt. In MaxScore, this type of interaction is toggled by the "buttonmode" message which allows arbitrary Picster elements to act as buttons. Figure 8 shows an example for a GUI element auto generated from parameter settings aiming to control the real-time generative program DJster [18].

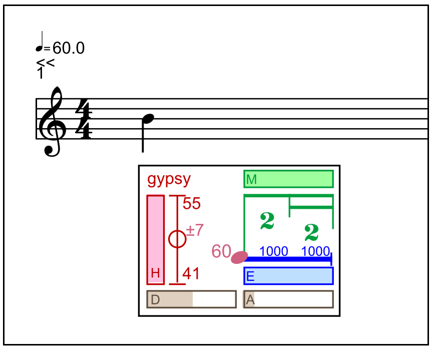


Figure 8. An example for an Picster expression, auto generated from DJster parameter settings. The code for this was created by Cheung.

By integrating a camera and projector into the body of the instrument, Sello [19] has converted a timpani and a tom-tom into hybrid instruments he named Hexenkessel and Hexenkesselchen, resp. (Figure 9) where the membrane serves both as (a) a score display and (b) a touch-sensitive controller in addition to being playable by mallets (c). Sello and Hajdu coined the term UMIS (short for *unified musical instrument surface*) to capture the triple nature of such hybrid instruments. Using OpenGL texture mapping, scores can be bent into a circular area to be projected onto the membrane of the instrument.

To achieve this in MaxScore, we connect the matrix outlet of the maxscore.bcanvas abstraction (containing the JavaScript object jit.pane.js, capable of rendering a score directly to a Jitter matrix) to a jit.gl.texture object and apply the texture to the circle via jit.gl.gridshape (Figure 10).

Figure 9. The Hexenkesselchen. © Jacob Sello

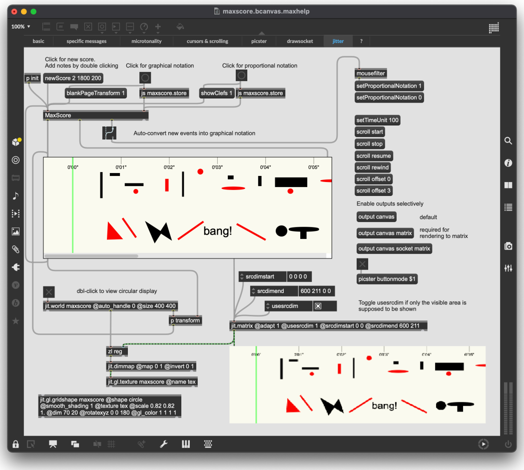


Figure 10. The Max patch in charge of generating and processing the Jitter matrix.

The score to be projected onto the membrane of the percussion instrument (Figure 11) can be played by tracking the position of the mallet and sending this information over the network to the computer serving the score. Due to their bending, a geometric transformation must be applied to calculate its Cartesian coordinates in order to identify the touched shape and trigger its expression.

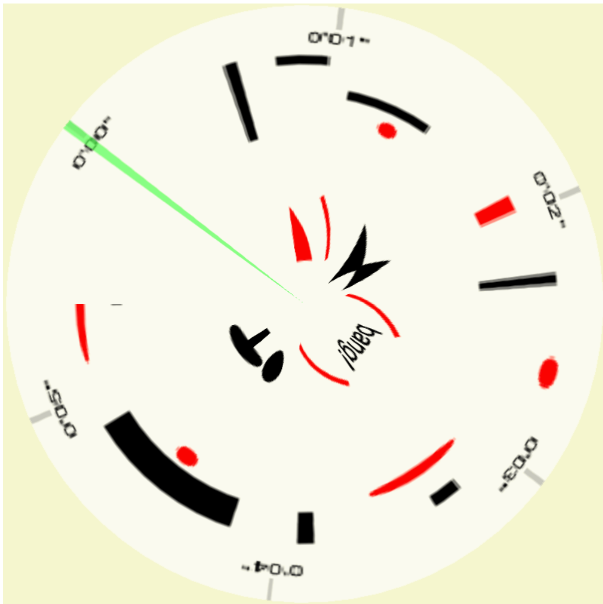


Figure 11. Circular score to be projected onto the surface of the Hexenkesselchen. This score can either be scrolled circularly or presented as a static page with its elements functioning as buttons to be struck by the player at will without necessarily following the given temporal order.

6. AUGMENTED AND VIRTUAL REALITY

We can easily imagine a scenario in which a performer or a conductor wearing augmented reality glasses can guide a performance without fixing a computer screen or a tablet. As a proof of concept, we have used Hololens 2 mixed reality glasses to interact with a score rendered in its browser via Drawsocket [20], which serves the score over the local network. The user interacts with the graphical elements in buttonmode through the virtual laser beam and thus executes the Picster expressions associated with them (Figure 12). While this approach is confined to a 2-dimensional plane, we can also conceive of a score as a three-dimensional arrangement of objects.

Figure 12. An ensemble can be directed by a conductor wearing Hololens 2 mixed reality glasses. In practice, the browser window is more transparent than it appears on the screenshot.

In her project Moving Sound Pictures, Konstantina Orlandatou deals with the question, whether it could be possible to consider art works of the 20th century - especially those of the abstract painters - as polymorphic graphic scores in Logothetis' sense? To this end, she turns paintings by famous artists such Wassily Kandinsky, Piet Mondrian or Kazimir Malevich into 3-dimensional spaces where the graphical elements of the painting become tangible objects. These objects emit sound upon tactile interaction. With the usage of controllers and a head-mounted display (VR), the user is able to grab these objects and generate music with his/her gestures. The user thus becomes the musician who interacts with a graphic score, but the score is the painting itself (Figure 13).

The way Orlandatou interprets the objects into musical elements align with Logothetis description of reading musical scores in the broader sense of polymorphism. Firstly, graphical elements become symbols of sounds. Every element has its own sonic characteristics based on its form. For instance, a triangle sounds edgy, but a circle may sound smooth. Secondly, the graphical elements as objects positioned in space can elicit associations. Their form or even colour can arouse emotional state or can resemble an object of daily routine. Finally, graphic elements become symbols of actions and commands. The interaction with the object is the action needed for making the object audible.

The potential of implementing graphic scores 3-dimensionally in a virtual reality environment is barely explored. A space, where the score is not only a linear text that has to be read in a specific timeline, opens a new path for creating holistic perceptual experiences in which the musician can directly interact with the graphic score.

Figure 13. Screenshot of a Moving Sound Picture project (Kazimir Malevich: Suprematism).

7. OUTLOOK: AI

Where we going to take this from here? There is a lot of talk about machine aesthetics and machine appreciation of art in the context of AI, but some of the current results are seriously wanting as the completion of Beethoven 10th symphony has shown[5]. Still, we foresee fascinating applications of AI in the context of machine reading of scores. Networks could be trained (e.g. by using eye-tracking data) to a corpus of graphic scores and learn how to interpret them. Once again, the local vs. global paradigm can be useful to define what the networks ought to be capable of.

Yet, we believe that for the foreseeable future, machine reading of graphic scores alone won't be the golden grail. Instead, hybrid readings which include both the human factor as well as the machine (be it as generators or interpreters) will be the most likely scenario, until a machine reading machine-generated scores while entertaining another one with machine music becomes a reality and an evolutionary advantage. Logethetis probably would have had some thoughts on this as well.

Acknowledgments

We would also like to acknowledge the Federal Ministry of Education and Research in Germany (BMBF), for their support of this research through the Innovative Hochschule: Stage_2.0 initiative.

8. REFERENCES

[1] A. Logothetis. Zeichen als Aggregatzustand der Musik. H. Krones, (Ed.), *Anestis Logothetis, Klangbild und Bildklang*, Verlag Lafite, Wien, 1998, pp. 32-71.

[2] Mongini, C. "Sign and Information: On Anestis Logothetis' Graphical Notations." *Deleuze and Contemporary Art*, 2010, pp. 227-245.

[3] M.-D. Baveli and G. Anastasia, "Towards a decodification of the graphical scores of Anestis Logothetis (1921-1994). The graphical space of Odysee (1963)," in *Proceedings of the 5th Sound and Music Computing Conference,* Berlin, 2008.

[5] https://www.telekom.com/en/media/media-information/archive/beethoven-s-10th-symphony-completed-by-ai-633060

[4] A. Logothetis, *Aus welchem Material ist der Stein von Sisyphos!?: ein multimediales Bühnenwerk*. Ges. zur Förderung Österr. Musik, 1995.

[5] L. R. Vickery, "Rhizomatic approaches to screen-based music notation," in *Proceedings of the International Conference on New Interfaces for Musical Expression,* Brisbane, 2016, pp. 394–400.

[6] I. Xenakis, *Formalized music: thought and mathematics in composition*. Second, revised English edition, with additional material translated by Sharon Kanach. Harmonologia Series No. 6. Pendragon Press, 1992.

[7] M. Xue, et al. "Sense-antisense gene overlap is probably a cause for retaining the few introns in Giardia genome and the implications," *Biology direct* 13.1, 2018, pp. 1-7.

[8] A. Logothetis. *Dynapolis*, for unspecified instrumental groups. Graphic notation. Universal Edition, 1967.

[9] G. Deco, M. L. Kringelbach, "Hierarchy of Information Processing in the Brain: A Novel 'Intrinsic Ignition' Framework," in *Neuron*, Volume 94, Issue 5, 2017, pp. 961-968.

[10] A. Logothetis, "Die Geschenke meiner Umgebung anhand der Frage 'was denn nun Musik sei'". H. Krones, (Ed.), *Anestis Logothetis, Klangbild und Bildklang*, Verlag Lafite, Wien, 1998, 78-137.

[11] Th. Coduys and G. Ferry. "IanniX aesthetical/symbolic visualisations for hypermedia composition," in *Proceedings of the 1st Sound and Music Computing Conference*, Paris, 2004.

[12] C. Hope. "From Xenakis's UPIC to Graphic Notation Today," in P. Weibel, L. Brümmer and S. Kanach (Eds.), Hatje Cantz. *Tempo* 74.294 (2020), pp. 105-107.

[13] R. Gottfried, "Symbolist Re-imagined: bidirectional graphic-semantic mapping for media notation authoring and performance," in *TENOR 2022: Proceedings of the International Conference on Technologies for Music Notation and Representation*, Marseille, 2022.

[14] D. Fober, Y. Orlarey, and S. Letz, "Inscore – an environment for the design of live music scores," in *Proceedings of the Linux Audio Conference* – LAC 2012, Stanford University, 2012, pp. 47–54.

[15] G. Hajdu, "Disposable music," in *Computer Music Journal* 40.1, 2016, pp. 25-34.

[16] C. Hope, A. Wyatt, and L. Vickery. "The Decibel ScorePlayer: New developments and improved functionality," in *Proceedings of the 40th International Computer Music Conference*, Denton, 2015.

[17] N. Didkovsky and G. Hajdu, "Maxscore: Recent Developments," in *TENOR 2018: Proceedings of the International Conference on Technologies for Music Notation and Representation*, Montreal, 2018, pp. 138-146.

[18] G. Hajdu. "Resurrecting a Dinosaur—the adaptation of Clarence Barlow's legacy software AUTOBUSK," in *TENOR 2016: Proceedings of the of the International Conference on Technologies for Music Notation and Representation*, Cambridge, 2016, pp. 181-186.

[19] J. Sello, "The Hexenkessel: A Hybrid Musical Instrument for Multimedia Performances," in *Proceedings of the International Conference on New Interfaces for Musical Expression,* Brisbane, 2016, pp. 122–131.

[20] R. Gottfried and G. Hajdu, "Drawsocket: A Browser Based System For Networked Score Display," in *TENOR 2019: Proceedings of the International Conference on Technologies for Music Notation and Representation*, Melbourne, 2019, pp. 15-25.

A MODEL OF RHYTHM TRANSCRIPTION AS PATH SELECTION THROUGH APPROXIMATE COMMON DIVISOR GRAPHS

Gonzalo Romero-García, Corentin Guichaoua, Elaine Chew
Sorbonne Université, Ircam, CNRS, Ministère de la Culture
Sciences and Technologies of Music and Sound Laboratory, Paris, France
{gonzalo.romero,corentin.guichaoua,elaine.chew}@ircam.fr

ABSTRACT

We apply the concept of approximated common divisors (ACDs) to estimate the tempo and quantize the durations of a rhythmic sequence. The ACD models the duration of the tatum within the sequence, giving its rate in beats per minute. The rhythm input, a series of timestamps, is first split into overlapping frames. Then, we compute the possible ACDs that fit this frame and build a graph with the candidate ACDs as nodes. By building this graph, we transform the quantization problem into one of path selection, where the nodes represent the ACDs and determine the note values of the transcription and the edges represent tempo transitions between frames. A path through the graph thus corresponds to a rhythm transcription. For path selection, we present both an automated method using weights for evaluating the transcription and finding the shortest path, and an interactive approach that gives users the possibility of influencing the path selection.

1. INTRODUCTION

Many techniques have been proposed for the problem of rhythm transcription and quantization: some build a rhythm tree [1, 2], some use probabilistic models and Monte Carlo pruning [3], and others use signal processing methods like autocorrelation [4]. Here, we present yet another approach to the problem using the notion of *approximate common divisors* (ACDs). This mathematical object, stemming from cryptography [5], is related to the rhythmic notion of tatum, which will be used to transcribe a series of onset times into a musical rhythm. A similar technique has been used to find the greatest common divisor of an inter-onset interval histogram [6]. The difference in our approach is that we consider also common divisors that are not the greatest, and we build a graph such that each candidate common divisor maps to a node in the graph.

This approach was first explored in [7] to automate the transcription of a rhythmic sequence in expressive music performances and arrhythmic heartbeats into musical notation in the context of the ERC project COSMOS (`http://cosmos.ircam.fr`). Here, we describe the original problem and algorithmic approach, which addresses a monophonic rhythm input, then propose an extension that can handle polyphonic input.

The main observation driving the algorithm design is that a given rhythmic input may have several transcription possibilities. We model the options using a graph where each path corresponds to a possible transcription. The choice of the path may be automated but it may also be selected interactively with a human in the loop. We will present both approaches and consider how they might coexist and cooperate within the same framework.

This paper is organised as follows: Section 2 gives an introduction to approximate common divisors, explaining how they can be used to quantize a rhythm by creating a temporal grid. After defining ACDs, we first address the problem of transcribing monophonic rhythms in Section 3, then broaden the description to the transcription of polyphonic rhythms in Section 4. While both cases can be approached using the same technique, they differ in how the input is split. Finally, in Section 5, we propose a graphical user interface that allows a user to select the path and thus influence the transcription result.

2. APPROXIMATE COMMON DIVISORS

The notion of an *approximate common divisor* (ACD) has been studied by Howgrave-Graham in [5], with integral ACDs in the context of cryptography. In this paper, we will allow ACDs to take on real values with a slightly different definition adapted to the purpose of rhythm transcription. Let us then provide an exposition on the ACD finding problem, presenting it in a form suitable for the rhythm transcription context, working our way up from the problem of finding (exact) common divisors.

2.1 The ACD finding problem in $\mathbb{R}$

The common divisor finding problem in $\mathbb{N}$ is that of finding the common divisors of a series of N natural numbers,

$$D = (d_1, d_2, ..., d_N) \in \mathbb{N}^N \,. \tag{1}$$

This problem is equivalent to finding the numbers $a \in \mathbb{N}$ such that $\forall n \in \{1, ..., N\}$, $d_n \in a\mathbb{N} \triangleq \{am \in \mathbb{N} : m \in \mathbb{N}\}$. This problem, which is well posed in $\mathbb{N}$, can be extended to $\mathbb{R}$ with some adaptations.

First of all, we will allow both a and the durations d_n to be positive real numbers, *i.e.*: $a > 0$, $d_n > 0$, $\forall n \in \{1, ..., N\}$. Then, we define the a-grid as

$$a\mathbb{Z} \triangleq \{am \in \mathbb{R} : m \in \mathbb{Z}\} \,. \tag{2}$$

This new framework allows us to extend the problem to a series of timestamps,

$$T = (t_0, t_1, ..., t_N) \in \mathbb{R}^{N+1}, \tag{3}$$

where $t_0 = 0$ and $\forall n \in \{1, ..., N\}, t_n = t_{n-1} + d_n$. Here, we will work with timestamps, but the definitions work for both timestamps and durations.

In practice, the timestamps are expressed in some time unit, typically MIDI ticks or seconds, and we search for numbers $a > 0$ such that t_n fits into the a-grid $\forall n \in \{1, ..., N\}$. In musical terms, a is the duration of a certain note value that functions as the tatum of the rhythmic sequence and the numbers m_n such that

$$t_n - t_{n-1} = d_n = am_n$$

are the multipliers of a to produce the d_n's.

2.2 Definition of approximate common divisors

In practice, unless algorithmically generated, durations are rarely if ever exact multiples of a non-trivial divisor. We thus introduce some flexibility into the common divisor finding problem through a threshold $\tau > 0$.

We now relax the notion of fitting the a-grid by requiring timestamps to be within the threshold τ of the a-grid, *i.e.*:

$$\epsilon(t_n, a\mathbb{Z}) \triangleq \min_{m \in \mathbb{Z}} |t_n - am| \leq \tau, \tag{4}$$

$\forall n \in \{0, ..., N\}$, where $\epsilon(t_n, a\mathbb{Z})$ is the closest distance between the timestamp t_n and the a-grid. Since we require all the timestamps to be within the threshold, τ, of the a-grid, Equation 4 thus implies that the maximum distance between the timestamp series T and the a-grid, $\epsilon_T(a)$, is also within the threshold τ, i.e.,

$$\epsilon_T(a) \triangleq \max_n \epsilon(t_n, a\mathbb{Z}) \leq \tau. \tag{5}$$

Figure 1 shows the fit between a timestamp series $T = (0, 0.98, 1.52)$, given in seconds (s), and the 0.5 s-grid with a threshold of 0.05 s. We will use the threshold $\tau = 0.05$ s for all remaining examples.

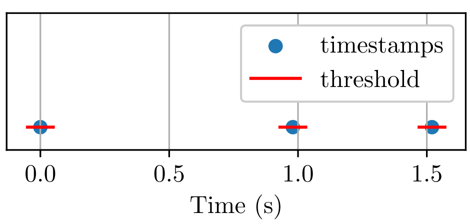


Figure 1. The fit between timestamp series $T = (0, 0.98, 1.52)$ s and a 0.5 s-grid for threshold $\tau = 0.05$ s.

With these definitions in place, we now formally define ACDs for a timestamp series T.

Definition 1 *Let $T = (t_0, t_1, ..., t_N) \in \mathbb{R}^{N+1}$ be a timestamp series. $a > 0$ is an approximate common divisor (ACD) of T with threshold $\tau > 0$ if*

1. *$\epsilon_T(a) \leq \tau$; and,*
2. *ϵ_T has a local minimum at a.*

The first condition ensures that the ACD satisfies the threshold requirement for the timestamp series T. The second condition makes the set of ACDs discrete by selecting the ACD that minimizes the error within each interval $\{a \in (0, \infty) : \epsilon_T(a) \leq \tau\}$.

In practice, since $\frac{a}{2}$ is an upper-bound for $\epsilon_T(a)$, we will see an increasing number of ACDs as a approaches 0, most of which are irrelevant since they are too small. This is why we will choose a lower bound for a, say 0.2 s.

Given a timestamp series $T = (0, 0.98, 1.52)$ s, we plot the function ϵ_T for the range $a \in [0.2, 1.0]$ as shown in Figure 2. A horizontal line marks the threshold $\tau = 0.05$ s and dots the ACDs. We compute $\epsilon_T(a)$ at steps of 1 ms. Since the definition of the ACDs involves local minima, the resolution of the computations can influence the results; in this paper, the precision is set to the millisecond.

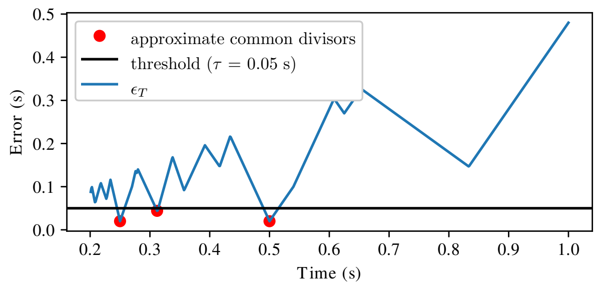


Figure 2. Approximate common divisors of the timestamp series $T = (0, 0.98, 1.52)$ s within the range $a \in [0.2, 1.0]$.

The computational complexity of the ACDs is linear with respect to the length of the timestamp series and the number of candidate values for a. The code uses Python libraries NumPy [8] and SciPy [9] and is available online [1].

2.3 Vectors associated with an ACD

The point of a number a being an ACD of a timestamp series $T \in \mathbb{R}^{N+1}$ with threshold τ is that there is an associated vector

$$M = (M[0], ..., M[N]) \in \mathbb{Z}^{N+1} \tag{6}$$

where $M[n] = \arg\min_{m \in \mathbb{Z}} |t_n - am|$, that consists of the integer values of the grid that fit the timestamps. Setting $\tilde{T} = aM \in \mathbb{R}^{N+1}$, we have

$$||\tilde{T} - T||_\infty = \epsilon \leq \tau, \tag{7}$$

where $\epsilon = \epsilon_T(a)$ will be called the approximation error. We can think of $\tilde{T}$ as the approximated timestamp series, which will never be more than ϵ off from the original.

Moreover, we can deduce the *integer durations*

$$\Delta = (M[n] - M[n-1])_{n=1}^N \in \mathbb{Z}^N \tag{8}$$

[1] `https://github.com/Manza12/TENOR-2022`

that satisfy the inequality

$$||\tilde{D} - D||_\infty \leq 2\tau\,, \tag{9}$$

where $\tilde{D} = a\Delta \in \mathbb{R}^N$ are the approximated durations.

In the example showed in the Figure 2 we get

$$\begin{array}{lll} a_1 = 0.5\,\mathrm{s} & a_2 = 0.312\,\mathrm{s} & a_3 = 0.25\,\mathrm{s} \\ \epsilon_1 = 0.02\,\mathrm{s} & \epsilon_2 = 0.044\,\mathrm{s} & \epsilon_3 = 0.02\,\mathrm{s} \end{array}$$

with a_i being the ACDs and ϵ_i being the corresponding errors, $\epsilon_T(a_i)$. We also retrieve the integer vectors

$$\begin{array}{lll} M_1 = (0,2,3) & M_2 = (0,3,5) & M_3 = (0,4,6) \\ \Delta_1 = (2,1) & \Delta_2 = (3,2) & \Delta_3 = (4,2)\,. \end{array}$$

The next section will show how to use ACDs and the resulting vectors to transcribe monophonic rhythms.

3. MONOPHONIC TRANSCRIPTION

Let us first pose the problem of monophonic transcription in a way adapted to the formalism proposed for ACDs: when we have a monophonic rhythm, for example one produced by an instrument playing one note at a time, we can assume for simplicity that the release of each note occurs at the same time as the onset of the following note.

Moreover, since two onsets cannot occur at the same time, integer durations should not be 0. This means that the ACDs should be small enough to prevent this from occurring, say by requiring that the minimum candidate ACD be smaller than the smallest duration we wish to transcribe.

Let us consider, for instance, the rhythm

♩ ♫ ♪.𝅘𝅥𝅯 ♩

played at a tempo of ♩ = 60. The exact timestamp series and durations will thus be

$$T = (0, 1, 1.5, 2, 2.75, 3, 4)\,\mathrm{s} \tag{10}$$

$$D = (1, 0.5, 0.5, 0.75, 0.25, 1)\,\mathrm{s}\,, \tag{11}$$

where we consider the onsets of all notes and the release of the last note.

If humans were to play this rhythm, they would deviate slightly from these timestamps. Throughout this section, we consider a human realisation of this rhythm given by

$$T = (0, 1.018, 1.531, 2.061, 2.888, 3.179, 4.286)\,\mathrm{s} \tag{12}$$

$$D = (0, 1.018, 0.513, 0.53, 0.827, 0.291, 1.107)\,\mathrm{s}\,. \tag{13}$$

Let us show how one can transcribe this timestamp series into note values using ACDs.

3.1 Frames of a timestamp series

As mentioned, the ACD definition can result in many very small ACDs, which are not very interesting for music transcription because they imply too fine of a time resolution and make the note values too large. We therefore only consider ACDs above a lower bound of $0.2\,\mathrm{s}$ in the example.

Putting a lower bound on ACDs will imply that, if the threshold is small enough, some timestamp series may not have any ACD. In the musical context, this can be thought of as a player that deviates from a metronome, the metronome playing the role of the grid with the ACD being the time interval between two beats[2]. But this can be overcame by splitting the timestamp series into smaller blocks.

Given a timestamp series $T \in \mathbb{R}^{N+1}$ and a frame length $L \in \mathbb{N}$, $\forall n \in \{0, ..., N-L+1\}$, the frame F_n of length L given by the vector

$$F_n = (t_n, t_{n+1}, ..., t_{n+L-1}) \in \mathbb{R}^L\,. \tag{14}$$

We now focus on frames of length 3 and on finding their ACDs. It is important to note that we need to shift our frame so that one of its timestamps is 0 in order to adapt the ACDs to that frame. We will then have

$$F_n = (t_n - t_i, t_{n+1} - t_i, ..., t_{n+L-1} - t_i) \in \mathbb{R}^L\,, \tag{15}$$

where $i \in \{n, ..., n+L-1\}$ is the centering index.

For instance, if we take the first frame of the T defined in (12), $F_0 = (0, 1.018, 1.531)$, we have the ACDs and the integer durations

$$\begin{array}{lll} a_0^0 = 0.51\,\mathrm{s} & a_1^0 = 0.255\,\mathrm{s} & a_2^0 = 0.212\,\mathrm{s} \\ \Delta_0^0 = (2,1) & \Delta_1^0 = (4,2) & \Delta_2^0 = (5,2)\,, \end{array}$$

which give us three ways of transcribing the first two note values that are

$R_0^0 =$ ♩ ♪ $\qquad R_1^0 =$ 𝅗𝅥 ♩ $\qquad R_2^0 =$ 𝅗𝅥‿♪ ♩,

where the unit is ♪ .

3.2 Consistency across frames

Using the approach of frame-wise transcription, we need to impose some consistency across frames. Because of the way we have defined the frames, consecutive frames have two overlapping timestamps so one duration is shared between the two frames. For a coherent transcription, the common duration should be the same.

For instance, if we take the shifted second frame of T, $F_1 = (1.018, 1.531, 2.061) - 1.018 = (0, 0.513, 1.043)$, we have the ACDs and the integer durations

$$\begin{array}{ll} a_0^1 = 0.519\,\mathrm{s} & a_1^1 = 0.259\,\mathrm{s} \\ \Delta_0^1 = (1,1) & \Delta_1^1 = (2,2)\,. \end{array}$$

Since the frames F_0 and F_1 share the second and first duration, respectively, they should satisfy the consistency condition expressed as the equation

$$\Delta_{k_0}^0[1] = \Delta_{k_1}^1[0] \tag{16}$$

for them to be consistent. This equation will be satisfied by certain pairs (k_0, k_1) but not by others. This consistency requirement motivates the construction of the graph that is presented in the next section.

[2] Note that a metronome usually plays the tactus rather than the tatum; here, the metronome analogy should be thought of in the sense of the tatum.

3.3 The ACD graph

For each frame, we model each ACD as a node. Then, we add an edge from an ACD of a frame to an ACD of the next frame if the integer durations are consistent. If we do this for the timestamp sequence from (12), we obtain the graph shown in Figure 3.

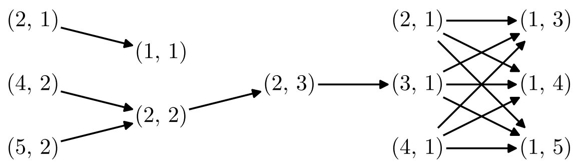


Figure 3. Graph of integer durations Δ.

Figure 3 shows how the consistency condition is fulfilled. It shows also how we can choose a transcription by selecting a path in the graph. Suppose we choose the path

$$(4,2) \to (2,2) \to (2,3) \to (3,1) \to (1,4)\,, \qquad (17)$$

then by merging the common integer durations, we get the rhythm $(4,2,2,3,1,4)$. If we set the unit to be ♪, we will recover the rhythm ♩ ♫ ♫ ♩.

We can get different transcriptions by selecting different paths, which allows transcription to be framed as an interactive task. However, we may be interested in automatic transcription and, in this sense, we may assign weights to the edges in order to have a notion of the *shortest path* and the ensuing notion of the *best transcription*. In the following section we will propose a way of assigning weights to the edges based on the notion of *tempo variation*.

3.4 Assigning weights to the edges

In the first instance, we may choose to assign weights to the edges by weighting them by some function of the error ϵ. This will imply that better transcriptions are the ones that have ACDs that more closely fit the timestamps. Since the error is associated to the nodes instead of the edges, and path finding problems typically have weights assigned to edges rather than nodes, we must decide if the error is associated with the incoming or outgoing edge.

Whereas this is a valid approach, we propose another way of weighting the edges based on tempo variation. To illustrate this concept, we will use Figure 4.

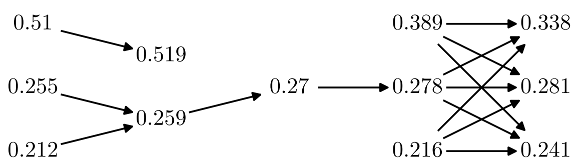


Figure 4. Graph of ACDs.

When we go from a node to another via an edge, we are changing the duration of the ACD and in so doing, we are varying the tempo. For example, if we go from $0.255\,\mathrm{s}$ to $0.259\,\mathrm{s}$ we are decreasing the tempo from ♪ = 235 to ♪ = 232, thus reducing the speed by $2\,\%$. This is not a big change, whereas if we consider instead the transition from $0.389\,\mathrm{s}$ to $0.241\,\mathrm{s}$, the tempo increment is $61\,\%$.

The tempo variation between two ACDs, measured in percent, may be defined as

$$\delta(a_1, a_2) = 100\left(\frac{a_1}{a_2} - 1\right) \qquad (18)$$

and we say that the speed from a_1 to a_2 has increased by $\delta(a_1,a_2)\%$ if $\delta(a_1,a_2) > 0$ and decreased by $\delta(a_1,a_2)\%$ if $\delta(a_1,a_2) < 0$. Notice that we divided a_1 by a_2 because we consider speed rather than duration; the two are inversely proportional.

We can then set the weight associated with an edge as a function of $\delta(a_1,a_2)$. In this case, we choose the logarithmic distance between a_1 and a_2, which is defined by

$$d_{\log}(a_1,a_2) \triangleq \left|\log_2\left(\frac{a_1}{a_2}\right)\right| = \left|\log_2\left(\frac{a_2}{a_1}\right)\right|. \qquad (19)$$

This has the property that it is symmetric and returns a value of one for a ratio of 2 : 1.

Figure 5 shows the ACD graph of T weighted by the logarithmic distance.

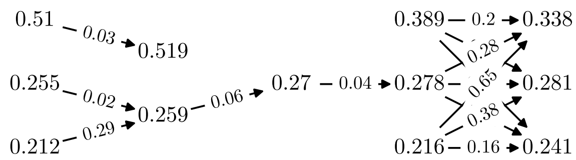


Figure 5. Weighted ACD graph

3.5 Shortest path problem

As we have a weighted graph, we can consider the shortest path problem and its corresponding transcription, the best one according to the parameters. The shortest path problem is defined by giving two nodes of the graph and finding the path that links them and has minimal weight. However, in our case, we may have multiple nodes for the first and last frames which makes ambiguous which nodes to choose. This is easily solved by adding an artificial source at the beginning that connects to all the ACDs of the first frame and an artificial sink connected to all the ACDs of the final frame. We may associate the weight 0 to these edges and then we will have a well posed shortest path problem. This is illustrated in Figure 6.

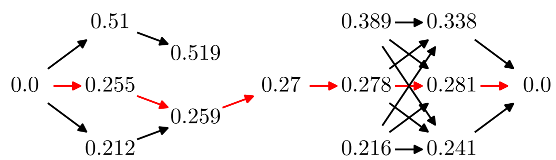


Figure 6. Shortest path outlined in red.

Since the graph is directed and acyclic, the shortest path can be readily computed; indeed, using the topological sorting of a directed acyclic graph, we have an algorithm in linear time [10]. We can see that we recover the path from (17) that gives the rhythm ♩ ♫ ♫ ♩.

As pointed out, the shortest path depends directly on the weights, and their selection highly influences the outcome. We have presented weights based on the tempo variation or on the error ϵ, but there are many other approaches and combinations thereof, like selecting weights to avoid complex durations that are inelegant to write down, like 5, 11, 13... Conversely, if we value precision above notation clarity or physical reproducibility, we can assign a large weight to the error and allow very small ACDs. The trade-off between precision and clarity then arises as a parametrization problem that could be tuned via statistics or machine learning given labeled data. The subject of human intervention will be covered in Section 5.

3.6 The influence of the frame length

Up to this point, we have been working with frames of length 3. This approach makes sense because we always have a common duration that constraints the paths to be consistent from one frame to the next. However, we may choose a different frame size, for instance if we have very fast rhythms. In this case, consistency will be required of all the integer durations that are common to consecutive frames.

If we have two consecutive frames F_n and F_{n+1} of length $L \in \mathbb{N}$, the condition for the integer durations will be

$$\Delta_n[1 : L - 1] = \Delta_{n+1}[0 : L - 2]\,. \qquad (20)$$

Here, indexing follows the Python convention where the second index is excluded.

Longer frames enforce the restrictions on ACDs and reduce their numbers. For instance, if we set the frame length to $L = 4$ for our previous example, we will have the graph shown in Figure 7, where the virtual initial and final integer durations are set to the null list.

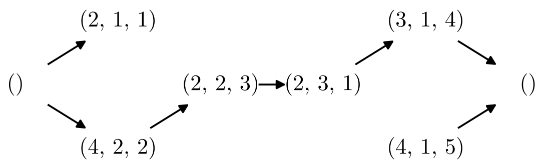


Figure 7. Graph of T with frame length $L = 4$.

We see in this case that there is a single valid path that corresponds to the integer durations we had in (17).

4. POLYPHONIC TRANSCRIPTION

Up to now, we have only considered monophonic rhythms, *i.e.*, rhythms formed by no more than one note at a time. This is useful in the case of singing voice, string instruments (in some cases), winds, etc. but it is insufficient in general and, for instance, for the piano.

When there are several voices playing at the same time, we must adapt our method so as to have meaningful results. First, we can no longer conceive of a frame as being a subvector of the time series T of fixed size L; indeed, we may have L timestamps occurring at almost the same time and then the integer durations would all be zero.

This leads us to an even deeper question: which timestamps should we consider, only onsets or also offsets? This is a very delicate question in music writing since it addresses directly the question of rests and articulation; indeed, since we no longer consider that the offset occurs at the subsequent onset, we have not only to quantize the onset but also the offset.

In order to simplify things, and acknowledging that our approach to the problem will be incomplete, we will focus on transcribing only the onsets. Then, we can account for the offsets in some sense, for instance rounding them toward the closest element in the grid. We will not tackle this problem, leaving it to future work, since it is affected by numerous factors such as performance, pedal, clean transcription and articulation.

That being said, let us consider the excerpt showed in Figure 8 from a Mozart sonata.

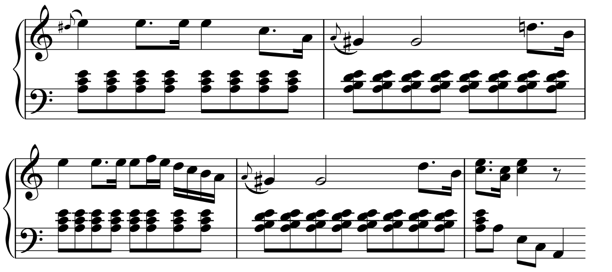


Figure 8. Start of the Mozart's Piano Sonata No. 8 in A minor, K. 310 / 300d.

We shall next attempt to transcribe a human performance of this excerpt, shown as a piano roll in Figure 9, by adapting the technique developed in the previous section.

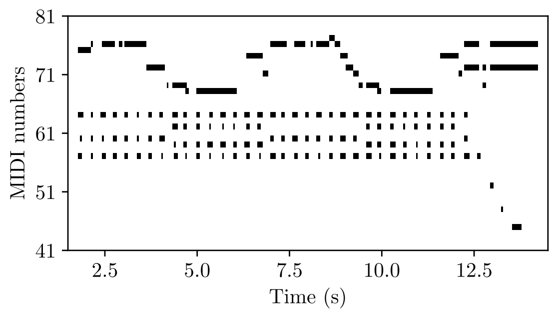


Figure 9. Piano roll of a human performance of the excerpt from Figure 8.

4.1 Polyphonic frames

For polyphonic transcription, we must first define a notion of a frame that will be adapted for polyphonic rhythm transcription. Consider, as previously, a timestamp series $T = (t_0, t_1, ..., t_N) \in \mathbb{R}^N$. As several timestamps can now be separated by a few milliseconds, like in the case of chords, we will set the duration of the frame to $L > 0$ in seconds. Then, we will define the frame F_t starting at $t \in \mathbb{R}$ of length L as

$$F_t = (t_n \in T : t \leq t_n < t + L) \,. \tag{21}$$

In this way, each frame contains the timestamps of T that are within the time interval $[t, t+L)$. The duration L may be tailored to each piece, or even section or segment, and may be pre-computed. It may even be time-varying according to the speed of the piece.

In order to produce a consistent transcription, we shall overlap frames such that they measure common durations. This will be done by selecting overlapping frames with a hop size of $H > 0$. In our case, we choose

$$L = 1.5\,\mathrm{s} \qquad\qquad H = \frac{L}{2} = 0.75\,\mathrm{s}\,.$$

In this way, we arrive at a family of frames

$$\{F_{mH} \subseteq T : m \in \mathbb{N}\}\,. \tag{22}$$

Even though the family is formally infinite, as time progresses, the frames will eventually be empty when the input ends, so we can consider only the subfamily of non-empty frames.

4.2 Transcribing frames

Now that we have frames, we can apply the same procedure of extracting the ACDs of the frame as in the monophonic case. By default, we will shift the frame by its first timestamp, that is $F' = F - F[0]$. However, as mentioned before, we can shift the frame on any of its timestamps, for instance the middle one, leading to potentially different results. This approach may give negative numbers in the vector M, but is completely valid from the perspective of the ACD computation, which is defined on $\mathbb{Z}$, and from the transcription paradigm.

As previously, we will recover the vectors $M \in \mathbb{Z}^{N_t}$ and $\Delta \in \mathbb{Z}^{N_t - 1}$ whose size vary for different $t = mH$. It is important to emphasise that now there may be several durations in Δ that are 0, which points to the fact that some timestamps are concurrent. We should abandon the concept of Δ being the duration of the notes since we are only considering onsets; rather, we will think of Δ as a consistency vector that will be used to check if two overlapping frames are consistent.

Figure 10 shows MIDI onsets fitting into a $0.218\,\mathrm{s}$-grid shifted by the first timestamp of the frame.

4.3 The ACD graph

As before, we will recover the ACDs of each frame to build a graph with ACDs as nodes, linking the nodes if they are consistent. Depending on the hop size, we may have more than two overlapping frames, but then we will only connect ACDs between consecutive frames.

The consistency condition between frames will now involve timestamps that are common to consecutive frames.

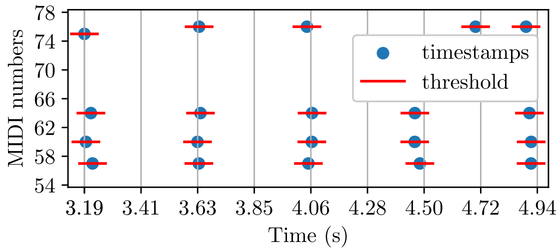


Figure 10. Timestamps of MIDI onsets fitting into a $0.218\,\mathrm{s}$-grid.

For two frames F_{mH} and $F_{(m+1)H}$, the consistency condition will be:

$$\Delta_{mH}[n_i - n_m] = \Delta_{(m+1)H}[n_i - n_{m+1}] \tag{23}$$

$\forall n_i \in \mathbb{N}$ such that $t_{n_i}, t_{n_i+1} \in F_{mH} \cap F_{(m+1)H}$, where $n_m \in \mathbb{N}$ and $n_{m+1} \in \mathbb{N}$ are the first index belonging to F_{mH} and $F_{(m+1)H}$ respectively.

Now that we have established the consistency condition, we can plot the ACD graph of the onsets of the excerpt presented in Figure 9. We added a source node and a sink node to complete the path and, as seen in the Figure 11, there is only one possible path linking them.

Of course, this will not always be the case. The single path solution was a consequence of the fact that the player followed the rhythm very strictly, in part due to the genre of the music. If we take the vector M built by following this path, we will recover the onsets expected by the score.

We will not repeat all the considerations outlined in the previous section regarding path weighting and automating of the transcription model. Rather, in the next section, we will show how transcribing can be done interactively.

5. INTERACTIVE PATH SELECTION

We have modeled both monophonic and polyphonic transcription as path selection problems. Solving either of these problems can be done automatically if we assign weights to the path, but it can be interesting to select the path interactively by means of a user interface. This transforms the rhythm transcription problem into a multiple output problem, and gives humans the responsibility of selecting their preferred transcription from among a set of possibilities.

Figure 12 shows a prospective user interface for selecting the path. This interface has not been implemented; it presents only a possible graphical layout of the essential functions. It is intended, however, to be implemented in a future version of OpenMusic [11, 12].

Let us go through an overview of the interface from top to bottom, then left to right.

5.1 The piano roll panel

In this panel, the user may see the actual MIDI file and navigate the frames with the arrows. S/he may change the size of the frame and its position by dragging and dropping

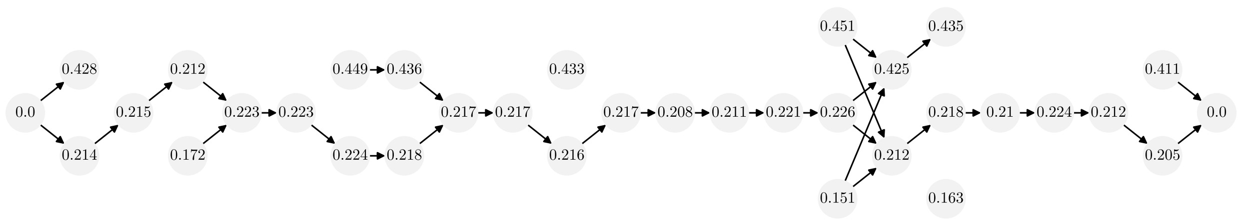


Figure 11. Graph of the excerpt from the onsets of Figure 9.

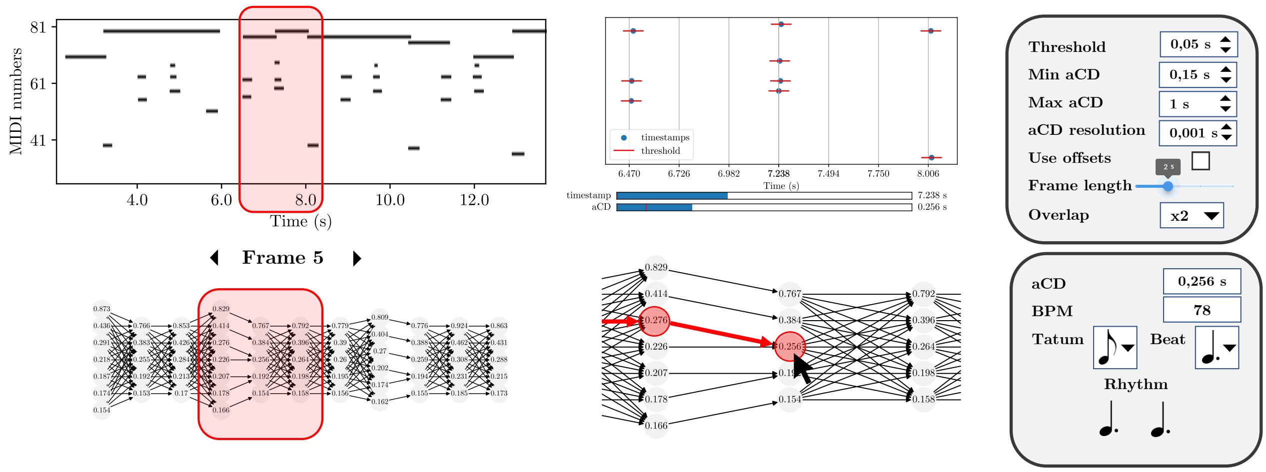


Figure 12. Design of a prospective graphic user interface for interactive path selection.

the limits of the frame. Ideally, the user should also be able to edit the MIDI file by selecting the notes and stretching or compressing them; this way, s/he could influence the transcription by editing the timestamps.

5.2 The full graph panel

This panel shows the full ACD graph and will be related to the frame selected on the piano roll panel. Its purpose is to keep track of the path selected and to allow the user to navigate through frames so that s/he may have a global view.

5.3 The grid panel

This panel presents information from the part of the piano roll where the frame is located. However, instead of MIDI bars, the onsets (and potentially the offsets) will be shown as timestamps so that we may know exactly where each fits in the grid. Timestamps about to be transcribed are plotted in blue and the threshold shown in red. The representation helps the user understanding the scale and the degree of freedom. With the two sliders in the bottom, we can tune which timestamp will be the shift parameter and the resulting grid if we change the ACD.

5.4 The focused graph

This sub-graph of the full ACD graph serves shows the ACD selected for each frame. With the cursor, we may select an ACD. The red line connects the current to the previous ACD. Options in the previous and next frames are shown to convey the available paths when choosing an ACD.

This will be one of the main panels on which to act. Indeed, changing the ACD will trigger updates on other panels, and it will be by selecting the ACDs one by one that the transcription will be made. Nevertheless, we can initialise the graph with some optimal or near optimal path, but the interactive part of the process will be to adapt the graph by changing the ACD.

5.5 The parameters panel

The parameters panel will let the user select all the parameters needed for the ACDs computation:

1. In order to select the timestamps, the user should choose the frame duration with a slider. Also, s/he can select the overlap via a drop-down list. We may also use a checkbox to determine if the offsets are used in the computations. If that is the case, they will also appear in the grid panel.

2. Once the timestamps are selected, the parameters for the computation of the ACD shall be tuned with spinners; these parameters are

   (a) the error threshold,

   (b) the lower bound,

   (c) the upper bound, and

(d) the computation resolution.

These parameters will play an important role in which ACDs are recovered.

5.6 The rhythm panel

In this panel, users can see the outcome of their choice; by selecting an ACD in the focused graph, both the ACD and the BPM (beats per minute) will be printed in non editable text boxes. The ACD corresponds to the tatum, which can be selected in a drop-down list. The beat, that will usually consist of several tatums, can also be freely chosen in another drop-down list.

In addition, when the ACD is selected, the timestamps will be transcribed into rhythms via a label in the panel. It should be noted that this rhythm corresponds to onsets and should be thought of as a musical grid rather than the durations of the notes, at least when offsets are not taken into account.

6. CONCLUSIONS AND FUTURE WORK

In this work, we have shown how an extension of the notion of common divisors to a continuous framework leads valuable contributions to rhythm transcription. Their linear computational complexity makes them a suitable and efficient tool for quantizing musical rhythms for large amounts of data.

We have also proposed a flexible framework where we considered transcription by splitting the timestamp series into frames and computing ACDs separately. However, when this is done, adjacent frames need to be consistent, and we have chosen to model this via a graph in which ACDs are nodes and edges represent the consistency between frames.

Once this graph is set up, the transcription problem then turns into a one of path selection. Using this paradigm, we proposed two complementary ways of solving the transcription problem: by assigning weights and determining the shortest path, or by allowing humans to intervene in the process by selecting a path that results in a more desirable transcription.

Regarding this last option, we presented a prospective graphical user interface and gave an exposition of its likely elements. This interface will allow the user to directly steer the transcription and may be implemented in musically-oriented frameworks.

In conclusion, we have proposed a framework that is simple in essence but highly parameterizable. Indeed, there are many ways in which we may affect either the ACD computation or the graph creation and weighting, for example by tuning the weights, limiting the tempo variation or by allowing only certain integer durations.

A remaining challenge is the organisation of the durations into groups and measures. By considering the tatum instead of the tactus, we may enter into a low-level quantization that does not account for how the durations are grouped together. The grouping of durations to form rhythm trees has been studied by [2] and is of major importance in OpenMusic [12].

A way to tackle this may be to build a rhythm tree on top of the ACD graph. How this may be done is outside the scope of this paper but several possibilities can be explored, like choosing different duration groupings and evaluating if they would form a tactus. A challenge for both these approaches is the handling of ties, which could also be broken through the user interface.

In the future, the tool we proposed could be incorporated into existing frameworks like OpenMusic either as a function that performs transcription automatically or as a user interface that allows composers and music editors to interact with the transcription. We would then be able to adapt criteria for weighting the graph to tune the standalone part of the algorithm. To that end, we may use several techniques, but one interesting option would be to record data from musicians transcribing rhythms in order to tap into their experience so as to make the output of the algorithm as human friendly and readable as possible.

Acknowledgments

This result is part of the COSMOS project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 788960.).

7. REFERENCES

[1] C. Agón, G. Assayag, J. Fineberg, and C. Rueda, "Kant: a Critique of Pure Quantification," in *In Proceedings of the International Computer Music Conference (ICMC)*, Aarhus, Denmark, 1994, pp. 52–59.

[2] A. Ycart, F. Jacquemard, J. Bresson, and S. Staworko, "A Supervised Approach for Rhythm Transcription Based on Tree Series Enumeration," in *In Proceedings of the International Computer Music Conference (ICMC)*, Utrecht, Netherlands, 2016, pp. 369–376.

[3] A. T. Cemgil and B. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," *Journal of Artificial Intelligence Research*, vol. 18, no. 1, pp. 45–81, 2003.

[4] J. C. Brown, "Determination of the meter of musical scores by autocorrelation," *The Journal of the Acoustical Society of America*, vol. 94, no. 4, pp. 1953–1957, 1993.

[5] N. Howgrave-Graham, "Approximate Integer Common Divisors," in *Cryptography and Lattices*, ser. Lecture Notes in Computer Science, J. H. Silverman, Ed. Berlin, Heidelberg: Springer, 2001, pp. 51–66.

[6] J. Seppänen, "Computational Models of Musical Meter Recognition," Master's thesis, Tampere University, Tampere, 2001.

[7] G. Romero-García, "Rhythm Transcription and Characterization for Performed Music and Arrhythmia Sequences," Master's thesis, Sorbonne Université, Paris, 2020.

[8] C. R. Harris, K. J. Millman, S. J. van der Walt, and others, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[9] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, and J. Bright, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Section 22.4: Topological sort,” in *Introduction to Algorithms*, 2nd ed. Cambridge, Mass: MIT Press, 2001, pp. 549–552.

[11] C. Agón, “OpenMusic : un langage visuel pour la composition musicale assistée par ordinateur,” PhD Thesis, Paris 6, Paris, 1998.

[12] J. Bresson, C. Agon, and G. Assayag, “OpenMusic: visual programming environment for music composition, analysis and research,” in *Proceedings of the 19th ACM international conference on Multimedia (MM ’11)*, New York, 2011, pp. 743–746.

NOTATION, TRANSMISSION, AND COMPROVISATION: A CASE STUDY OF THE ONCEIM IMPROVISATION ORCHESTRA

Diemo Schwarz, Clément Canonne[1]
STMS - Ircam CNRS Sorbonne Université
Paris, France
{schwarz, canonne}@ircam.fr

ABSTRACT

ONCEIM is a collective free improvisation ensemble consisting of 30 musicians. Beyond free improvisation, the orchestra also performs new comprovisation works, commissioned from a variety of composers such as Eliane Radigue, Stephen O'Malley, John Tilbury, or Jean-Luc Guionnet. In this paper, we present a case study based on the 23 pieces commissioned by ONCEIM over a period of ten years, from 2012 to 2022. We first give an account of the different approaches encountered, illustrated by some examples of pieces. We then show how ONCEIM's musicians use re-notation strategies in the process of rehearsing such comprovisation pieces. Finally, we reflect on the role and use of electronics, as imagined by the composers with whom ONCEIM has collaborated, within a mostly acoustic setting.

1. INTRODUCTION

This article presents a case study of the use of notation in comprovisation [1, 2] pieces written by 24 composers over more than 10 years for the large musical ensemble *Orchestre des Nouvelles Créations, Expérimentations, et Improvisations Musicales* (ONCEIM).[2] The Paris-based ensemble was created in 2011 by the pianist Frédéric Blondy and is made up of some thirty instrumentalists with various musical backgrounds (jazz, improvised music, classical and contemporary music, noise music, music informatics, etc.). Its artistic activities are based both on large-ensemble "free improvisation" and on commissions from various composers —works that typically combine a written score with improvised material, and that, as such, raise the typical issues associated with comprovisational creative processes [3, 4]. In both cases, ONCEIM performances are characterized by a high level of indeterminacy, which calls for modes of organizing and working that are unusual for large instrumental ensembles (i.e. a relatively fluid distribution of roles, an emphasis on a "work-in-progress" approach, etc.), resulting in a tension that structures all of ONCEIM's activities [5]. Such activities have been the subject of musicological studies [5, 6, 7] dealing in particular with the question of coordination and aesthetic negotiations in this kind of large, leader-less, improvisation orchestra.

ONCEIM has so far created 23 pieces by 24 composers, aiming at exploring a large spectrum of compositional practices —and thus acting as a sort of artistic laboratory for new ways of creating and composing music. The complete list of pieces is given in table 1. Note that the last line of table 1 stands for the 20 free improvisations ONCEIM has performed in concert so far, which are deliberately considered to be a series of pieces called *Laminaire*, collectively created by ONCEIM's musicians themselves, without leader nor any predetermined structure. A certain number of these are available on video[3] and on a forthcoming double CD. For the comprovisation pieces, ONCEIM primarily chooses to commission composers whose aesthetics is broadly compatible with the sonic framework of electroacoustic improvisation (EAI), and, as such, who are able to benefit from the diverse backgrounds of its musicians, and the propositions they can bring forward in terms of sonic material and extended techniques. Another important point is that the composers are expected to be willing to enter a preparatory phase of exchange with the musicians, in order to discover their specificities and elaborate a shared musical ground. Such a participatory approach can open up the possibility of commissioning works from composers from non-academic backgrounds such as improvised music, metal, noise, turntablism, electronic music, by removing the barrier of having to use conventional score notation as the privileged way to communicate with ONCEIM's musicians.

In the remainder of this article, we will first give an account of the different approaches encountered (section 2), illustrated by some examples of pieces, discussing the various ways in which composers chose to transmit their intention to the orchestra and the extent to which they allowed for indeterminacy and drew upon the sonic and musical specificities of the up to 30 improvising musicians of ONCEIM. We will then show how the ensemble's musicians used re-notation strategies in the process of rehearsing such comprovisation pieces (section 3). Finally, we will reflect on the role and use of electronics, as imagined by the composers (section 4).

[1] Equal contribution
[2] http://onceim.fr
[3] https://onceim.fr/media/ and https://www.youtube.com/results?search_query=onceim+laminaire

Composer	Year	Title	Release
Blondy, Frédéric	2012	*Reflets de sillons*	
Beliah, Sébastien	2012	*Garden of Sounds*	
Mariage, Jean-Sébastien	2013	*La haine de la musique*	
Riviere, Arnaud	2014	*Encore*	
O'Malley, Stephen	2014	*Gruidés*	Vinyl [8]
Denzler, Bertrand	2014	*Morph*	CD [9]
Badaroux, Pierre Antoine	2015	*Composition No. 19*	
Radigue, Eliane	2015	*Occam Océan XXV*	CD [10]
Noetinger, Jérôme	2016	*Les machines orphelines*	
Normand, Eric	2016	*Jeu de cartes pour orchestre d'improvisateurs*	
Tilbury, John	2017	*Sans*	
Walker, Deborah	2017	*Gonesse extension*	
Galiay, Frédéric	2018	*Time Elleipsis*	
Ablinger, Peter	2018	*Notes & Bloc-Notes*	digital album [11]
Beins, Burkhard	2018	*Ambush*	
Bosshard, Patricia	2018	*Sillons*	CD [12]
Duboc, Benjamin	2019	*Volumes*	
Naegelen, Karl	2019	*Concerto*	
Tetreault, Martin	2019	*Octaves*	
Charles, Xavier	2019	*Court–Fauve–Circuit*	
Laubeuf, Vincent	2021	*L'appel de l'océan*	
Guionnet, Jean-Luc	2021	*Tournures cessent orchestrales*	
Liu, Germaine & Sorbara, Joe	2022	*Quarantine Playground*	
ONCEIM	ongoing	*Laminaire* collective improvisations	forthcoming CD

Table 1. List of works composed for ONCEIM in chronological order, with the last line standing for the ongoing series of pieces collectively improvised by ONCEIM itself.

2. A CARTOGRAPHY OF ONCEIM'S COMPROVISATION PIECES

When reviewing the corpus of 23 pieces composed so far for ONCEIM, we can identify five principal, highly independent dimensions along which the pieces vary. These feature dimensions are:

Transmission: The mode of transmission of the piece to the orchestra can mainly use a material support of notation (a *score*), be *oral*, or use *audio* examples. Of course, and contrary to what can happen in more traditional settings —in which the score is usually seen as standing in for the composer, ideally encoding the composer's musical intentions so clearly that performers do not require any further interaction with him or her [13]— there is always an oral component in the process of explanation and elaboration of these pieces. But here, we are interested in where the bulk of defining information lies.

Interestingly, material transmission based on a notated score is still the most common practice, playing a crucial role in 19 of the pieces performed by ONCEIM. The most extreme example of oral transmission is Radigue's 2015 piece for ONCEIM which is elaborated and transmitted purely orally — a procedure that is typical of all Radigue's work for instrumentalists. Tilbury's 2017 piece also importantly relied on oral transmission, with the second part of the piece consisting of a few broad verbal instructions provided by the composer during the rehearsals [5]. An example of audio transmission is given by Noetinger 2016, where the composer uses a text-based score giving the sections of activity for each musician (figure 10), but the material within each section is here prescribed by electro-acoustic audio loops produced by the composer, which the musicians have to emulate (or at least reinterpret) using purely instrumental means. Audio transmission was also involved to some extent in Guionnet 2021, in which the composer chose to transcribe 13 of 15 sections of an earlier electro-acoustic tape piece into a graphic score for the orchestra (figure 2). However, the original sound elements were here presented as verbal descriptions within the score, with all the decisions on how to "orchestrate" this for the available instruments having already been made beforehand by the composer, together with ONCEIM's artistic director. Only for the last two sections the musicians heard and had to acoustically render the original sound excerpts.

Notation: In score-based transmission, we can identify three main types of notation *textual, graphic, staff*, and their combinations. (See Blondy's 2012 piece in figure 1 for an example of combined text and graphic notation.)

Staff-based notation is in fact often used by the composers with whom ONCEIM has collaborated, ei-

Composer	Year	Figure	Transmission	Notation	Construction	Impl.	Det.	Role of Electronics
Blondy	2012	(1)	score	text, graphics	timeline	high	high	record+transform
Beliah	2012		score	text	deontic	mid	mid	
Mariage	2013		score	text, graphics	timeline	mid	mid	menace
Riviere	2014		score, oral	graphics	timeline	high	high	
O'Malley	2014	(3)	score	graphics	timeline	mid	high	
Denzler	2014		score	text	deontic	mid	low	
Badaroux	2015		score	text	deontic	mid	mid	remanence
Radigue	2015	(12)	oral	none	subjective	high	mid	no electronics
Noetinger	2016	(8)–(11)	audio, score	text, audio	timeline	low	high	record+transform
Normand	2016	(6)	score	text	deontic	low	low	
Tilbury	2017		oral, score	text, graphics	subjective	high	low	
Walker	2017		score	graphics	subjective	mid	high	ambience
Galiay	2018	(7)	score	graphics	timeline	mid	high	
Ablinger	2018		score	staff	timeline	low	high	
Beins	2018		score	text, graphics	timeline	low	high	
Bosshard	2018	(5)	score	staff	subjective	high	high	
Duboc	2019		score	graphics	timeline	high	high	
Naegelen	2019	(13)	score	staff	timeline	mid	high	record+reduce
Tetreault	2019		score	staff	timeline	low	high	
Charles	2019		score	text, graphics	timeline	mid	mid	
Laubeuf	2021	(4)	score	graphics	timeline	low	high	
Guionnet	2021	(2)	score, audio	graphics	subjective	mid	high	record+transform
Liu & Sorbara	2022		score	text, staff	deontic	low	mid	
ONCEIM	ongoing		audio	none	subjective	high	low	

Table 2. Classification of the works composed for and by ONCEIM according to the 5 feature dimensions (defined in section 2) and role of electronics (see section 4). The last line stands again for the ongoing series of collective improvisations.

ther at a local level (to prescribe selected snippets or cells (Bosshard 2018, figure 5), or, more rarely, at the level of the whole piece, akin to what would be observed in a classical score (Naegelen 2019, figure 13). As for graphic notation, it can be used synoptically, to describe the temporal organisation and evolution of musical parameters (see for example Blondy 2012, figure 1, Guionnet 2021, figure 2, O'Malley 2014, figure 3), or iconically, to suggest textures and materials (Laubeuf 2021, figure 4). Finally, while textual notation is present to some extent in most of the pieces (providing written explanations, rules, or sound descriptions), only Tilbury 2017 takes the form of a purely textual score.

Construction: The type of construction encountered in ONCEIM pieces is either *time-based* or *deontic* (based on rules). The time-based pieces can be either based on a fixed *timeline*, referring to a timer clock visible to all musicians (Blondy 2012, figure 1, O'Malley 2014, figure 3, Laubeuf 2021 figure 4), or on *subjective time*, where either the orchestra musicians or the artistic director (thus de facto acting as a conductor) can decide when it is time to move to a new cue or section (Guionnet 2021, figure 2, Bosshard 2018, figure 5). Note also that the short-scale temporal organisation within the larger sections can rely on *metric time* based on standard measures and tempo indications (Bosshard 2018, figure 5).

The deontic pieces are constructed around a set of rules and an approximate duration. The rules are usually given textually, but can contain snippets of notation, or be embodied in game props (e.g. playing cards in Normand 2016, see figure 6).

Musicians' involvement: ONCEIM musicians are more or less involved in the compositional process of such comprovisation pieces. In particular, the source of the musical material can either be *endogenous* (coming from inside the orchestra) or *exogenous* (provided by the composer). Notable extremes include, on the one hand, Radigue 2015 and Bosshard 2018 (figure 5), where all the musical material was proposed by the orchestra, and subsequently worked into a composition, and, on the other hand, Tetreault 2019 and Ablinger 2017, where the musical material was entirely determined by the composer prior to the rehearsing sessions —a process more akin to what can be observed in classical or contemporary music ensembles. In most cases, however, the tension that might arise between the elicitation of endogenous material and the composer's own wishes is usually resolved through a participatory composition process, as in Tilbury 2017 [5].

Determination: The degree of (in)determination of a comprovisation expresses the amount of choices the musicians have to make when rehearsing or performing the piece, from fully composed pieces such as Tetreault 2019 or Ablinger 2017, to the obviously more open deontic pieces. A highly common solution, here, is to aim for some sort of fertile middle ground, with the choice of acoustic and musical material (timbre, texture, notes, chords, pulsation) being either completely free, or broadly constrained by a more or less precise range of options (as in O'Malley 2014, see figure 3), and the overall temporal and dynamic structure being more fully determined.

The 23 works created by ONCEIM so far are ranked on the five feature dimensions introduced above in table 2.

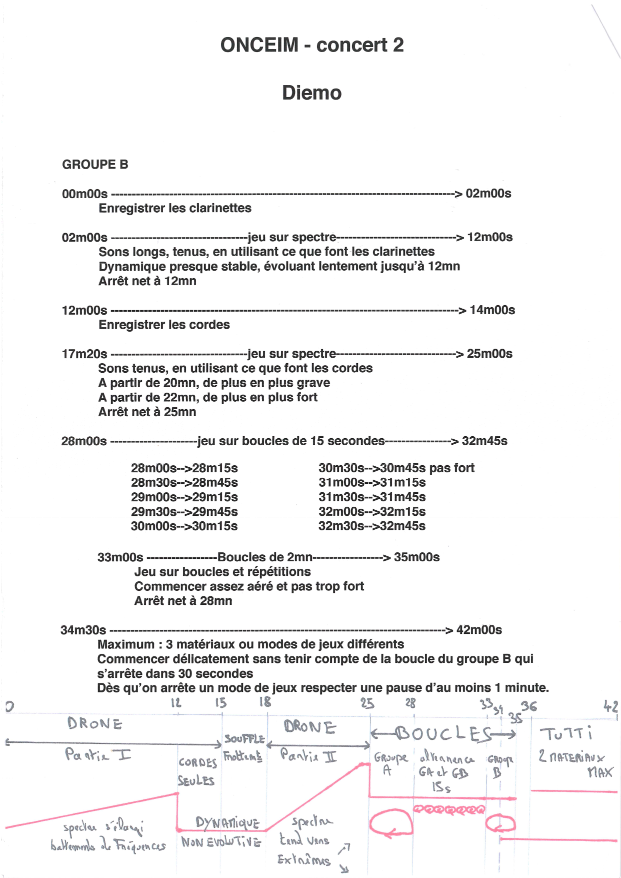


Figure 1. Example of a mix of textual and graphic notations in Blondy 2012. The lower graphic part is the conductor score giving the overall structure and dynamics of the piece. The textual part above is the score for electronics 1, giving instructions for when to record certain instruments of the orchestra (e.g. 0–2min *"Enregistrer les clarinettes"*), and instructions for when and how to use such recordings.

3. NOTATION BEYOND THE WORK

A striking aspect of ONCEIM's comprovisation output is that notation extends well beyond the works and scores transmitted by the composers.

Similarly to what can be found in most score-based musical practices, scores are often annotated by ONCEIM's

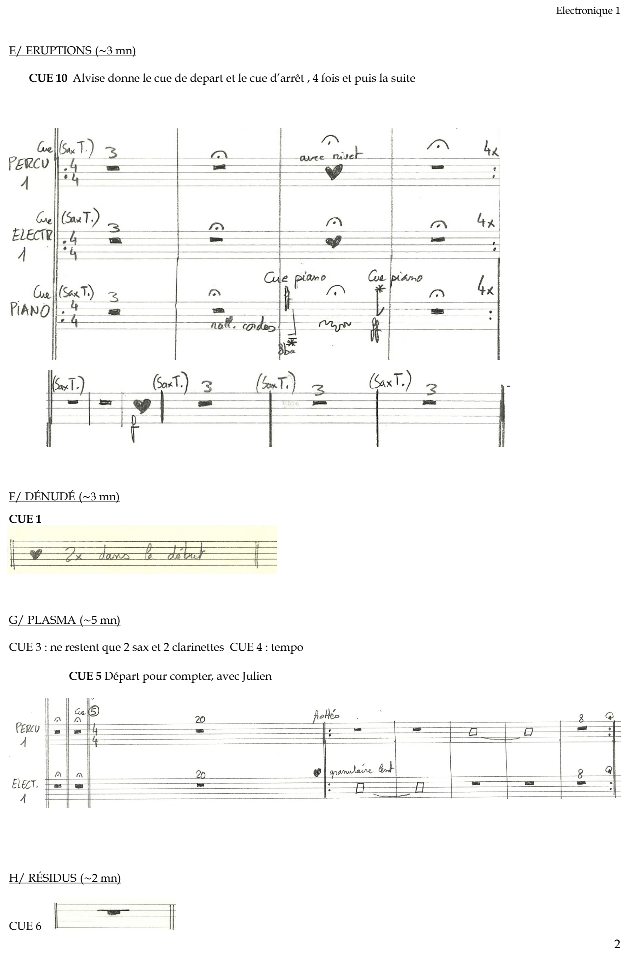


Figure 5. Snippets of staff notation in Bosshard 2018, showing the arrangement of the "preferred phrase" of each musician, symbolised by a heart. The overall construction is based on cues given by a conductor (subjective time) with metric timing in the individual sections.

musicians [14]. Beyond the usual indications of clarifications, fingerings, and dynamic markings, one can also find markings that are more specifically tied to the indeterminate nature of the works performed. For example, in O'Malley 2014, which allows at various points the performers to spontaneously choose from a given set of pitches, most musicians in fact chose to explicitly notate in their scores the precise pitch they were going to play — first as a way to avoid what musicians would perceive as infelicitous pitch arrangements within a given instrumental group, and, second, to allow the musicians to focus on fine-grained acoustic coordination. In that perspective, it should be noted that, according to Frédéric Blondy in the preliminary interview conducted by the second author for the present research, the general tendency when rehearsing comprovisation works was to go towards more and more determination. This is of course due to the often-limited rehearsing time which, when combined with the intrinsic difficulty of coordinating actions and decisions within such a large ensemble, certainly pushes musicians to determine the various aspects of the piece early in the process. For example, an early version of Guionnet 2021's piece was

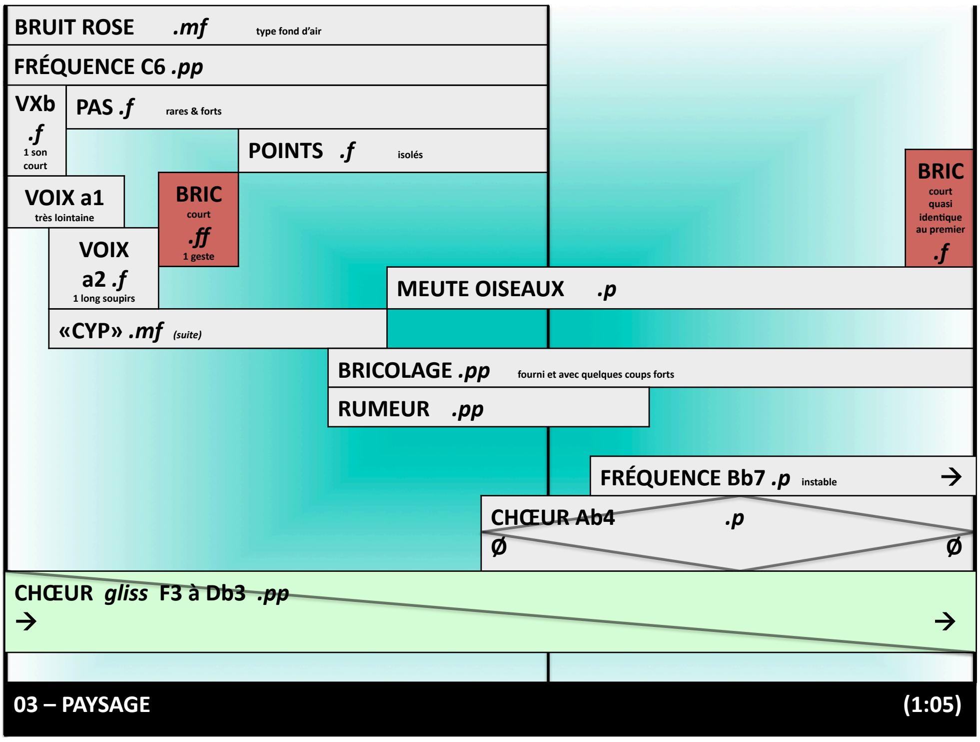


Figure 2. Page 3 from Guionnet 2021. Each rectangle transcribes an element of the pre-existing electro-acoustic piece, determined by its material (described in the instructions of the piece), dynamics and its temporal evolution. Each block is to be played by one or more pre-determined musicians. Red blocks are synchronization points, grey blocks are started with subjective timing.

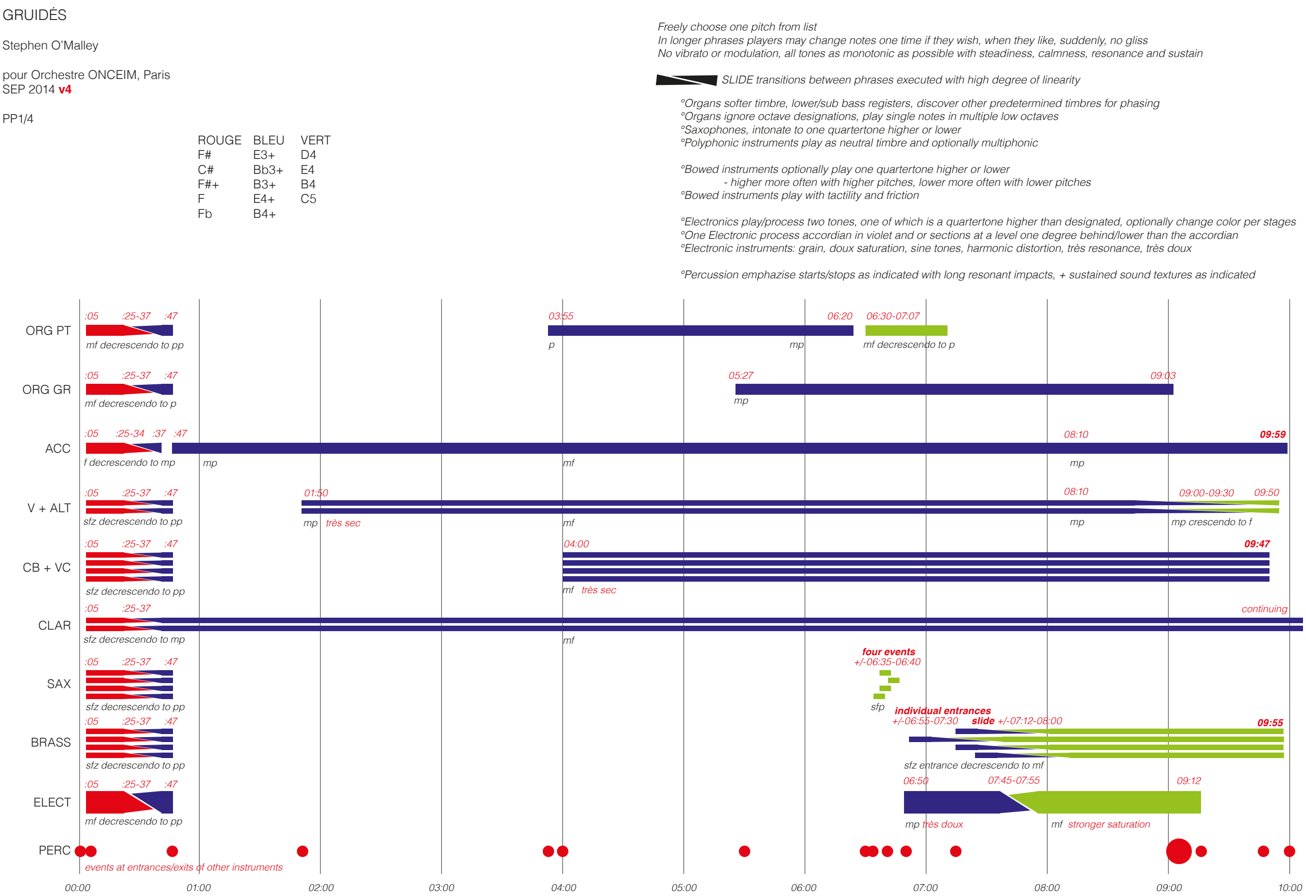


Figure 3. First page of O'Malley 2014. Each colour stands for a list of pitches from which each individual musician can freely choose.

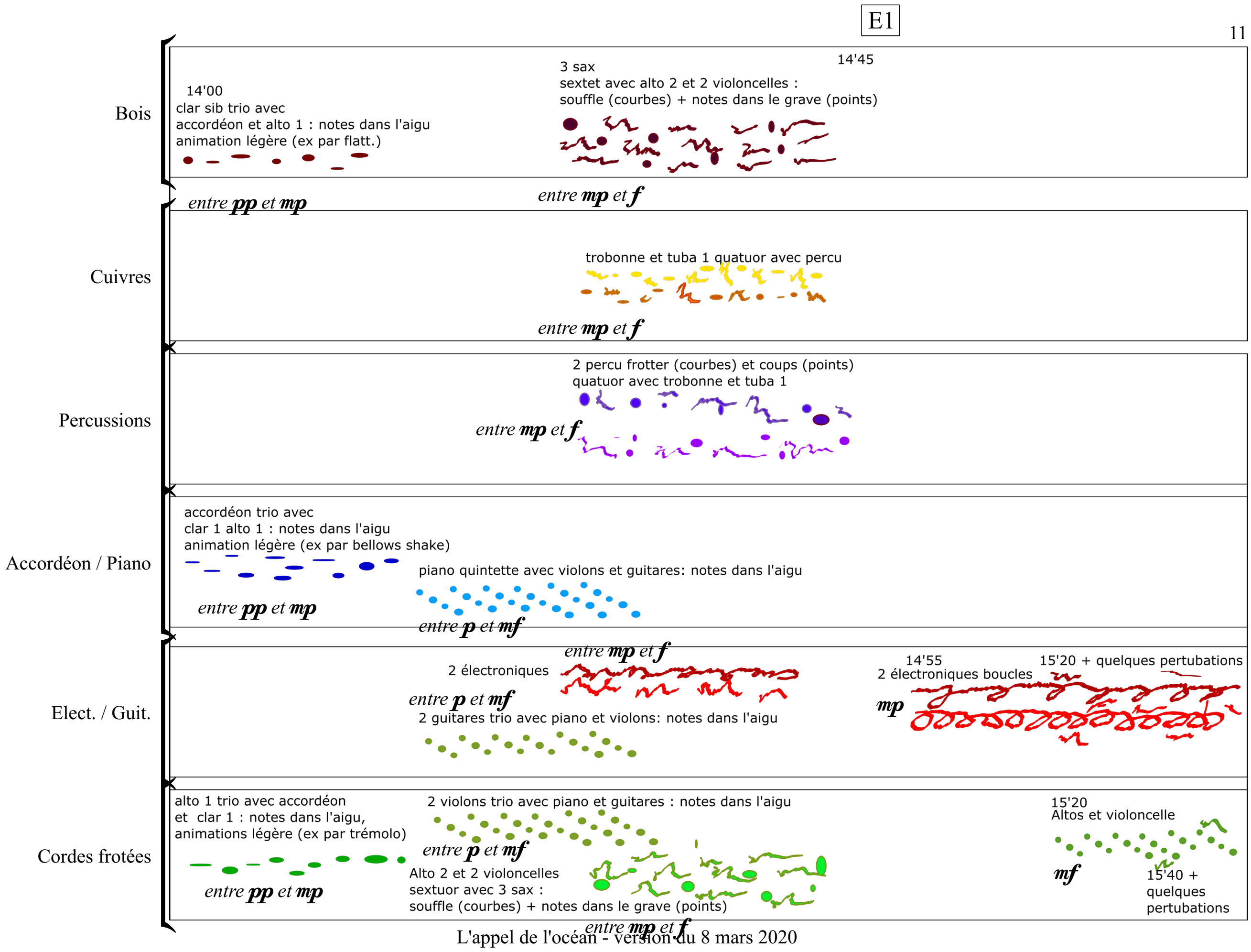


Figure 4. Page 11 of Laubeuf 2021 shows an example of graphical notation used to express musical textures — note the iconic signs used for loops (*boucles*).

completely indeterminate in terms of its instrumentation, but after a first series of exchanges with ONCEIM's musicians, a second version of the score was jointly produced by Guionnet and Blondy which precisely distributed all the sonic material comprised within the score to the various members of ONCEIM. But more generally, such a process of determination can also be seen as a way for ONCEIM's musicians to fully make the pieces their own. In this setting, scores are not treated as "finished" objects, but rather as objects that can be collectively transformed, edited, or appropriated. It is thus not rare to observe that, during the rehearsing process, entire sections were in fact modified or simply discarded, resulting in significant material alterations of the composer's score (see Galiay 2018, figure 7 for an example). In that perspective, the dialectics of indetermination and determination which is at work in such comprovisation pieces is precisely what makes possible the distribution of creativity and authorship (see [15] for similar insights).

The rehearsing of comprovisation pieces also involved processes of transcription and renotation. Noetinger 2016 provides the clearest example of such processes. As discussed above, this piece involved the musicians emulating as closely as possible short electro-acoustic audio samples. Here, some musicians simply chose to transcribe some of the samples in standard notation, in order to make their underlying metrical structure more explicit and optimize the ensuing rhythmical coordination. Figure 9 shows an example of the transcriptions made by the pianist for a few audio samples of Noetinger's piece. An alternative solution was simply to print out the waveform of some samples and to use such graphic representation as a kind of score, to ensure better temporal coordination between the various individual parts (see figure 11). Ultimately, the score provided by Noetinger was wholly renotated by ONCEIM so as to make it easier to parse and perform, from a series of individual scores, which simply prescribed the overall temporal sequence of the audio samples to be emulated (see figure 10), to a more detailed conductor score, which contained the same information but organized in a more traditional way, with the various instruments on the y-axis and time on the x-axis (see figure 8).

Finally, and strikingly, notation seems to play an ineliminable role when it comes to the long-term life of such comprovisation pieces, even the more "oral" ones such as Radigue 2015. For Radigue's piece, a textual score has indeed progressively been produced as part of the overall creative process. Such a score serves several functions.

Figure 6. Instruction page for the deontic piece Normand 2016, performed by ONCEIM with the GGRIL ensemble, where musical form is created by laying down playing cards, prescribing different configurations, interactions, and materials for the musicians.

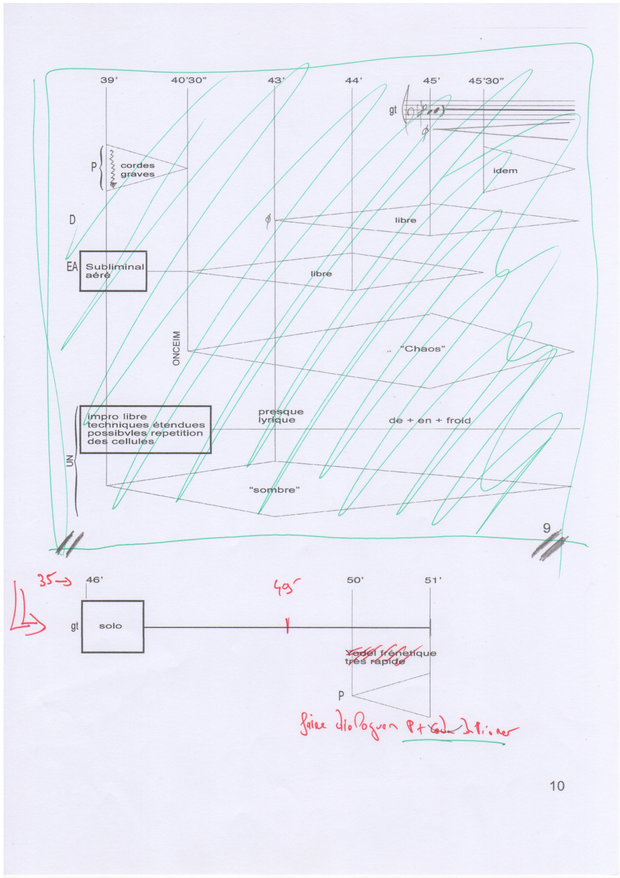


Figure 7. Score example of Galiay 2018, where a 7 minute part was discarded during the rehearsals.

First, it is generally read at the beginning of a new rehearsal, to allow the musicians to quickly remember what the piece is about and the state of mind that the musicians are supposed to achieve (aesthetic function). Second, it serves as a memory aid for the conductor of the piece (usualy Frédéric Blondy), laying out the entrance order of the various instrumental groups (conducting function). Note that this part of the score is constantly in flux, with new versions produced depending on the performing conditions and/or the desire to explore new orchestral mixture in how the instrumental groups will follow one another. Third, it centralizes the performing knowledge of the orchestra members, by explicitly notating the various sonic material that each instrumental group is supposed to produce. The score thus acquires a normative function, which is made particularly salient when a new musician joins the orchestra and has to learn how to perform the piece: for example, the score precisely prescribes how to achieve the desired microbeats effect for the accordion player (see figure 12), or the precise pitches the strings are supposed to play in various parts of the piece. In other words, notation plays a crucial role in the second life of the piece, i.e., in how the piece is transmitted from ONCEIM's musicians — who originally received the piece from Eliane Radigue in a purely oral fashion— to the broader circle of musicians (instrumentalists, sound engineers, etc.) with whom they interact.

4. ROLE OF ELECTRONICS

ONCEIM counts two musicians playing electronic instruments: Arnaud Riviere, who played in the orchestra from 2011 to 2020 (replaced by Jean-Philippe Gross from 2021), and Diemo Schwarz (first author of this article). Riviere and Gross both play analog low-fi electronics, feedback mixer, and cassette recorders. Schwarz plays a digital software system developed since 2005 in MAX [4] called CATART [5] [16]. The specificity of CATART is its use of corpus-based concatenative synthesis [17, 18], where all generated sound is based on real (acoustic, environmental, instrumental) sound, recorded before or even during a performance. These sounds are then selected, shaped, and transformed using embodied gestures, captured by tangible interfaces [19].

The presence of electronic instruments presents a specific challenge for the composers in their endeavour to create a piece for ONCEIM: while expectations exist for all the acoustic (and electric) instruments, nothing can be known in advance about the specific capabilities of the highly singular and bespoke electronic instruments used in ONCEIM before consulting the musicians who assembled them, since there is no standard organology for such instruments.

[4] http://cycling74.com/products/max
[5] http://ircam-ismm.github.io/max-msp/catart.html

However, and perhaps surprisingly, as seen in the last column of table 2, most of the time, there is no special role for the electronics. Instead, it is considered just like the other acoustic instruments, capable of spontaneously creating musical gestures, timbres, and textures, and being reactive to the propositions made by the other musicians and to the ever-changing musical situations. This is underlined by the fact that both electronic musicians play over individual loudspeakers placed behind them, and not over a general PA system, installed left and right of the stage, in order to convey a distinct acoustic position of each electronic musician in the spatial image of the orchestra. At most, there is sometimes the more or less unspoken expectation by the composer to make use of the extended frequency range the electronics is capable of, i.e., to produce sub-bass sounds (via the subwoofers usually present for a concert), or extremely high pitches.

There are only seven pieces for which the composer has ascribed a specific role to the electronics. In four of these pieces, one or both of the electronic musicians had to record specific parts of the concert, in order for them to serve as a source material for later re-evocation of the orchestra's timbre, with specific transformations. This requirement was always based on the composer's previous knowledge of the capability of the electronics (using corpus-based concatenative synthesis for Schwarz's digital instrument, and cassette recorders for Riviere's or Gross's setup), acquired by prior discussion with the musicians. In the three remaining pieces, the electronics had to create an overall sonic atmosphere, either by playing field recordings (Walker 2017), by capturing and freezing all harmonic partials played during the piece, saturating the frequency spectrum (Badaroux 2015), or by creating the sound of a "black mass" that would engulf all the other instruments (Mariage 2013). These last two examples, together with Guionnet 2021 —in which the recordings are to be played back highly distorted and very loudly— could be interpreted as revealing a subtext of electronics as a menace or negative force within the music.

The electronic instruments played by Riviere, Gross, and Schwarz also raise specific issues in terms of their notation. Given the idiosyncrasies of such instruments, composers sometimes tend to adopt a looser approach when notating for such instruments, and to rely more extensively on the know-how of the musicians. This is particularly striking for the pieces making extensive use of standard notation. For example, a passage of Naegelen 2019 simply asks the electronics to perform "scattered waves" modeled after the harmonic footprints of the multiphonic played by the solo clarinet (figure 13), whereas the other instrumental parts for the same passage are notated in a much more precise way. A similar treatment of the electronics would be found in Tetreault 2019. Conversely, notation for electronics is largely on par with that of the other instruments when the piece mostly relies on textual or graphic notation (e.g. O'Malley 2014, figure 3). In other words, the notation for electronics seem to exhibit an intrinsic indeterminate quality, at least within our corpus.

GAUCHE	SOUFFLE	SOUFFLE	SOUFFLE	CLIC SOLO
VIOLON	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire une fois
VIOLONCELLE	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire trois fois – effet stéréo
CONTREBASSE	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire une fois
EUPHONIUM	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire trois fois – effet stéréo
TROMPETTE	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire une fois
CLARINETTE	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire une fois
SAXOPHONE TÉNOR	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire une fois
BATTERIE	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire une fois
GUITARE	Crescendo de 0'00 à 2'00 avec Clic solo à la fin – MF	Crescendo de 2'10 à 3'10 avec Clic solo à la fin – MF	Crescendo de 3'15 à 3'30 avec Clic solo à la fin – MF	Entre 3'30 et 8'30 le faire une fois
ARNAUD	Crescendo de 0'00 à 2'00	Crescendo de 2'10 à 3'10	Crescendo de 3'15 à 3'30	Entre 3'30 et 8'30 le faire une fois
				Sci-Fi2 x 2 dans le même temps

Figure 8. The final conductor score produced by ONCEIM for Noetinger 2016.

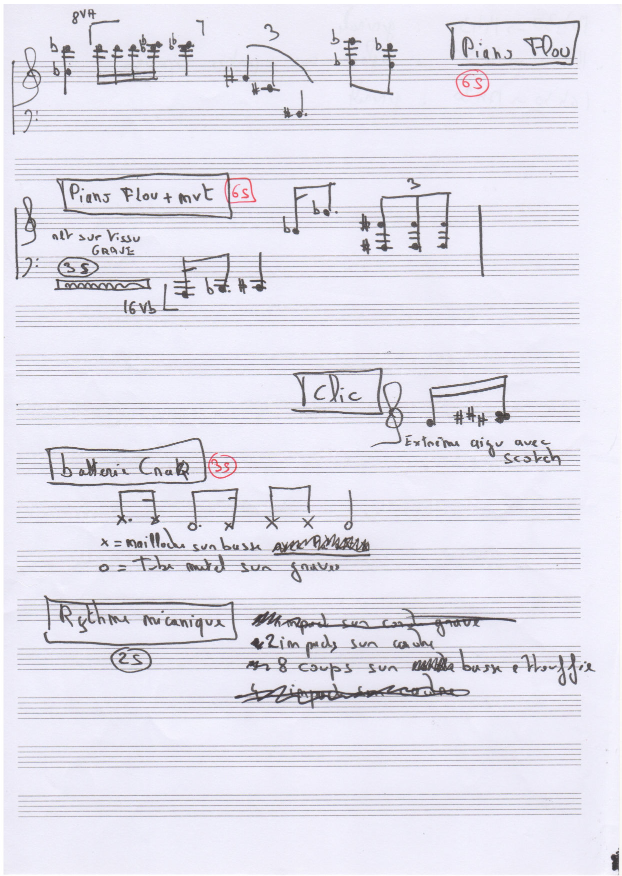
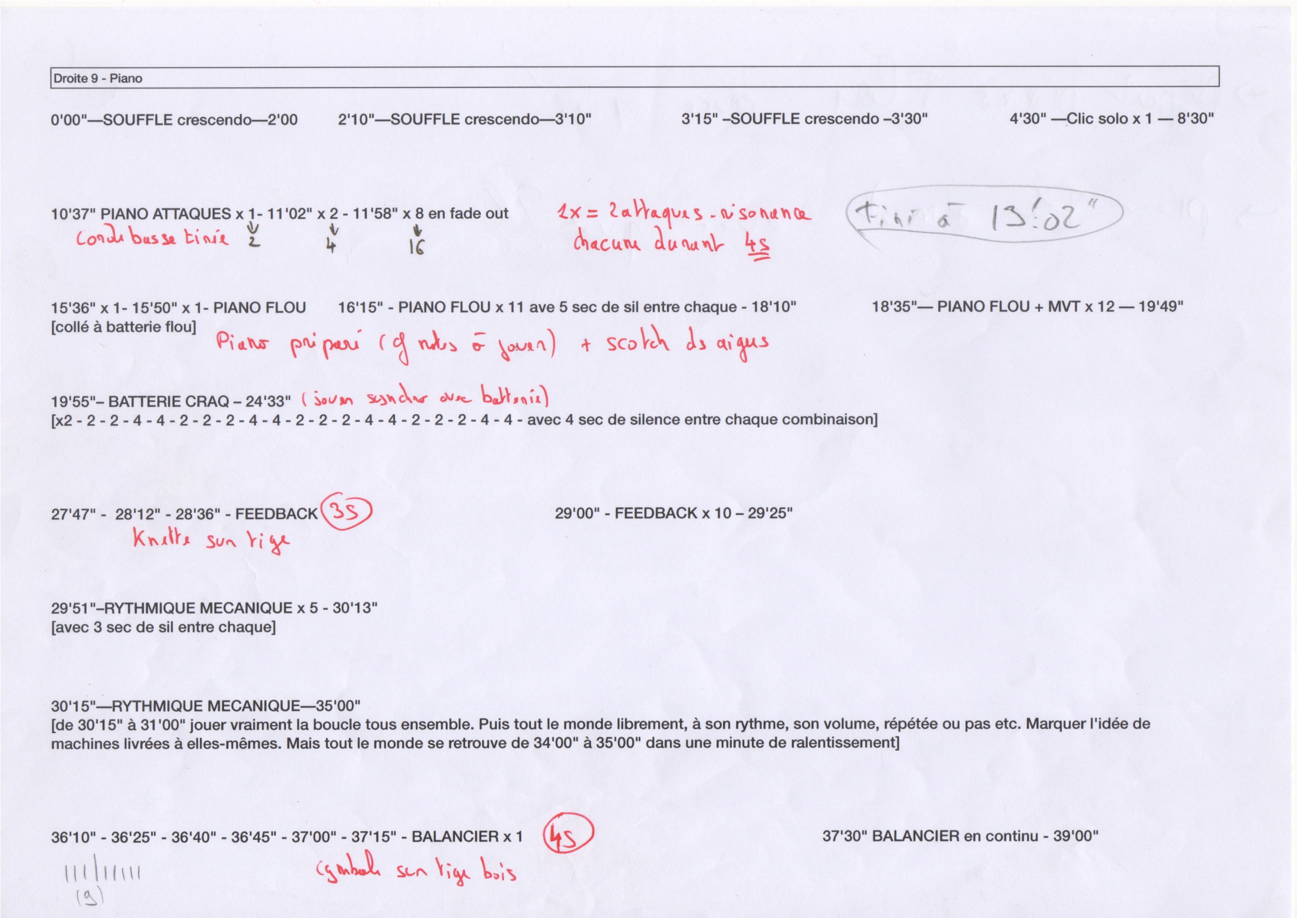



Figure 9. An example of the transcriptions made by ONCEIM for Noetinger 2016.

5. DISCUSSION

We have shown in this paper that the collection of 23 comprovisation pieces produced for ONCEIM covers a large variety of approaches to comprovisation, notation, transmission, and use of electronics. After having discussed in some detail how ONCEIM approached comprovisation,



Droite 9 - Piano

0'00"—SOUFFLE crescendo—2'00 2'10"—SOUFFLE crescendo—3'10" 3'15" –SOUFFLE crescendo –3'30" 4'30" —Clic solo x 1 — 8'30"

10'37" PIANO ATTAQUES x 1- 11'02" x 2 - 11'58" x 8 en fade out
2 4 16
1x = 2 attaques - résonance chacune durant 4s
Finir à 13'02"

15'36" x 1- 15'50" x 1- PIANO FLOU 16'15" - PIANO FLOU x 11 ave 5 sec de sil entre chaque - 18'10" 18'35"— PIANO FLOU + MVT x 12 — 19'49"
[collé à batterie flou]
Piano préparé (9 notes à jouer) + scotch ds aigus

19'55"– BATTERIE CRAQ – 24'33" (jouer synchro avec batterie)
[x2 - 2 - 2 - 4 - 4 - 2 - 2 - 2 - 4 - 4 - 2 - 2 - 2 - 4 - 4 - 2 - 2 - 2 - 4 - 4 - avec 4 sec de silence entre chaque combinaison]

27'47" - 28'12" - 28'36" - FEEDBACK 3s 29'00" - FEEDBACK x 10 – 29'25"

29'51"–RYTHMIQUE MECANIQUE x 5 - 30'13"
[avec 3 sec de sil entre chaque]

30'15"—RYTHMIQUE MECANIQUE—35'00"
[de 30'15" à 31'00" jouer vraiment la boucle tous ensemble. Puis tout le monde librement, à son rythme, son volume, répétée ou pas etc. Marquer l'idée de machines livrées à elles-mêmes. Mais tout le monde se retrouve de 34'00" à 35'00" dans une minute de ralentissement]

36'10" - 36'25" - 36'40" - 36'45" - 37'00" - 37'15" - BALANCIER x 1 4s 37'30" BALANCIER en continu - 39'00"
(9)
cymbale sur tige bois

Figure 10. First version of the text score of Noetinger 2016 for piano, giving the timing and names of the various audio examples to be emulated.

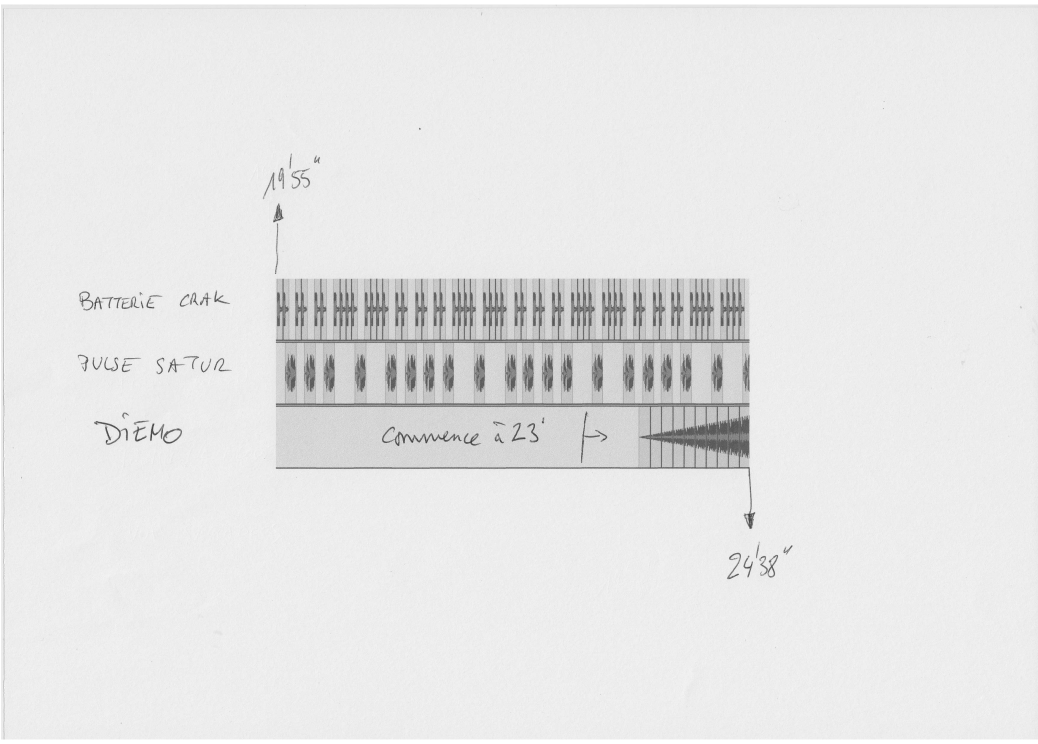


Figure 11. Additional graphic score for Noetinger 2016 suggested by the ONCEIM musicians, giving the precise temporal synchronisation and dynamic evolution of one section of the piece via a screenshot of the multi-track arrangement of the audio score.

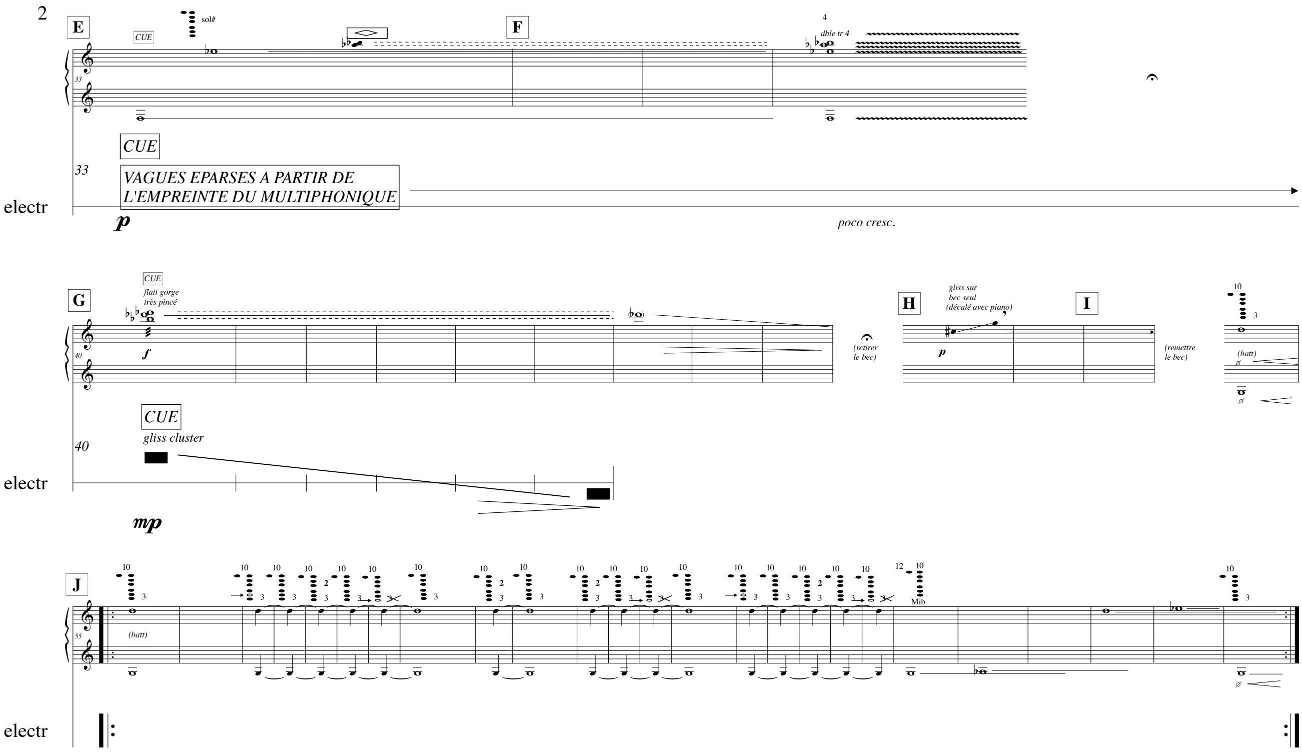


Figure 13. Excerpt from the score for electronics of Naegelen 2019. The first system is the clarinet part which serves as conductor in this concerto for clarinet and orchestra.

OCCAM OCÉAN

Cuivres
- Tuba, trombone, euphonium : notes pédales
- notes le plus piano possible
- harmoniques de ré (en ut). Cf. Clarinettes
- pas de respiration circulaire mais des relais => donc pas de respiration synchronisée
- être très attentif aux entrée et sorties. Fade-in et fade-out

Accordéon
- Note grave Sol en dessous de la clés de fa
- Pour battements : .ajouter des demis tons et clusters qui s'élargie
  .Jeux sur arrivées d'air entre main droite et main gauche

Figure 12. Excerpt from the "secondary" score produced by ONCEIM for Radigue 2016.

two important questions remain. First, can the modes of construction and transmission observed here apply to other large improvisation ensembles —for instance, the Splitter Orchester (Berlin), Ensemble UN (Bordeaux), Insub Meta-Orchestra (Geneva), GGRIL (Rimouski, Quebec)— or are they (partly or entirely) specific to ONCEIM?

A large part of the answer lies in the singular organisational setup of ONCEIM, with its founder and leader Frédéric Blondy *not* being one of the performing musicians, but instead serving as a critical external listener, and as an interface between the composers and the ensemble. Here, such an organisation seems to provide an advantage in how ONCEIM is able to work with composers, creating procedures that channel and optimize the exchange and transmission processes between the composers and the ensemble. Further work should be conducted to contrast ONCEIM's comprovisational output to the approaches and processes of other ensembles.

Second, what is the relation between the various dimensions of comprovisation introduced in section 2 above and the aesthetic appreciation of such comprovisation pieces? As could be expected, the comprovisation pieces commissioned by ONCEIM have been received with various degrees of appreciation by its members. In particular, timeline-based pieces seem to have been generally considered by many musicians as too stringent and hampering an organic development of the musical ideas. Just having to watch the timer constantly can distract from actually listening to the musical development — to the point that the ensemble now explicitly asks the composers it commissioned works from to avoid relying on a timer, but rather on more subjective forms of temporal organisation, which allows the musicians to mobilize qualities of interaction and coordination that are more typical of collective improvisation. Moreover, the musically and psychologically most satisfying pieces seem to have generally been the ones for which the composer had a good knowledge of the specifics of each musician, either because he or she knew them, or took the time to contact them beforehand, and when the material was proposed by the musicians themselves, as in Radigue 2015 and Bosshard 2018. In other words, it is precisely the pieces that take advantage of the fact that ONCEIM is an improvisation orchestra —which, as such, favours a construction of the performance through listening and interaction— that seem to have been both more relevant and pleasing for ONCEIM's musicians. Here again, further work should be conducted to understand in more detail the aesthetic processes that underlie the reception of such comprovisation works, both from the point of view of the musicians involved and from the point of view of their audience.

Acknowledgments

We'd like to thank Frédéric Blondy for valuable input and discussions, and the musicians, administrators, and technicians of ONCEIM for their openness to participate in scientific studies. This work is partially funded by the *Agence Nationale de la Recherche* grant ANR-17-CE27-0021 "MICA" to C.C..

6. REFERENCES

[1] S. Bhagwati, "Glossaire raisonné," *Circuit: musiques contemporaines*, vol. 28, no. 1, pp. 15–22, 2018.

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

[3] E. Clarke, M. Doffman, and L. Lim, "Distributed creativity and ecological dynamics: A case study of Liza Lim's 'tongue of the invisible'," *Music and Letters*, vol. 94, no. 4, pp. 628–663, 2013.

[4] C. Canonne, "Improvisation et processus compositionnel dans la genèse de fenêtre ovale de Karl Naegelen," *Critical Studies in Improvisation/Études critiques en improvisation*, vol. 10, no. 1, pp. 1–28, 2014. [Online]. Available: http://www.criticalimprov.com/article/view/2965

[5] C. Canonne and A. Robert, "An orchestra faces the test of indeterminacy: John Tilbury and the Sans project," *Biens Symboliques = Symbolic Goods*, no. 8, pp. 1–24, Jun. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03478235

[6] C. Canonne, "Rehearsing free improvisation? An ethnographic study of free improvisers at work," *Music Theory Online*, vol. 24, no. 4, 2018. [Online]. Available: https://mtosmt.org/issues/mto.18.24.4/mto.18.24.4.canonne.html

[7] L. Goupil, P. Saint-Germier, G. Rouvier, D. Schwarz, and C. Canonne, "Musical coordination in a large group without plans nor leaders," *Scientific Reports*, 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03019776

[8] S. O'Malley, "Gruidés," CD, label DDS, 2015. [Online]. Available: https://boomkat.com/products/gruides

[9] B. Denzler and ONCEIM, "Morph," CD, label Confront Recordings, 2015. [Online]. Available: https://confrontrecordings.bandcamp.com/album/morph

[10] Éliane Radigue, "Occam Ocean Vol. 2," CD, label Shiiin, 2019. [Online]. Available: http://www.shiiin.com/shiiineer2.php

[11] P. Ablinger, "ONCEIM plays Notes & Bloc-Notes," Digital Album, label Remote Resonator, 2019. [Online]. Available: https://remoteresonator.bandcamp.com/album/onceim-plays-notes-bloc-notes-by-peter-ablinger

[12] P. Bosshard, "Sillons – Reflets," CD, label Potlatch, 2021, two pieces performed by ONCEIM and string ensemble CoÔ. [Online]. Available: https://www.corticalart.com/product/patricia-bosshard-sillons-reflets

[13] N. Cook, *Music as creative practice*. Oxford University Press, 2018.

[14] E. Payne and F. Schuiling, "The textility of marking: Performers' annotations as indicators of the creative process in performance," *Music and Letters*, vol. 98, no. 3, pp. 438–464, 2017.

[15] C. Canonne, "À la croisée d'un double continuum? Retour sur une expérience de co-création entre un compositeur et cinq improvisateurs," *Circuit: musiques contemporaines*, vol. 30, no. 2, pp. 11–25, 2020.

[16] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-time corpus-based concatenative synthesis with CataRT," in *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, Montreal, Canada, Sep. 2006, pp. 279–282. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01161358

[17] D. Schwarz, "Corpus-based concatenative synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 92–104, Mar. 2007, special Section: Signal Processing for Sound Synthesis.

[18] D. Schwarz, "Concatenative sound synthesis: The early years," *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, Mar. 2006, special Issue on Audio Mosaicing. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01161361

[19] D. Schwarz, "The sound space as musical instrument: Playing corpus-based concatenative synthesis," in *Proceedings of the Conference for New Interfaces for Musical Expression (NIME)*, Ann Arbor, MI, USA, May 2012, pp. 250–253. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01161442

OSSIA SCORE 3

Jean-Michaël Celerier
ossia.io
Talence, France
jcelerier@ossia.io

Myriam Desainte-Catherine
Univ. Bordeaux, LaBRI- UMR 5800, CNRS, INRIA
Talence, France.
myriam@labri.fr

Pia Baltazar
ossia.io
Talence, France
pia@ossia.io

ABSTRACT

The ossia system has been introduced in 2015 as a notation for interactive scores. We present the result of seven years of usage and improvements to the *ossia score* software which acts as both an editor and player for such scores, and how it morphed from a simple OSC-only control sequencer to a fully-fledged multimedia system supporting live audio and video processing, live-coding with multiple embedded programming languages, communication with a variety of software and hardware and adaptations for more traditional music creation such as support for varying tempo and time signatures. In particular, we mention a few "original sins" and implementation mistakes done at the beginning of the software engineering process and how they had to be fixed.

1. INTRODUCTION

In 2015, the foundations for a method of authoring interactive scores were presented to the TENOR conference, in a paper titled: "OSSIA: Towards a Unified Interface for Scoring Time and Interaction"[6].

This paper presents the result of seven years of active use in a variety of artistic settings, what has held and what hasn't, and how the ossia paradigm for interactive scores was extended. It gives an overview of the various research aspects that were studies for interactive scores so far, reports on their status and talks about upcoming research.

From the very beginning, the implementation software, originally named **i-score** and renamed **ossia score** – to highlight the from-scratch rewriting of the system and its inclusion in a larger software ecosystem – , has been conceived as a platform for fostering art-science research.

1.1 A primer on interactive scores

The visual syntax of interactive scores as defined in ossia is composed of a few basic elements, visible in Figure 1 which are combined together by the composer to create scores. Due to user feedback and evolutions of the system, some names have been changed since [6]. These changes are mentioned here:

- A *process* is an object which performs a computation over time; it is similar to e.g. a Max/PureData object or SuperCollider ugen. A specificity of processes is that they can be used to define both non-temporal content (e.g. an audio filter such as a distortion, an oscillator...) or content with an implicit temporality (e.g. an automation curve, a MIDI piano roll) directly.
- An *interval* (renamed from Constraint in [6]) allows to define the span of time over which a set of processes will run.
- A *time sync* (renamed from Time Node in [6]) allows to synchronize the start and end of multiple intervals, either at a fixed time or by waiting for an external event: OSC message, mouse click, etc. In the latter case, the external event is represented by a visual object called *trigger*, shown at the top of the time sync.
- A *condition* allows to enable or disable a set of intervals at run-time depending on the truth value of an expression at the time at which it is evaluated in the score (after the end of the previous intervals and before the start of the next intervals).
- A *state* (renamed from Event in [6]) indicates the start and end of an interval. States can carry messages such as OSC packets that will be sent immediately when the state is reached during the execution of the score.
- Time syncs and intervals form a graph which we call a *scenario*. A *scenario* is a process; this means per the definitions of intervals and processes that intervals and scenarios allow to define a hierarchical system.

2. SEMANTIC EVOLUTION

We detail here how the visual language and system have evolved since its introduction in 2015 and the areas of focus for the research and development.

2.1 Loop semantics

The original loop semantic, described in [5] was through a specific process which implemented a looping behaviour for a single temporal interval, isolated from its parent. This added a hierarchy level, and separated entirely the looped content, from the overall loop duration.

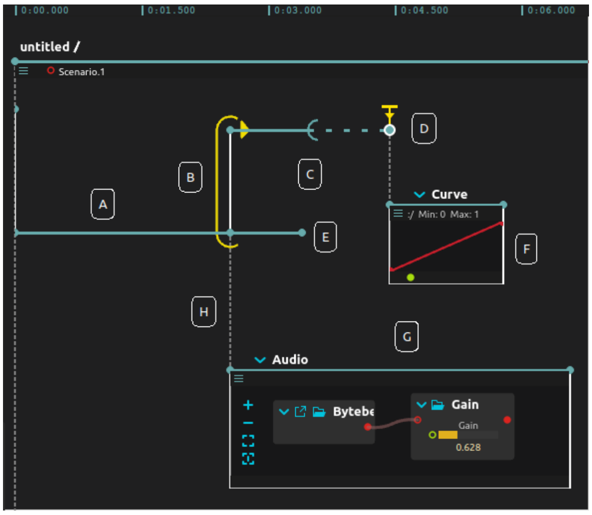


Figure 1: Visual syntax of ossia score. **A**: an *interval* of fixed duration. **B**: a *condition*. **C**: an interval of variable duration; the minimum can in this example be adjusted with a visual handle. **D**: the blue dot is a *state*. The white circling around it indicates that the state contains messages to be sent. The yellow T on top indicates a *trigger*. **E**: an empty state. **F**: an interval which contains an automation *process*. **G**: an interval which contains two connected audio processors, represented as dataflow nodes. **H**: a *time sync*. Note that this score is itself contained in a *scenario* process, itself a child of an *interval* which is the hierarchical root of the score.

After years of experimentation, while we observed that this method was able to cover a fair amount of use cases, we also noticed how unwieldy it was for artists as it made scores harder to read. The visual similarities between the looped content, and the parent interval containing the loop object, made understanding the behaviour of a score at a glance more complicated. In addition, looping could not easily be changed "on the fly": artists would often want to simply loop a sound for some time, but our original method required conceiving the temporal structure of the score beforehand.

Study of the use cases for loops led us to a separation of the original loop process into two distinct features:

- Looping an individual process through a simple UI control: each process with a temporal information can be looped: sound files; automations, etc. The process' actual duration is decoupled from its parent interval's duration. This kind of loop is outside of the temporal structure of the scenario, it is merely a filter which can be applied to any process, and can be toggled at will at any point in the user interface.
- A generic go-to mechanism for the scenario. Visually a new primitive, this is actually just an interval of duration zero, for which the restriction of "always going forward" is relaxed. This mechanism enables more generic cases than the previous loop primitive, and makes writing state machines in *ossia score* an easy task, as the end of an interval can now start any other interval in a score.

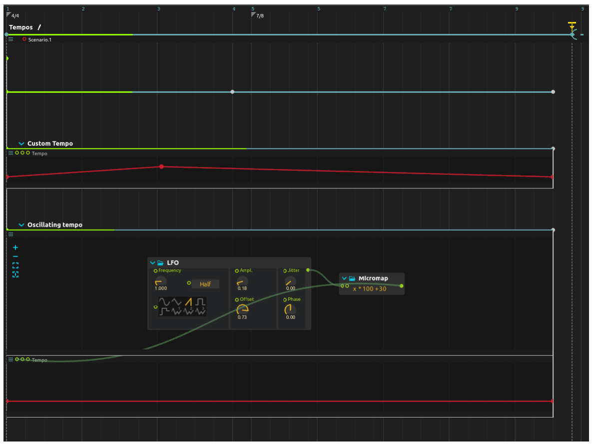


Figure 2: Example of tempo and musical grid variations: multiple time-line have different tempo curves. The first timeline will follow the tempo of the parent. The second will follow its own tempo curve. The third one takes its tempo from a low-frequency oscillator: the speed of that timeline will constantly oscillate. The grid switches from $\frac{4}{4}$ at the first bar, to $\frac{7}{8}$ at the fifth bar; bar 4 is an implicit $\frac{1}{4}$ bar as the start of bar 5 can be arbitrarily moved to simplify user interaction.

2.2 Musicality, polyrhythms, quantization

For the longest time, even though the interactive scores project was born out of a musical context, it did not provide any tooling to help the composer used to traditional and western musical cues. The software provided no beat grid, no notion of tempo, and no quantization to allow synchronizing multiple competing elements on a single beat which would be determined by the grid in which the musicians are playing.

In *ossia score 3*, we introduce the ability to specify metric modulations in the score (time signature changes through a musical grid, and tempo changes through automation curves). Both features can be seen in Figure 2.

Both musical grid and tempo curve are optional properties of an interval. If an interval does not have either, it and its children processes will stay synchronized with respect to the parent interval's grid, recursively. At any point, it is possible to dissociate a child interval from the parent's musical grid or tempo, in the first case by enabling a custom grid (through a GUI widget visible when the user selects an interval), in the second case, by adding a tempo process to it.

The tempo process is by default an automation curve which allows to define a fixed tempo variation. This process also provides inputs: these inputs allow to control the tempo, or the playback position, through the score. This enables writing scores where the execution speed of a part comes from an external input, a mathematical function, or any combination of either.

Quantization is another feature which is propagated hierarchically: by default, interactive triggers will wait for the next quantized time as defined by their parent, for instance the start of the next bar or next quarter note. Any interval can redefine it instead, to allow some parts of a score to quantize to the next bar, and other parts to quantize to the next eighth note relative to this interval's internal musical grid and current execution time. For instance, given a fixed BPM of 60, if a trigger's boolean expression becomes true at 32.3 seconds since the interval started, it will actually trigger at 33 seconds if this interval's quantization is set to quarter notes, and 32.5 seconds if it is set to eighth notes.

2.3 Data processing, combining data flow and time flow

The original implementation focused on control signals: sending and receiving OSC and MIDI messages, for instance. It quickly became obvious that having to start a whole other software for the sake of playing a synchronized sound file and setting up OSC communication was cumbersome for the users of the software. This led to research about introducing an audio pipeline into the system. A second requirement quickly came up: applying audio effects such as VST plug-ins or Faust signal processors, as using systems such as Soundflower or JACK for routing audio between software was also too complicated for a part of the target user base. By then it became obvious that a built-in dataflow pipeline was required, with the usual affordances provided by the patching software paradigm: processes (called nodes, objects, ugens in other systems) with multiple input and output ports (controls, audio or MIDI inputs and outputs), connected together through virtual cables.

The main question was then: how to make a dataflow pipeline, which describes an invariant computation, fit as part of an interactive score where processes constantly change over the execution of the score. The solutions for this were twofold:

- In addition to the ability to connect cables showcased in Figure 3, the ports in *ossia score* can read and write data directly from a global environment, through OSC-like addresses. When a cable is connected, that address is overriden.
- Cables have two axes of configurability: direct and delayed, strict and glutton. Direct (the default) means that if process A and B are running at the same time in real-world clock, B processes the output of A just like in a traditional patcher. Delayed means that the cable acts as a delay line between A and B: if A starts a second before B, B will receive all the data that was produced by A since it started and thus be a second late. Glutton (the default) means that the connection of a cable has no impact on whether A and B runs, strict means that if A and B are connected, and either is not running, the other will also be disabled. This is mainly useable as a performance optimization, to make sure that complicated signal processing pipelines are not kept running if the data source has not started yet.

This has led to the introduction of a dual-visualization for intervals: at any point, the user can switch between the time-line view, where the horizontal axis is used to represent the evolution of time, and the so-called nodal view, which shows all the processes as nodes similar to traditional

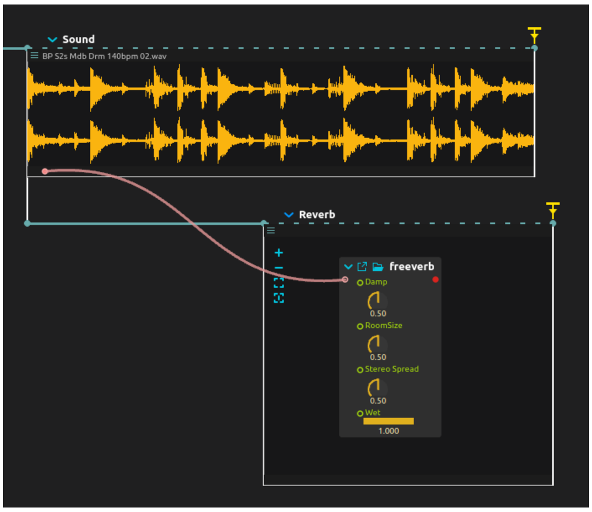


Figure 3: Applying an effect to a sound file with an explicit cable.

patching software. This way, the user can focus on editing temporal behaviour such as synchronizing automations with soundfiles, etc. in a timeline, and on editing complex signal processing flows in a more adapted user interface.

2.4 Live visuals

Shortly after artists querying about "simply playing back a sound file", the same question happened for video. Prototypes of efficient OSC-controlled video players were made [1] yet this also proved insufficient in terms of synchronization precision and capabilities.

Thus, score 3 introduces support for video playback and video processing through the same data-flow system as for audio, showcased in Figure 4. This has been done with the Qt RHI library. [2] which provides a small wrapping layer over modern GPU APIs: Direct3D 11, Vulkan, Metal (and more traditional OpenGL).

Some complexity for the implementation of this feature comes from the fact that processing happens on the GPU, not on the CPU: working with a modern graphics API is about preparing a set of commands that are being send to the GPU for rendering at a regular interval. The GPU generally works at a 60 Hz rate while the rest of *ossia score* follows the audio buffer rate. The implementation works by creating a mirror graph of GPU nodes which lives in its own thread. While cables look like they pass texture data, what they actually do is instruct the source node to render on a texture stored in the input of the sink node; this enables multiple nodes to render on the same target. Render order and blending options are not made available to the user yet and is a work-in-progress for an upcoming minor release.

Multiple outputs are possible; each output will only render the nodes that are directly connected to it. For instance, it is possible to render to multiple windows, or to both a window and a system such as Spout or shmdata.

[1] https://github.com/OSSIA/ossia-videoplayer
[2] https://www.qt.io/blog/graphics-in-qt-6.0-qrhi-qt-quick-qt-quick-3d

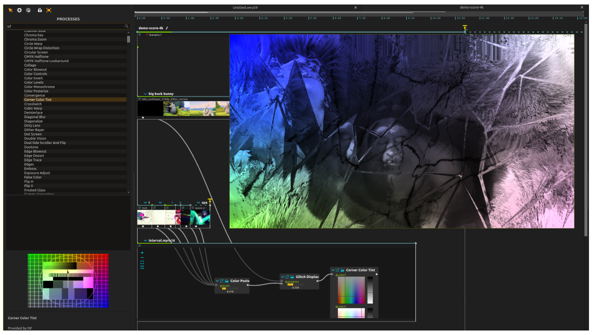


Figure 4: Mixing multiple video sources and applying GPU video effects through shaders following the Interactive Shader Format specification.

Leveraging the GPU has some advantages: the author has for instance prototyped a particle renderer process which was able to maintain 60 FPS for ten million particles with a GTX 1080 graphics card.

One feature which is not available is the delayed cables for GPU textures: this is because unlike audio or control data which can be expected to be bufferable for reasonable durations on the order of minutes or even hours, video data memory usage grows too fast to make this viable. [3]

2.5 Pragmatisms for live playback of interactive scores

Scores in *ossia score* until now allowed for what could be called "known unknowns" in a show. The classic use case is: a sound has to trigger ten seconds after an actor reaches the center of the stage. The actor being human implies that the time taken to reach the center of the stage is unknown: the durations of the sound start cannot be fixed as a duration from the beginning of the play, only from when the center of the stage is reached. Thus, the *ossia score* user can add a trigger point and use for instance position sensors to encode in the score the actual conditions for the sound starting to play, as well as add bounds for the time that the actor must take, to make sure that the score continues.

What remained was "unknown unknowns": consider a catastrophic case where all the sensors stopped working for unforeseen reasons. It does not make sense to account for such cases during the authoring of the score; yet in the midst of the action, it makes sense to still have a safeguard to allow a stage manager to take control of things manually, even if this does not respect the semantics originally written score: the show must go on.

Thus, multiple features were introduced over time to allow to override the written score as a last measure during playback:

- Triggers can be triggered by hand and through a remote interface.
- Interval speed can be adjusted live, independently for each interval.

[3] To give a reference point, merely storing 60 frames (one second) of raw 4K (0 × 0) textures in GPU memory with ARGB float32 texture format uses upwards of 7 gigabytes; most consumer GPUs do not even have that much memory. In addition, most GPUs are very limited in terms of maximum texture count: many high-end NVidia and AMD GPUs are limited to 4096 total allocations for a given GPU context

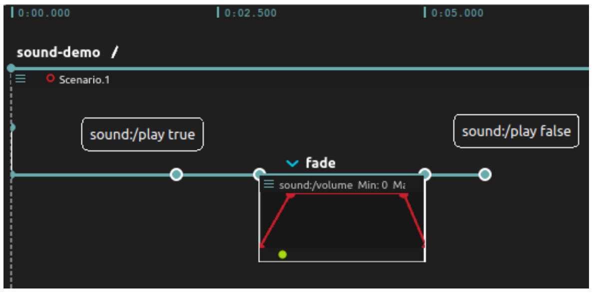


Figure 5: A score which controls an external sound file player through OSC.

- The "main playhead" can be moved: this is similar to transport in the usual audio-video software.
- Intervals can be started manually at any point in the score.
- States can be sent manually at any point in the score.

Starting an interval will follow quantization settings; implementation of quantization for starting states and global transport remains to be implemented. This enables very simple live-looping systems to be put in place.

Transport is an interesting feature, as it is in a sense equivalent to starting the execution of a software from any point in its code: it is not possible to ensure that it will always make sense. Two semantics are provided by *ossia score*, that the user can choose in the settings:

- Simply start executing from the given point visually.
- Try to compute the state in which every external system should be at this point, send them this state, and start executing.

Consider the score of Figure 5: it first sends a message to start an external sound playback device. If one wants to start the score from the 3 second mark, the `play` message has to be sent beforehand for the execution to make any sense.

2.6 Web

Having *ossia score* work on a web browser has been the result of a long process, which started in 2015 with Qt's NaCL port, then migrated to Asm.js and finally to WebAssembly, the web standard which enables compiling native C++ code and running it in a web browser.

The result of this work can be found at the address `https://ossia.io/score-web` which allows to write and execute simple scores in a web browser. Not all features are currently available, as more dependencies need to be recompiled for the WebAssembly target. The objective of this work is to allow *ossia score* to export a score as a web page so that it can be easily experienced over the internet.

2.7 Computational performance

One of the first supported system for audio effects in *ossia score* was Faust [8], the well-known DSP language. The software embeds the Faust compiler, which itself leverages

LLVM, the compiler framework. The C++ compiler `clang` is also based on LLVM; from there, it was relatively easy to introduce a C++-based add-on system [2].

The end goal of that add-on system is to ensure that the users of the software get the most performance out of their hardware: most native C++ software is compiled for a basic baseline in order to maximize compatibility. This means that someone with a recent Intel Skylake CPU most of the time does not benefit of the advanced vector instructions in that CPU, since it would break compatibility with people using older CPUs, unless the program author instructs the compiler to build multiple versions (which would multiply the size of the binaries). In contrast, by embedding the compiler inside the software, we can work towards rebuilding the hot paths directly on the user's computer, as we can probe the exact CPU instructions that are available there.

Right now, this feature is used only for external add-ons; when it will have undergone enough testing on enough hardware the plan is to migrate the various signal processing plug-ins which the software provides to that system.

2.8 Live-coding and scripting in scores

A certain focus was put on using score as a platform for live-coding. Multiple languages are embedded in the software and can be recoded on-the-fly [3], during the execution of the score:

- The first one was Javascript, through Qt's ES7 interpreter, and allowed to quickly devise which features made sense as actual processes implemented in native code.
- The Faust language integration supports live recompilation.
- The math expression language ExprTK is used [9].
- Likewise for the ISF shaders used for GPU visuals.
- PureData has been embedded thanks to libpd [1]; patches can be edited live. The `send` and `receive` objects are used to create GUI controls in *ossia score* which can be used to score parts of a PureData patch over time [4].
- C++ code can also be live-recompiled. This comes up in three different processes: Bytebeat (which executes Bytebeat expressions), Texgen (which generates 2D textures for visuals) and CPP JIT (a generic C++ object which can be used to implement general objects with any kind of input or output ports).

3. IMPLEMENTATION NOTES

Some fairly deep changes came up during the software development process, which make sense to discuss with the broader community mainly so that pitfalls in which the project fell can be avoided by others who would encounter similar situations.

3.1 Evolution of time-tracking

When the development was started, time was counted in milliseconds stored in double values. This was a mistake: due to the way double arithmetic works, we encountered after a few years cases where the execution would be stuck for scenarios running over the course of multiple days, for installations. This was due to the expectation $t + \epsilon > t$ not holding anymore for orders of magnitude of ϵ which start to be relevant to our use-cases. For instance, the following assertion does not hold for double-precision floating-point: $3600 * 24 * 1000 < 3600 * 24 * 1000 + 10^{-9}$. That is, after merely a day, adding a nanosecond does not have an effect anymore. Thus, we migrated in version 2 to using audio samples for storing time internally in the engine, while staying with the original time format for the user interface in order not to break the save format. This worked better, but now made it harder to work in preparation of the video format support, as many common video formats timestamps are not easily divisible by audio rates (for instance, 24 images per second versus 44100 samples per second). Thus, for *ossia score* 3, a migration to flicks was done: this unit, originally introduced by Oculus VR, is defined as a common multiple of most time formats used in media production. One second holds 705600000 flicks; timestamps for every common audio sample rate and video frame rate can be represented without loss of precision while keeping flicks an integral value.

3.2 Save format and JSON woes

ossia score leveraged from the very beginning JSON as its save format, as it is widely understood, and easily readable. However, the migration to flicks mentioned before had an unforeseen impact. The JSON specification does not mandate a minimal precision for storing numbers; in particular, the library we had been using, QJson which is part of Qt, makes the choice of converting every integral number to floating-point which had not been an issue until then, but looses precision for flicks representing long durations.

It was hence necessary to migrate to a library which supported 64-bit integers to allow us to safely store durations without data loss and without breaking compatibility with the existing format – the RapidJSON library was chosen.

3.3 Threaded networking

The first implementation of network protocols was threaded. Incoming OSC messages for instance would be processed in a specific thread; callbacks would be called from that thread. Users of the *libossia* library would have to take care of thread-safety for their own data structures. While this work was done in *ossia score*, in practice doing it this way was a mistake: many environments would only support single-thread callbacks (Python, Unity3D) and require a lot of busywork for repatriating data from the network threads to the main thread. A recent change was to use the Asio [7] library which provides an event-loop approach where the user of the API can choose whether processing must occur in a separate thread or not; doing so from the beginning would have saved dozens of hours of work with threads. Addition-

Figure 6: Nebula.

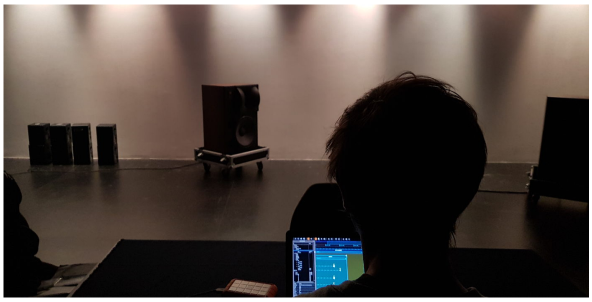


Figure 7: Quarrè.

ally, using the Asio library in *libossia* allowed us to leverage its support for multiple networking protocols: besides the traditional UDP, *libossia* and score now support OSC over TCP, Unix sockets (both stream and datagram), WebSockets and serial port, with various encodings. Unix sockets in particular are interesting when staying on the same machine: the author measured an improvement of 10-15% when measuring how many messages can be transmitted per second when compared to UDP on loopback network interfaces.

3.4 Forked libraries

Various libraries forked during the development process, mainly to provide more efficient versions of the originals without having to retain API compatibility which was not useful in our case since the software was being developed from scratch:

- A fork of Ross Bencina's OSCPack library for OSC performance, adaptability to different back-ends and modernization for compatibility with more recent C++ standards. [4]
- A fork of RtMidi [10] into libremidi [5] with many improvements: greatly reduced memory allocations, support for more backends (UWP on Windows, WebMidi with Emscripten, raw ALSA for sending large SYSEX messages), notification of newly available devices, and modernization for compatibility with more recent C++ standards. This fork also merges and improves the ModernMIDI library which provided SMF reading and writing.

4. ARTWORK MILESTONES

We would like to discuss here artworks made with *ossia score* which provided useful as feedback mechanism: they evidenced the need for new features which were developed in tandem and can now benefit the greater community.

4.1 Nebula

> Nebula is a sensory installation. The visitor enters a dark room. In the centre of the room, a black cylinder is placed on the floor. A coloured and luminous mist sculpture emerges from the top of the inert object.

Nebula (Figure 6) is an art installation created by the duo Les Baltazars. It is one of the first artwork which used the software at a "non-toy" scale: the score lasts for 30 minutes and contains hundreds of automations controlling evolving patterns of lights and mist. It was useful as a benchmark of the software, in particular for the UI rendering which had to show thousands of lines on-screen while maintaining acceptable performance, and was used a lot as part of optimization work, and for the device tree, which contains more than a thousand parameters.

4.2 Rain of Music

Rain of Music is a work-in-progress hybrid human-machine opera, which involves soundpainting for both human and robot performers. Technically, it has been used as a way to test various integrations of protocols inside *ossia score*, and most notably direct serial port control for the Metabots, the robots used for the performance.

4.3 Quarrè

Quarrè (Figure 7) by Pierre Cochard was one of the first works which explored multi-user interaction for a single score. Up to five participants are each given a phone; the score broadcasts instructions to each phone which will show distinct user interfaces over the duration of the score. The user interfaces have widgets which give back feedback to the score, which then uses it to control generative sound processes to create eerie ambiances and textures.

4.4 HERMÈS

HERMÈS by João Svidzinski [11] explores multi-user interaction, over the internet instead of in a same-room setting. It used *ossia score*, OpenStageControl and Max/MSP to provide a performance during the Journées d'Informatique Musicale 2021.

4.5 Carrousel Musical

The Carrousel was the largest incentive for introducing a native audio data-flow system in *ossia score*. It is a merry-go-round located in "L'Abbaye aux Dames" in Saintes, France, with multiple instruments and sensors. The songs contain

[4] `github.com/jcelerier/oscpack`
[5] `github.com/jcelerier/libremidi`

Figure 8: Jeu de la Marelle.

Figure 9: Phonemacore performance.

parts which record and replay the young player's musical performance with an improved rhythm. Dozens of VST instruments such as Kontakt and various effect busses are output in spatialized sound with low-latency, and an additional DMX light show. The project was first prototyped in Ableton Live and Max4Live but the performance was not adequate, which prompted the implementation of a custom solution.

4.6 Jeu de la Marelle

Laurence Bouckaert and Olivier Moyne devised an augmented game of hopscotch in their piece "Jeu de la Marelle" (Figure 8). Each square is a pressure sensitive area that is able to detect the impact of the stone thrown by the player. They can sense the weight distribution along their X and Y axis. Throwing the stone or stepping on either of these tiles then triggers a specific behavior of the system, through the RGB lighting on the ground, the octophonic sound system surrounding the player, and the visual projection on the wall. The score was able to run on a Raspberry Pi 4 computer connected to a multi-channel sound card, a video projector and an ArtNet node for controlling LED strips. Live audio and video processing is done with Faust and ISF shaders respectively.

This project was one of the impulsions for having a complete video pipeline, and making sure that the software was performant enough to be useable from low-power embedded computers such as Raspberry Pis.

4.7 Phonemacore

Phonemacore (Figure 9) was a musical project started in 2018 to explore the possibilities of *ossia score* for adding interactivity to popular-ish music (metal with simple $\frac{4}{4}$ and $\frac{12}{8}$ signatures). The performers were a guitarist, a keyboardist and a drummer: the band Phonema. Bass tracks were stored in the score. The song had 7 parts; a participant chosen from the public would play a video game made for the song. Whenever a level would be completed, the musicians would move on to the next part of the song, and keep playing until the player reaches a high enough score. This work was used as a basis to implement musical metrics, tempo changes, and quantization in the software, as those all had to be worked around during the creation of the piece.

4.8 Twitch live-scoring sessions

The author regularly presents live sessions on a Twitch channel [6] , where a score is created and played "live": the execution of the score runs at all time; process creations, connections and Faust, JS, ISF scripting are all done while the score is running. It is mainly used as a debugging exercise, to make sure that live editing is reliable enough for allowing live-coding performances to take place.

5. CONCLUSION

We gave an overview of the state of the research on the applicability of the ossia system for interactive scores of various forms.

An upcoming work is the extension to less art-centric use-cases: a student is currently working on protein diffusion sonification with the system, which opens a set of issues related to parsing and processing efficiently large amount of data.

Another ongoing issue is applying generative and procedural techniques for creating scores programmatically: providing an efficient API for allowing users to generate scores from Javascript code both offline and during the execution is a priority.

Acknowledgments

This work has been financed by: SCRIME, ANR, ANRT, Blue Yeti, Epic Megagrants, SNI, Région Nouvelle-Aquitaine, and kind private donations.

References

[1] Peter Brinkmann et al. "Embedding pure data with libpd". In: *Proceedings of the Pure Data Convention*. Weimar, Germany, 2011.

[2] Jean-Michaël Celerier. "A Cross-Platform Development Toolchain for JIT-Compilation In Multimedia Software". In: *Proceedings of the Linux Audio Conference*. Stanford, USA, 2019, pp. 37–42.

[3] Jean-Michaël Celerier. "Leveraging Domain-specific Languages in an Interactive Score System". In: *Sonic Arts Today* 10.1 (2018), pp. 28–33.

[6] `https://twitch.tv/jcelerier`

[4] Jean-Michaël Celerier. "Patches in a timeline with ossia score". In: *Vortex Music Journal* 9.2 (2021), pp. 1–14.

[5] Jean-Michaël Celerier, Myriam Desainte-Catherine, and Jean-Michel Couturier. "Graphical Temporal Structured Programming for Interactive Music". In: *Proceedings of the International Computer Music Conference (ICMC)*. Utrecht, The Netherlands, 2016, pp. 377–380.

[6] Jean-Michaël Celerier et al. "OSSIA: Towards a Unified Interface for Scoring Time and Interaction". In: *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*. Paris, France, 2015, pp. 82–91.

[7] Christopher Kohlhoff. *Boost.Asio*. 2008. URL: `https://www.think-async.com`.

[8] Yann Orlarey, Dominique Fober, and Stéphane Letz. "Faust: an efficient functional approach to DSP programming". In: *New Computational Paradigms for Computer Music*. Paris, France, 2007, pp. 65–96.

[9] Arash Partow. *C++ Mathematical Expression Toolkit Library (ExprTk)*. 2000. URL: `http://www.partow.net/programming/exprtk/`.

[10] Gary P Scavone and Perry R Cook. "RtMidi, RtAudio, and a synthesis toolkit (STK) update". In: *Proceedings of the International Computer Music Conference (ICMC)*. 2005, pp. 327–330.

[11] João Svidzinski. "HERMES v2 - WEB CONCERT COLLABORATIF EN TEMPS RÉEL". In: *Journées d'Informatique Musicale 2021*. AFIM. Remote conference, France, July 2021.

THE DECIBEL SCOREPLAYER – LEARNING FROM AND FOR USABILITY

Aaron Wyatt, Cat Hope
Monash University
Melbourne, Australia
{Aaron.Wyatt, Cat.Hope}@monash.edu

ABSTRACT

This paper outlines new developments in the Decibel ScorePlayer application for iPad, the functionality of the associated .dsz score file, and the evolution of the Decibel Score Creator desktop application that have been driven by recent user experience. This includes the introduction of an annotation layer, so that users may 'write' directly on to the score via the iPad screen; new methods for transferring scores to the iPad application, including the prototyping of a score server using QR code functionality; the ability to have different audio parts within a single score file and the expansion of the Decibel Score Creator's capabilities. Each of these developments has been primarily driven by user feedback, and to a lesser extent the evolution of various operating system compatibilities. The changes have enabled the Decibel ScorePlayer to remain relevant, easy to use and a valuable tool for reading animated, graphic notation.

1. INTRODUCTION

The Decibel ScorePlayer is an iPad application designed to coordinate the reading of scores featuring predominantly graphic notation in rehearsal and performance. It features scrolling and 'tracking' modes for reading, and can receive drawing commands for real-time score generation [1]. Developed by the six peice Australian Decibel new music ensemble from 2012, the application is distributed on the Apple App store and is used by composers and performers worldwide. The project has recently been open sourced and has been made available on GitHub [2] under the GNU GPLv3. The Decibel Score Creator desktop application creates the unique score file format required for upload to the Decibel ScorePlayer application. These files can include multiple parts, audio, and instructions [1]. It is available for free download from the Decibel new music website [3].

2. USER FEEDBACK

Users play a dominant part in the innovation process of software development [4], and this has held true for software designed for the display of generative and animated notation. Recent developments in MaxScore, for example, have catered to user demand for features like proportional notation [5] and the ability to view notation for MIDI clips in Max for Live [6], as well as being driven by a desire to increase user-friendliness. In the Bach library, observations of the shortcomings that users faced in Max when trying to implement complex, generative algorithms led to the development of the bell (bach evaluation language for llll s) language, simplifying the process for users [7]. Likewise, the Decibel ScorePlayer and Score Creator have been developed in close collaboration with end users.

Initially, these were the performers and composers in the Decibel new music ensemble, but once the application became available internationally via the Apple App store, a broader demographic of users become engaged. These users sometimes contact Decibel members with feedback and requests. Decibel composers also use the applications in conjunction with other performances, gathering feedback in those experiences, and large scale projects often drive new ideas, as was the case in Hope's opera *Speechless* (2019) [8]. It has been demonstrated that high levels of innovation result when suggestions come from non technical users, that is, those without the specific programming knowledge used in the applications development [9]. These suggestions, however, need to be tempered against the practical, structural dimensions of the project and the ability to maintain it into the future. Given the large number of extant score files, created as far back as 2012, maintaining backwards compatibility has become an increasingly important task. To that end, when implementing new types of scores based on composer feedback, it remains important to consider how such features might fare into the future. Does the proposed code and the design allow sufficient flexibility given that these choices will likely need to be maintained long term? Sometimes choosing sensible defaults is sufficient to ensure this, while other times a more concerted design process is required to identify potential pitfalls that could hamper future programming directions. What follows is a discussion of a number of features in the Decibel ScorePlayer and its associated software that have been driven by user input.

3. NEW DEVELOPMENTS

3.1 Annotation Layer

The annotation layer was developed in direct response to performers expressing a need to make notes on scores in the Decibel ScorePlayer, as they would do with traditional, paper scores and with other tablet music reading applications, such as forScore. The feature first appeared in version 2.0.0. It enables users to activate an annotation layer by touching an ‘Annotate’ button in the top right hand corner, that becomes visible while the player is stopped or paused (Figure 1). This button reveals the option to write, or remove, annotations via pencil and eraser icons (Figure 2). The primary default activation setting is the pencil icon. Touching the pencil means annotations can be made on the section of the score displayed, in red, made using a finger (Figure 3) or the purpose built Apple Pencil that can be purchased for use with the iPad. Touching the eraser icon enables the user to erase annotations in a similar fashion (Figure 4). When activated, the ‘Annotate’ button changes to ‘Done’ and when this is touched, editing is disabled and the player returns to the normal score display mode. Changes are automatically saved to the user’s iPad when no touches have been detected for more than three seconds, when the user changes where they are in the score, or when the ‘Done’ button is pressed. The annotations made are specific to the individual iPad and are not propagated across the network. This enables each performer to make their own personal notes on the score (Figure 3). While annotating, a user cannot change the playback state of the score and the ‘Play’ and ‘Reset’ buttons at the bottom left of the screen (Figure 1) are greyed out (Figure 2).

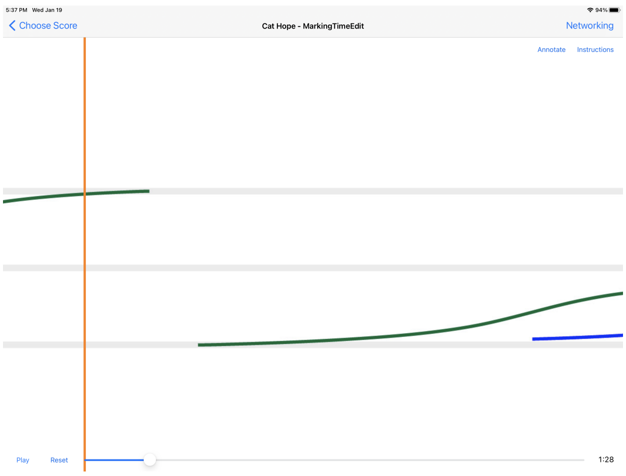


Figure 1. A screenshot showing the annotate button in the top right hand corner of the open score file on the iPad, in this case as it appears in a score by Cat Hope.

While this functionality seems relatively straight forward, implementing it was a complicated process that required a major change to the architecture of the application. In previous versions, there was a tight coupling between the user interface (UI) and the network messaging code that synchronised any connected iPads: a single module of code, PlayerViewController.m, was responsible for both. If the position of the score was changed on one iPad using the location scroll bar, the UI on all of the other connected iPads would instantly reflect this as soon as the network messages generated by the event were received and interpreted. While this is effective when simply displaying scores, it sets up a conflict when one user wants to write on a score while others can still control its position and playback. To resolve this, the code was split into two parts: the original module continued to control the UI while a new module,

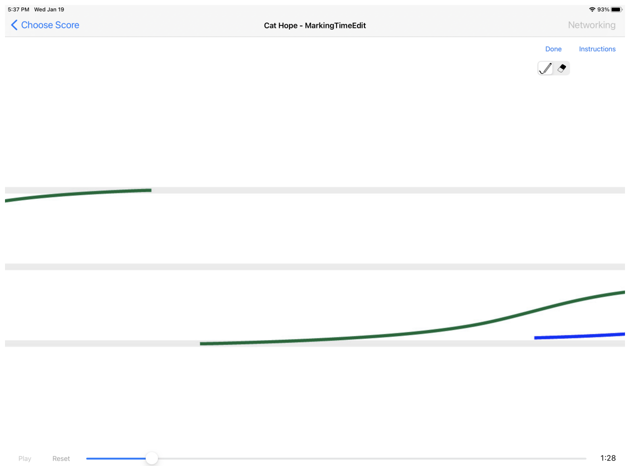


Figure 2. A screenshot showing the screen once the ‘annotate’ button has been touched. Note the greyed-out ‘erase’ icon at the top right, and the grey (instead of blue) ‘Play’ and ‘Reset’ buttons in the bottom left.

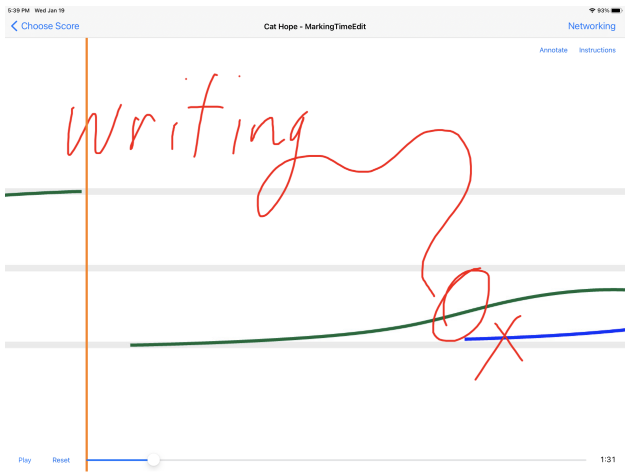


Figure 3. A screen shot showing some annotations made on the score in the ‘annotation’ layer, with the layer closed and in pause mode, ready to play or reset to the start.

PlayerCore.m, dealt with networking and the state of the ScorePlayer.

When a user enters the annotation mode these two modules are temporarily ‘detached’ from one another. The player core continues to receive network messages, storing the external state of the score’s location and playback, while the user can manipulate their local copy independently. When the annotation mode is exited using the ‘Done’ button the player synchronises with the other iPads, effectively catching up to what the rest of the ensemble has been doing. Another member of the ensemble, for instance, can start a score playing and the annotating user remains undisturbed from completing their markings until they have finished. The only case in which an interruption occurs is if another user triggers a change of score. In this case, any existing annotations are automatically saved immediately, the user is placed back in the standard viewing mode and the new score is loaded. While this isn’t an ideal situation, it is likely to be an edge case that doesn’t warrant the increased

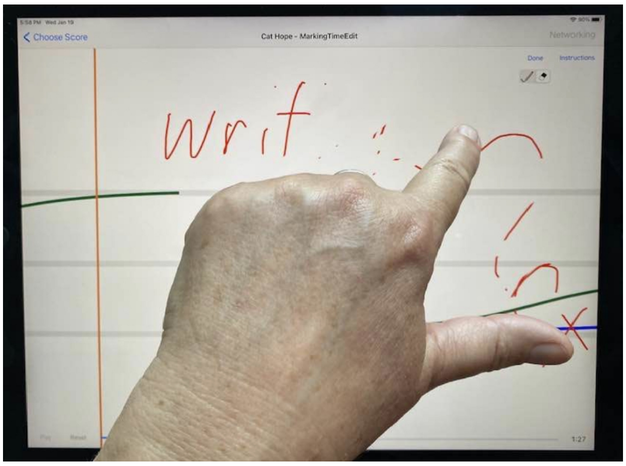


Figure 4. A photograph of a user in ‘erase’ mode using their finger on the iPad screen. Note the pencil icon is now greyed.

code complexity that would arise from delaying the load of the new score.

Another issue faced when implementing the annotation layer arises from the modular nature of the Decibel ScorePlayer. Different rendering modules control the drawing and animation of different types of scores, allowing wildly different score paradigms to share common UI and networking code [10]. Code that would work to handle the annotations on a scrolling score, the default format for the application, wouldn’t work for a score that consists of a series of timed flash cards, for instance. [1] And for some highly generative or non-linear scores, there is no real necessity for an annotation layer. To deal with this, the responsibility of managing the layout of annotations is given to the individual rendering modules. If a rendering module lacks the required methods to handle this then annotation is disabled and the ‘Annotate’ button is hidden from the user. A new module, PlayerCanvas.m, deals with the common task of accepting drawing input from the user. This module receives a snapshot of the currently visible section of annotations from the rendering module which can be used as a working copy. Changes are made to this image based on user input, and when a save is triggered the altered image is sent back to the rendering module to be incorporated into the saved annotations at the appropriate location.

3.2 Transferring Scores

When first launched on the App Store, the Decibel ScorePlayer shipped with four scores from composers within the ensemble. With the addition of Cat Hope’s *Her Pockets Full of Inertia* (2019) in version 2.0.2 it now has five [12]. This number provides a good balance between showcasing some of the features that the ScorePlayer offers to composers and performers, and keeping the download size of the app small enough that users aren’t dissuaded from installing it on their device. This work is the first score containing inbuilt audio to be shipped with the application. The real power of the Decibel ScorePlayer however comes from the ability to upload additional scores to the application, and this function is commonly used. In the past, this was done by connecting the iPad to a computer with a cable, and using the file sharing capability of iTunes software on either a Mac or Windows machine to transfer files [2]. This was clunky and not particularly intuitive, to the point that an instructional video was made by Decibel to deal with the high number of queries from users unfamiliar with the file sharing capacity of iTunes. Facility of upload has therefore been an important aspect sought by users and developer alike.

Bringing support for Apple’s AirDrop protocol to the ScorePlayer in version 1.10.3 addressed these difficulties, at least on MacOS. [3] This is a feature of Apple’s operating systems that is well known to users of iOS, being used for sharing in other music applications such as the aforementioned forScore and piaScore. The ScorePlayer is able to take advantage of this feature because of the separation that exists between the distribution and playback of scores, a convenience that is not afforded to platforms that rely on web based playback mechanisms, like MaxScore’s websocket mode [5] or SmartVox [13]. (Although this approach has its own, distinct advantages.) And because this is largely handled by the operating system, no additional code needed to be written to support it. Instead, the application simply needed to let the operating system know that it was capable of handling files with the extension .dsz by declaring this in its metadata [14]. Specifically, in the application’s property list file. Users can then choose the score in the Finder on their computer, click the ‘share’ icon, select AirDrop from the drop-down menu that appears, and choose the iPad with the Decibel ScorePlayer installed that they want to send it to. To date, this has proven to be the easiest and most reliable method of uploading the scores to the application.

A more platform neutral way of loading scores from a network was been added in version 1.9.6 in the form of a downloads window in the Score Management screen. Users can enter a URL to a score file hosted on a web server, and this would then be downloaded and would appear immediately in the application’s score selection screen. For added convenience, and to save the user from having to type in a long web address, a QR code hosted on the same web server or generated locally with freely available tools can be used to enter this. Apple’s AVFoundation framework is used to capture the camera feed from the iPad after the user grants the application permission to access it, in line with Apple’s privacy requirements. The framework includes the AVCaptureMetadataOutput object which can be attached to the camera’s capture session and will alert the application when a QR code is detected. The existence of these system libraries that can do the heavy lifting makes adding these sorts of conveniences to the App a very straightforward procedure which can be accomplished in relatively few lines of code.

[1] Flash cards is a Decibel ScorePlayer mode where different images appear on the screen in random or pre-determined order or timings. See works such as Cat Hope’s *Sub Arial* (2015) for an example of this format. [11]

[2] Use iTunes to share files between your computer and your iOS or iPadOS device: https://support.apple.com/en-au/HT201301

[3] Windows users are still required to use iTunes to deliver scores to the application.

3.3 Score Server

Using the download manager, with its QR code reader, as a platform agnostic means of distributing scores is convenient, but requires some initial IT setup. At the bare minimum it requires a functioning web server. Beyond that basic setup, there's a lot of scope to develop additional features on the server to manage the available collection. The Decibel Score Server is a prototype of such an environment (Figure 5). It uses a number of PHP scripts to allow users to upload and then access scores. The initial upload script modifies the uploaded score so that it contains a user supplied version number and a reference to the site in its metadata. That way, if changes are required to the score, the composer can upload a new revision with a higher version number and such updates can then be discovered from within the Decibel ScorePlayer's downloads window without the need to manually revisit the website [8]. Once the upload is completed, a QR code is also generated which can be used by the Decibel ScorePlayer to directly download the initial version of the file from the server. An additional script can be used to generate a table showing QR codes and direct download links for all of the scores that have already been uploaded.

While these scripts are still a work in progress, lacking any sort of security or authentication mechanisms, there are plans to eventually release them under the GNU Lesser General Public License [15]. Porting the code to a platform like node.js in the future would also allow for a more self-contained server solution [16] that wouldn't be reliant on a more traditional web server like the Apache server that the current prototype runs on [17]. This is similar in some regards to the approach taken by the SmartVox project [18], but is much more limited in scope: the node.js server would only be responsible for the distribution of the score and would not be involved in playback or synchronisation. And like other score distribution systems, if a web presence isn't required, the server could easily be packaged onto an embedded device like a Raspberry Pi to aid with deployment [13].

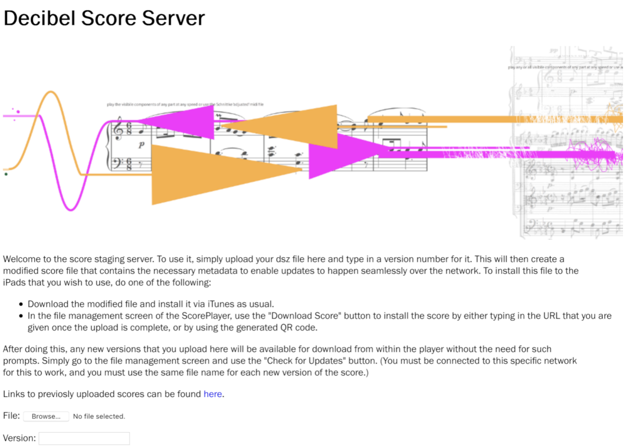


Figure 5. A screenshot of the Decibel Score Server prototype. At the bottom of the image you can see the html form elements that allow the user to select a score file and enter a version number for it.

3.4 Multiple Audio Uploads In a Single Score File

The .dsz score file for the Decibel ScorePlayer stores visual, motion, audio and other organisational information. The capacity for audio to be included in the score file is a long standing feature, having been introduced back in 2015 [19], and more recently featured in a score shipped with the application as noted above. However, some composers wanted the ability to ascribe unique audio to different parts within the score. This way, each iPad in an ensemble environment can generate it's own audio, linked to the part of that particular performer. While this has become available in the Decibel ScorePlayer with the latest release (version 2.0.6), workarounds have been used for a while. Using the Score Creator to create multiple different score files that contain exactly the same data but different audio files allowed this to be realised before it was an official feature, and the first score to use this workaround was Lindsay Vickery's work *Sanctuary* (2018) for percussion quartet [20]. The individual audio files were used to deliver unique click tracks to each of the performers, enabling them to deliver radically complex poly-temporal rhythms. As demand for this functionality has increased, work was undertaken to bring it officially into the Decibel ScorePlayer, culminating in the most recent release. The next step will be adding this to the Score Creator, so that the feature will be accessible to all composers.

As this feature was an evolution of existing features such as audio playback and multiple parts, rather than something entirely new, implementation was relatively straight forward and simply involved decoupling the code that loaded audio files from the code that initialised the audio playback object. That way, different audio files could be loaded as required, independent of the playback object's creation, which only needs to happen once. Notifying the audio player of the part change was a little more convoluted. Swiping up and down on the score has become the de facto standard when it comes to changing parts in the Decibel ScorePlayer, but in spite of this, it's still technically a mechanism that is dependent on the type of score. It was implemented in this way to allow different score types flexibility in the way that they reacted to user gestures. As a result, the UI code that passes the gestures to the rendering code has no knowledge of the connection between the swipe and a part change. To keep everything neatly compartmentalised, this means that the UI sends notification of the gesture to the current score renderer, and if that swipe corresponds to a part change the renderer notifies the UI in turn of this so that the UI can notify the audio player of the part change.

3.5 Decibel Score Creator

The Decibel Score Creator is the tool that converts images into the score files readable by the Decibel Score Player. For scrolling scores, the most common score type and the first to be implemented in the Score Creator, the image supplied by the user passes a playhead, which appears in its default form as a vertical, orange line. With the addition of an increasing number of score format options to the Score Creator, it is now possible for an image to be supplied as a 'playhead,' replacing the default vertical line.

Two works that provide different ways of using this function are Cat Hope's *The Rupture Exists* (2020) [21] and Bergrún Snæbjörnsdóttir's 'SOAP' (2020) [22]. Hope uses an image of coloured dots as points of performance in her work (Figure 6) and Snæbjörnsdóttir creates the illusion of a ring which score images pass through in her work (Figure 7). This feature enables experimentation with the concept of 'point of performance.'

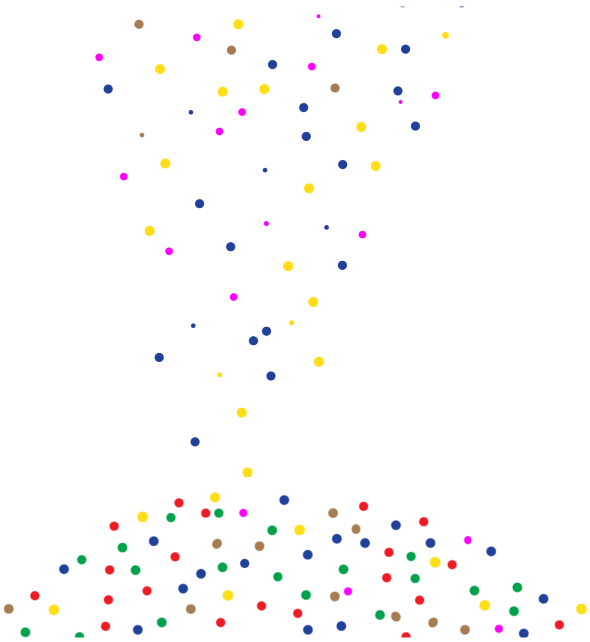


Figure 6. A screen shot from Cat Hope's *The Rupture Exists* score, showing a cloud of dots (at the bottom of the image) acting as a 'playhead' for the image descending from the top of the page.

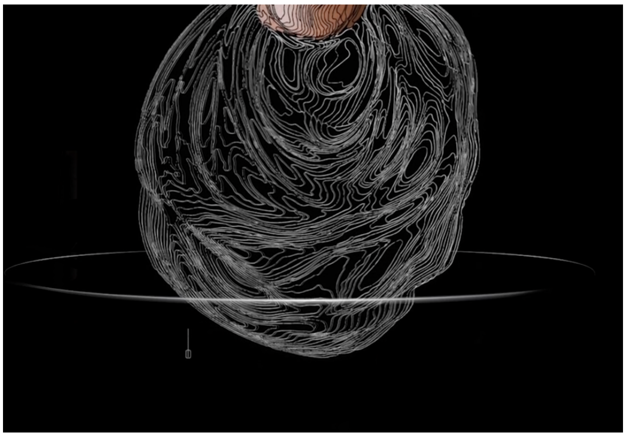


Figure 7. A screen shot from Bergrún Snæbjörnsdóttir's 'SOAP' score. The silver circle is actually the playhead, which the image appears to 'pass through' as if three dimensional.

The score creator also now enables users to create their own versions of the 'Talking Board', a score format developed for a work of the same name by Lindsay Vickery and Cat Hope [23]. This is a radically different approach to the horizontal or vertical reading of the score format. It is a roaming form where each player plays the contents of a circle that traverses the image [24]. Now, users can insert their own image to be read this way, and can define the number of circles that appear by default, currently set as up to 6, with a range of colours selectable. There are additional score types available within the Decibel ScorePlayer that are yet to be added to the Score Creator, offering potential to provide composers with an expanded array of possibilities. Before this work is undertaken though, there is a current push to port the existing features from Objective-C using Apple specific frameworks (Cocoa) [4] to C++ using Qt [5]. This will allow the Score Creator to run on Linux and Windows as well as on MacOS with one common codebase supporting all three platforms. This broader accessibility is something that has often been raised as a concern by composers without access to Apple hardware, and while the Decibel ScorePlayer remains tied to iOS, this is an important first step towards more neutral platform support.

4. CONCLUSIONS

As this paper demonstrates, user feedback from both within the Decibel ensemble and from composers and performers outside the group has continued to drive improvements in the Decibel ScorePlayer and Decibel Score Creator applications. Some of these, like the addition of different audio files for individual parts, have been small incremental changes that have built on established features, while others, such as the annotation layer, have been major additions. Further, user feedback has seen two improved methods for score file upload developed, and new possibilities for playhead design. In each case, the aim of such developments has been to respond to users feedback, and provide more choice in how the software can be used, providing greater access to a wider variety of creative options. These options have been alternately devised by composers and performers in the ensemble, users who download the application for their own use, and by the programmer of the application. With the transition of an increasing number of the project's components to an open source development model, it is hoped that this increased accessibility will result in more avenues for feedback between user and developer, allowing the application to continue to develop and improve.

Acknowledgments

All members of the Decibel new music ensemble have contributed to the ideas behind the developments outlined in this paper. Special thanks to the other members Lindsay Vickery, Louise Devenish, Stuart James and Tristen Parr. Also, to all Decibel ScorePlayer and Decibel Score Creator users around the world, for your feedback and ideas.

5. REFERENCES

[1] Decibel New Music Ensemble, "Decibel Score-

[4] What Is Cocoa?: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CocoaFundamentals/WhatIsCocoa/WhatIsCocoa.html

[5] Qt Framework - One framework to rule all!: https://www.qt.io/product/framework

Player - Decibel." [Online]. Available: https://decibelnewmusic.com/decibel-scoreplayer/

[2] A. Wyatt, "Decibel ScorePlayer," Jan. 2022, original-date: 2022-01-11T00:19:31Z. [Online]. Available: https://github.com/psiborg112/ScorePlayer

[3] Decibel New Music Ensemble, "Decibel ScoreCreator - Decibel." [Online]. Available: https://decibelnewmusic.com/decibel-scorecreator/

[4] C. A. Voss, "The role of users in the development of applications software," *Journal of Product Innovation Management*, vol. 2, no. 2, p. 113, 1985, publisher: Elsevier.

[5] G. Hajdu and N. Didkovsky, "MaxScore: recent developments," in *Proceedings of the International Conference on Technologies for Music Notation and Representation–TENOR*, vol. 18, Montréal, 2018, pp. 138–146.

[6] ——, "Maxscore: Current State of the Art." in *ICMC*, 2012, pp. 9–15.

[7] J. Bell, "Improvements in bach 0.8.1, a User's Perspective," in *SMC-Sound and Music Computing*, Toronto, 2020, pp. 18–24.

[8] C. Hope, A. Wyatt, and D. Thorpe, "Scoring an animated notation opera–the decibel score player and the role of the digital copyist in'speechless'," in *Proceedings of the International Conference on Technologies for Music Notation and Representation–TENOR*, vol. 18, Montréal, 2018, pp. 193–200.

[9] C. A. Voss, "The role of users in the development of applications software," *Journal of Product Innovation Management*, vol. 2, no. 2, pp. 116–117, 1985, publisher: Elsevier.

[10] A. Wyatt, C. Hope, L. Vickery, and S. James, "Animated Music Notation on the iPad," in *ICMC*, Perth, 2013, pp. 201–207.

[11] C. Hope, "Sub Aerial," Aug. 2019. [Online]. Available: https://doi.org/10.26180/5cda650b7044b

[12] ——, "Her Pockets Full of Inertia," Sep. 2019, publisher: Monash University Type: dataset. [Online]. Available: https://doi.org/10.26180/5cedc3203741b

[13] J. Bell, D. Fober, D. Fígols-Cuevas, and P. Garcia-Velasquez, "The BabelBox: an Embedded system for Score Distribution on Raspberry Pi with INScore, SmartVox and BabelScores," in *International Symposium on Computer Music Multidisciplinary Research*. Marseille: Springer, 2019, pp. 345–359.

[14] Apple, "Declaring New Uniform Type Identifiers." [Online]. Available: https://developer.apple.com/library/archive/documentation/FileManagement/Conceptual/understanding_utis/understand_utis_declare/understand_utis_declare.html

[15] "LGPL - MDN Web Docs Glossary: Definitions of Web-related terms | MDN." [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/LGPL

[16] Lena, "Deploy your Node.js application as a single executable," Jul. 2020. [Online]. Available: https://medium.com/@tech_girl/deploy-your-node-js-application-as-a-single-executable-4103a2508dd7

[17] Apache Software Foundation, "About the Apache HTTP Server Project - The Apache HTTP Server Project." [Online]. Available: https://httpd.apache.org/ABOUT_APACHE.html

[18] J. Bell and B. Matuszewski, "SMARTVOX-A Web-Based Distributed Media Player as Notation Tool For Choral Practices," in *TENOR 2017*, A Coruña, 2017, pp. 99–104.

[19] C. Hope, A. Wyatt, and L. Vickery, "The Decibel Score-Player: New developments and improved functionality," in *ICMC*, 2015.

[20] L. Vickery, "Sanctuary [2018] for percussion quartet, by Lindsay Vickery," 2018. [Online]. Available: https://lindsayvickery.bandcamp.com/track/sanctuary-2018-for-percussion-quartet

[21] C. Hope, "The Rupture Exists," Sep. 2020. [Online]. Available: https://doi.org/10.26180/5f741b3e4c4c6

[22] B. Snæbjörnsdóttir, "SOAP for flute, clarinet, viola, cello, prepared piano and percussion." 2020. [Online]. Available: https://vimeo.com/466824665

[23] L. Vickery and C. Hope, "The Talking Board," Aug. 2019. [Online]. Available: https://doi.org/10.26180/5ceb41e023365

[24] C. Hope and L. R. Vickery Dr, "The decibel scoreplayer-a digital tool for reading graphic notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*, Paris, 2015, pp. 59–70.

ON NOTATIONAL SPACES IN INTERACTIVE MUSIC

Vincent-Raphaël Carinola
ESM-BFC (Dijon), Université Jean Monnet (Saint-Étienne) France
vr.carinola@gmail.com

Jean Geoffroy
LiSiLoG, CNSMDL
Lyon, France
jean@lisilog.com

ABSTRACT

This article presents a reflection on the nature of notational spaces in interactive musical works using digital devices. It builds on the author's experiences in *Toucher*[1] (2009) for theremin and computer and *Virtual Rhizome*[2] (2018) for Smart Hand Computers[3].

In interactive music, notational spaces are correlated to the spatial structure of the *dispositif*[4], a notion that must be understood in the sense of an extension of the traditional instrument. That's why composing a work is equivalent, at least in part, to composing the instrument. The notational spaces — in other words: the places making possible a writing, and thus a musical interpretation — are distributed among the different components of the *dispositif*. The way in which its digital devices are interconnected (the mapping), the algorithmic logic of the "if-then-else" and the notion of openness play a fundamental role for the composer and the performer.

However, in the case of miniaturized (or embedded, or embodied) *dispositifs*, this spatial structuring of its components seems to be absent and, consequently, questions the existence of a place for composition and interpretation. One of the solutions explored here is to conceive the work as a virtual architecture that recalls a "world" in the field of video games. This architecture, open to a plurality of courses, then assumes the function of a notational space by calling, paradoxically, on techniques of memory specific to orality.

1. WHAT IS THE PURPOSE OF NOTATION?

Since its invention at the end of the Middle Ages, Western musical notation has fulfilled three important functions that should be distinguished:

(1) It offers the composer a "field of operations" making it possible to relate certain types of objects (the signs of notes, chords, rhythmic or intervallic structures) and the hierarchy of ideas. In so doing, it constitutes a sort of heuristic map, or navigation map, allowing thoughts to move. The score is like a workbench where

(2) the composer manipulates symbolic objects [1] bearing musical values and with which he carries out rational operations. This operative function was particularly important for the composers of the Second Viennese School and their post-war successors, and it can be found today in musical formalisms [2][3] influenced by the work and writings of Xenakis.

(3) At the same time, notation is the support in which the results of these operations are fixed, memorized. This means that it possesses a physical consistency (as a paper surface) and a spatial configuration (the staves), where the information is recorded, which will then allow the work to be projected in time. The importance of this function is evident in music known as "music for support" or for "fixed sounds" [4], but it can easily be extended to software in the case of digital works or "computer music" [5].

(4) For this information to be expressed acoustically, it must also concern the way in which the instruments will be activated, or at least describe the conditions necessary to produce musical sounds. From this point of view, it is possible to affirm that notation has a prescriptive function: it is an instruction manual or a user's guide. One manifestation of this function is the tablature. Variations of it can be found in the indication of the gestures that the musician must make on his instrument in contemporary scores (cf. Lachenmann's scores for strings).

1.1 Operativity of the notation

In fact, the purely symbolic, referential function of musical signs, that is to say their capacity to aim at an external reality, independent of them and which they represent,

[1] https://www.vrcarinola.com/toucher

[2] https://www.vrcarinola.com/virtual-rhizome

[3] The conception of the Smart Hand Computer is due to Christophe Lebreton. It describes and generalizes one of the important characteristics of everyday tools, such as smartphones, which combine a gesture capture interface and a computer in the same object. https://www.lisilog.com/shc/#:~:text=Cr%C3%A9ation%20d'une%20application%20Smart,instrument%20stable%20pour%20l'in-terpr%C3%A8te.

[4] We keep the French term of *dispositif*, which contains a polysemy (*device*, *apparatus*, *machine*, *plan*) and an etymology important for us (cf. the proximity of *dis-ponere*, which means "to arrange", to distribute objects in the space, and *cum-ponere*, that to give "to compose", to put in relation objects between them).

appears as secondary to the capacity of notation to be at the same time (a) the support of logical operations allowing the production of new musical forms and (b) a manual of instructions intended to transform a spatial object (a "score") into a temporal object (an "interpretation"), instructing sometimes on what to do (the tablature, the fingerings) and sometimes on what to hear (the note, the intended pitch). This distinction between the representative function of musical signs and their operative function is essential to understand the role of notation in the new dialectic between composer and performer that appeared during the 20th century, as witnessed, among others, by the open forms [7], the graphic notations [8] or the lexicons, sometimes very extensive, describing the extended techniques of numerous contemporary productions [9]. We will see that this notion of openness is essential in the conception of interactive works. Let us add for the moment that in these works, the three functions of notation that we have just described are assumed by the different technical objects that compose the *dispositif.*

1.2 Composing the instrument

We must consider the notion of digital *dispositif* as an extension of the *instrument*. This is because, unlike traditional configurations, in which the score is clearly distinguished from the instrument, and where the link between gesture and sound obeys deterministic laws that organology studies, in interactive works the conception and construction of the *dispositif* is an essential part of the elaboration of the work, and therefore of the writing, because of the con-tingent relation existing between the interfaces, the synthesis devices, the samples, the spatialization system and the different technical objects used. The notation is then distributed among the various components of the *dispositif* which thus assume the function of support to a writing, their arrangement being able then to be assimilated to a composition. For this reason, understanding the mechanisms by which the different components of the *dispositif* are arranged is essential to situate the places and the different functions of the notation. Let's see this in the example of *Toucher*.

2. NOTATIONAL SPACES IN *TOUCHER*

Toucher's[5] *dispositif* consists of a theremin, a pedal, a microphone, a computer and a set of speakers, the sound being diffused on six channels **(**Figure 1**).**

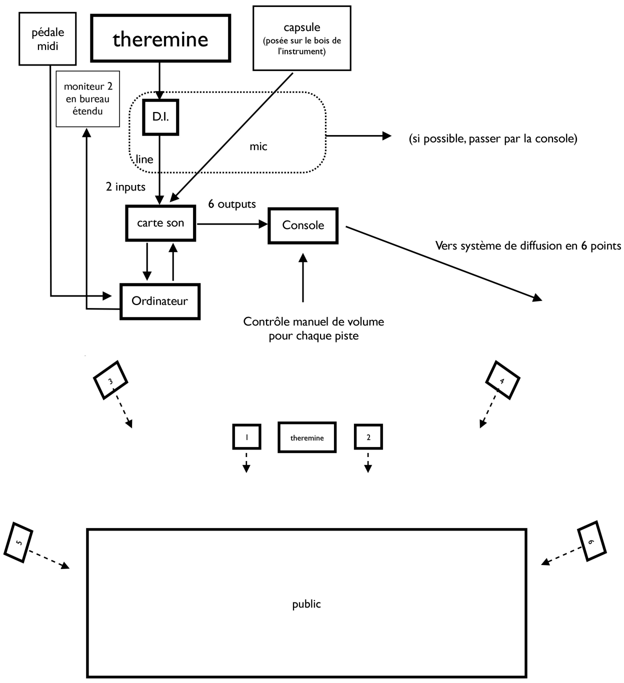


Figure 1. Dispositif's diagram of *Toucher*.

The audio output of the theremin is connected to the computer, which transforms the pitch and intensity data measured in the received signal into control data for the various sound processing modules used in the work. The theremin is thus a part of the *dispositif* that has a gestural interface function, the audio signal variations being the transduction of the musician's action, in other words, their imprint. The electronic sound of the theremin is only directly audible in two strategic places - from a formal point of view - in the work.

2.1 Mapping

An important part of the process of writing consisted in structuring the performer's playing space around the theremin's antennae, by cutting it into different concentric zones. When the hand evolves in one or the other of the zones, crossing them from one to the other, when it does so by approaching or moving away from the antennae, when it modifies its speed of displacement or when it combines this movement with that of the other hand, each of these movements thus differentiated produce a specific sound result and contribute to the definition of a gestural syntax which, with its sonorous equivalent, will condition the structuring of the musical discourse.

Let us note that it is the characteristic of any instrument to offer a structured space (a keyboard, a neck, a pipe), in other words an interface, imprinting its logic on the gesture of the instrumentalist [10]. Even in the absence of a physical object, the existence of this space, described in software form, is essential to give meaning to the musician's playing. Thus, in *Light Music* (2004) by Thierry de Mey

[5] *Toucher*, for theremin and computer, was premiered by Claudio Bettinelli on August 13, 2009 in Les Échelles (73), during the Festival Les Nuits d'été.

(1956), there is a close link between the structuration of the "wall of light", which one could assimilate to a virtual matrix, its evolution and the form of the piece [11][12]. This link is all the clearer that the structuring of the space gives to see in *Light Music*, very concretely, a calligraphy projected on the screen behind the musician which reminds the first cheironomic notations. The relation between instrumental *dispositif* and the support of a notation is then, literally, evident.

In the case of *Toucher*, the structuring of the space around the antennae is closely correlated to the functioning of the different audio processing modules contained in the software made with the computer software environment Max. We have seen that the control of the parameters of these different modules comes from the data of the analysis of the audio signal produced by the theremin. The connection between the interface and the parameters to control it was the second step of the process of writing. It was then a question of defining the "mapping", a term which refers directly to the function of cartography or navigation map and to the meaning of the notation. It is this structuration that will give each of the musician's gestures its audible equivalent.

The mapping evolves throughout the piece and thus participates in its formal conception. It is a privileged space of the writing, in which meet the structuration of the musician's playing space, the software description of the processing modules, and the algorithms determining the forms of interaction between them.

2.2 Graphic score

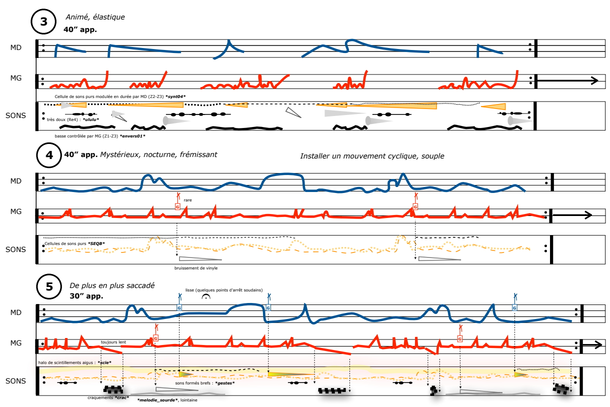


Figure 2. Excerpts from the score of *Toucher*.

The notation used in the graphic score (Figure 2) shows in part this double aspect: the two upper staves indicate the global movement that the hand must make between the different zones, represented by the lines of the staff; the lower staff represents the sounds produced.

However, this score containing the musician's gestures and a graphic representation of the sounds correlated to them, is only a part of the totality of the "score" that the musician will have to interpret. A part of it escapes the graphic notation and is contained in the software.

2.3 Automaticity of *dispositifs*

It is important here to emphasize a characteristic of digital *dispositifs*: their ability to function automatically, following the instructions contained in the software. This means that the performer plays an instrument that, at least in part, functions independently from his or her action.

This has two important consequences. The first is that, unlike with traditional instruments , the musical sound is not always the product of the performer's gesture or intentionality. He extracts from the depths of the algorithm a sound material "already there" in virtual form, which he sculpts while keeping a listening presence at each moment of his performance, in a hand/ear correlation of great requirement. The second is that the musician must then know this space of notation, invisible during the performance, encoded in the computer program[6], because it is in this program that the logical characteristics of the functioning of the instrument are contained, in particular in these two components: (a) the information fixed in the program, concerning for example the samples or the evolutions of the parameters of synthesis, which make the instrument a robot, and (b) the modalities of interaction emanating from the external data which characterize the space of play of the interpreter.

From then on, the sound material given to be heard is the simultaneous product of the musician's action *and* the software computation. The musical sound, one could say, results from a meeting between the logic of bodies and the logic of algorithms [13]. For this reason, the graphical score remains in *Toucher*, as in *Virtual Rhizome*, relatively open. It instructs on the type of movement that the musician must perform and on what must be heard, but it is also coupled with an indication of character (*Mysterious*, *Nocturnal*, *Oriental*, *Saccadic*, etc.) whose primary function is to induce the intentionality necessary for any form of musical expression, correlated here with the exploration in time of a sound content that is already there, present in the functioning of the software[7].

3. NOTATIONAL SPACES IN *VIRTUAL RHIZOME*

The possibility of an interaction between the performer and a notational space escaping from a symbolic representation was the object of a singular reflection in *Virtual Rhizome*[8] because of the original nature of the *dispositif*.

[6] Except in the case of Live Coding, where the notational space, that is the computer code, becomes not only visible but is the very object of the performance.

[7] This aspect is perfectly assumed in the script proposed at the very end of the score of *Toucher*, which describes in a few lines the functioning of each sequence of the piece.

[8] *Virtual Rhizome*, for a performer and Smart Hand System was premiered on March 3, 2018, at the Auditorium-ONL in Lyon by Jean Geoffroy as part of the Biennale Musiques en Scène. *Virtual Rhizome* benefited in 2018 from an Aid to the writing of an original musical work from the French Ministry of Culture. It responds to a proposal by Christophe Lebreton and Jean Geoffroy to compose a work for the original Smart

3.1 Dispositif *apparatus* and dispositif *device*

The visual artist Samuel Bianchini distinguishes two types of interactive *dispositifs*: *apparatus* and *device* [14]. Extrapolating to the musical domain: the first refers to *dispositifs* external to the performer, occupying a physical space with which the latter interacts, as we have seen in *Toucher*. In contrast, the *dispositifs devices*, to which *Virtual Rhizome* belongs, are closer to the embedded systems.

The miniaturization of the device poses an interesting problem for the composer and for the interpreter because there is no more any physical space to be structured and, consequently, there is no place for the elaboration of a cartography (mapping), from where a notation can be relevant. Consequently, there is no place allowing the deployment of a musical discourse and the expression of a legible intentionality clearly translated in audible form. The spatial support which, as in any instrument, constrains the musician's gesture, charges it with tension, practically no longer exists. The danger is then that the interface becomes a sort of primitive rattle, capable of producing such an infinity of sounds that the audience is left with a profound feeling of arbitrariness and formlessness.

3.2 Memory palace[9]

The solution adopted here was to introduce a technique that takes its source in orality. The formal structure of *Virtual Rhizome* can be compared to a "memory palace" [15] [6] that the interpreter goes through according to a certain number of constraints to which we will return. In other words, the interface, contained here in the smartphones, fulfils the function of a rudder that the interpreter handles to navigate inside a virtual architecture. It is virtual in the sense that it is comparable to a "world" in a video game whose images would be totally internalized by the musician and would have no other appearance than sonorous. In *Toucher*, a certain instrumental virtuosity was still readable in the musician's gestures thanks to the existence of a physical instrumental space. Unlike, the *dispositif device* of *Virtual Rhizome* produces an almost total internalization of the performer's action. The performer is constantly listening to the state of the system, the slightest movement of the hands being able to produce changes in the timbre, figures, emergences which are simultaneously the object of his contemplation. The memory of the system, the intuition, the intimate memory of the musician, are intertwined and embodied in the performance.

The score made available to him contains two types of information: the precise description of the functioning of the *dispositif* and the graphic statement of a possible conduct which is, in fact, only an example of projection in time of the spatial form of the work.

3.3 Operation of the Virtual Rhizome's *dispositif*

The piece is structured in twenty-three *situations* which correspond to as many states or configurations of the system, giving rise to as many musical sequences that can be combined with each other. Each *situation* is described in terms of the type of motion capture, the audio processing modules, the samples used and the control parameters. The *situations* are isolated from each other, like Leibnizian monads, "without doors or windows". This was one of the constraints linked to the technology, which did not allow the stacking of different processing modules. A musician's gesture of the lateral impact type allows to pass to the next *situation*, another one to return to the previous. The performer plays with two smartphones whose outputs are directly addressed to two stereo pairs forming a quadraphonic. Each smartphone has the same application - but they do not communicate with each other. It is the performer who decides on the succession and combination of *situations* according to the model proposed by the score, in a sort of two-voice counterpoint that gives rise to a multiplicity of possible paths.

In the example of Figure 3 (H), the musician alternates between *situations* 7 and 8 with the two smartphones. He can go forward, then backward, then forward again, obtaining each time new sound combinations and always renewing the musical discourse.

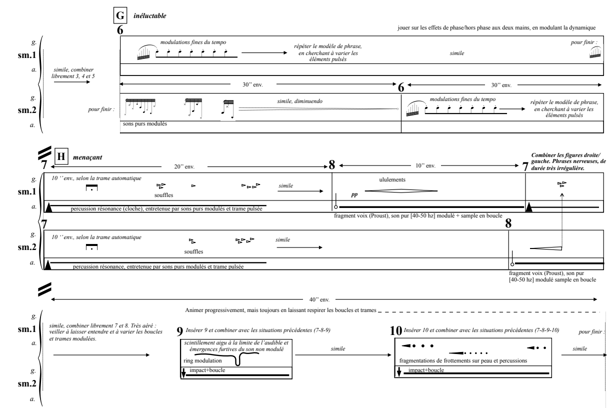


Figure 3. Excerpt from the score of *Virtual Rhizome*.

Each *situation* is composed of three elements:

(1) An impact-type articulatory element triggered when the situation is entered. These are samples of the same family taken from a repertoire specific to certain *situations*.

(2) An interactive module whose parameters are controlled by the musician's gesture. For example, a sound file transformed by a granulator whose volume, grain size or playback pointer position is controlled.

Hand Computer system, developed at GRAME-CNCM from the FAUST language and allowing to play smartphones as a musical instrument.

[9] The Memory Palace refers to the "method of loci" (*loci* being Latin for "places") is a strategy of memory enhancement which uses visualizations of familiar spatial environments in order to enhance the recall of information. This method is a mnemonic device adopted in ancient Roman and Greek rhetorical treatises. See https://en.wikipedia.org/wiki/Method_of_loci

(3) Loops of samples or frames of evolving synthetics generated live from various automated stochastic controls.

Elements (1) and (3) are characterized by an amount of indetermination that fulfills a triple function: to create a listening tension on the part of the performer, on the lookout for unexpected events; to produce a rhythmic environment that induces tempo, phrasing and conduct in the interaction described in (2); to renew each recurrence of the same *situation*.

3.4 Openness and “if-then-else” model

The automatic behavior of these indeterminate elements constitutes in fact an important reference for the musician's play, a sort of background through which the figures he draws take shape. This background and these figures, if they were fixed and predictable, would transform the performer into an operator whose main role would be reduced to the activation of a system thus entirely determined. This is because, unlike instrumental or mixed works, in digital devices the interaction between the musician's gesture and the perceived sound can only be described in an algorithmic form. This means that this interaction is governed by what can be called the “if-then-else” logic model specific to computer languages.

Now, so that the interpreter will not be reduced to one of the terms of the algorithm fixing the modalities of the interaction, so that this logic will not impose itself on him, it is necessary to create a space of interaction which escapes the determinism of the “if-then-else”. This, theoretically, is not possible in the case of digital devices, except by using a complexity that makes their behavior unpredictable, in whole or in part —for example, by means of stochastic functions— thus creating the illusion of a certain capacity for initiative by the computer. This is the solution adopted in *Virtual Rhizome*. This complexity associated with the automaticity of the device requires from the musician an attentive listening and an ability to react “in real time”, at each moment, as a gamer does when he is confronted with unexpected situations. It is a process that generates constraints that push the performer to make choices. Therefore, it constitutes a notational space (because of its prescriptive function) at the same time as the equivalent of the force feedback characteristic of traditional instruments, fundamental element for expressiveness.

The relationship between the performer and the work is in some ways close the one existing in open works. The instrument produces a background in perpetual evolution to which the performer reacts according to the constraints inherent to the musical material that himself generates and which is partly unpredictable, inside a formal architecture which, in its software fixation, lends itself to an infinity of parcourses.

3.5 Graphic statement, trace, direction, script

It is thus in the conception of the automatic functioning of the *Virtual Rhizome*’s *dispositif* that the possibility of an expression and a musical discourse is considered, in the sense that expression and discourse are the emanation of the singular intentionality of the performer. To accompany and orient the direction of this discourse, to the division of the work into twenty-three *situations* comes to be added a second division expressing by evocative terms indications of play, thus inviting the interpreter to work out his own script: *threatening*, *ghostly*, *ineluctable*, etc.

It should be noted that the graphic score is proposed as a model of form among others, resulting from the proposal of the first performer and co-author of this article, who participated directly in all the stages of elaboration of the work. Naturally, the meaning of “model” is not the same as that of the classical score containing the essential information for the performance of the work. The status of the notation here is close to that of a “trace” and a possible path.

3.6 Product, process and composition kit

In *Virtual Rhizome*, perhaps more than in *Toucher*, the performer must know the technical functioning of the device, here practically reduced to the dimensions of the two smartphones, while keeping in mind the global architecture of the work. This architecture is virtual in the sense that it does not manifest itself in a fixed temporal or symbolic form, but has in fact a content, a consistency, a certain logic induced by the nature of the sound samples, of the digital audio processing modules, by the software ordering of the *situations,* the different mappings or the expression indications. The whole of these contents, partly explained in the score and completed by other traces which are the video recordings of the various interpretations, constitutes in fact a kind of “composition kit” —another manifestation of a notational space— of which the interpreter appropriates to make the work emerge. From then on, this work is both an identifiable product (the “kit”, in its concrete components) and a work in progress [16][10].

4. CONCLUSION

The interactive *dispositifs* induce a dialectic between composer and performer that is reminiscent of open forms. This openness was most often reflected by a notation that sometimes questioned the linear representation of time, leaving the performer the choice of constructing his or her own path —for example in the André Boucourechliev’s (1925-1997) *Archipels* series—, and sometimes, by means of graphic notations, delegated to the performer a more or less important part of the definition of the musical material

[10] Apart from the versions recorded by Jean Geoffroy, two other interpretations, that of Martin Malatray (Lyon, CNSMD, June 25, 2019) and of Meng Fu (Moscow, Tchaikosvky National Conservatory, October 13, 2021) are now available. The function of the video recording has also been of great importance in the different interpretations of *Toucher*.

—a common approach in the John Cage's (1912-1992) works[11].

In interactive pieces, the notion of openness stems from the need for a space for interpretation that existed in symbolic notation and that must be found in works governed by the algorithmic logic of "if-then-else". The interpreter plays an instrument which is characterized by automatic processes which are independent from him actions. Without this openness, which implies a certain degree of indeterminacy in the interaction with the device, his role would be reduced to that of a simple machine operator.

Toucher and *Virtual Rhizome* provide two examples of the proximity between these new notational spaces specific to digital devices and the open works. But if in *Toucher* the notation still reflected the temporal form of the work's performance, in *Virtual Rhizome* it only exemplifies a possible path among an infinity. The notational space exists above all virtually, in the form of a labyrinthine memory palace, algorithmically conceived, that the performer roams at will and embodies.

5. BIBLIOGRAPHIE

[1] I. Xenakis, *Musiques formelles : nouveaux principes formels de composition musicale*, Paris, Stock, 1981.

[2] M. Andreatta, " Musique algorithmique ", in Donin Nicolas, Feneyrou Laurent (dir.), *Théories de la composition musicale au XXe siècle, vol. 1*, Lyon, Symétrie, 2013, p. 1239-1268.

[3] A. Riotte et M. Mesnage, *Formalismes et modèles musicaux : un recueil de textes (1963-1998)*, Paris, IRCAM-Centre Pompidou, Delatour France, 2006.

[4] M. Chion, *Musiques, médias et technologies : un exposé pour comprendre : un essai pour réfléchir*, Paris, Flammarion, 1994.

[5] J.-C. Risset, *Écrits, vol. 1*, Paris, Hermann, 2014.

[6] V.-R. Carinola, *Composition, technologies et nouveaux agencements des catégories musicales*, Saint-Etienne, Presses Universitaires de Saint-Étienne, 2022 (à paraître).

[7] U. Eco, *L'œuvre ouverte*, Paris, Seuil, 1979.

[8] J.-Y. Bosseur, *Du son au signe : Histoire de la notation musicale*, Paris, Éditions Alternatives, 2005.

[9] Ph. Lalitte, *Analyser l'interprétation de la musique du XXe siècle : Une analyse d'interprétations enregistrées des Dix pièces pour quintette à vent de György Ligeti*, Paris, Hermann, 2015.

[10] M. Chemillier, *Les mathématiques naturelles*, Paris, Odile Jacob, 2007, p. 131-158.

[11] V. Potapova, *Le geste, le mouvement et des nouvelles lutheries dans la musique contemporaine à travers* Light Music *de Thierry de Mey*, Mémoire de Master, Université Lumière Lyon 2, Master DPACI, 2016.

[12] J. Geoffroy, " Nouvelles interfaces et création ", in Revue Francophone d'Informatique et Musique, nº 6 – Techniques et méthodes innovantes pour l'enseignement de la musique et du traitement de signal [online], June 2018, <http://revues.mshparisnord.org/rfim/index.php?id=544>.

[13] E. Couchot, "Des changements dans la hiérarchie du sensible – Le retour du corps", in Borillo Mario, Sauvageot Anne (dir.), *Les cinq sens de la création. Art, technologie, sensorialité*, Seyssel, Champ Vallon, 1996, p. 127-131.

[14] B. Guelton Bernard, *Dispositifs artistiques et interactions situées*, Rennes, Presses universitaires de Rennes, 2016, p. 147-162.

[15] M. Carruthers, *The Craft of Thought: Meditation, Rhetoric and the Making of Images. 400–1200*. Cambridge, Cambridge University Press. 1998.

[16] N. Cook, "Between Process and Product: Music and/as Performance" [online], Music Theory Online, 7, april 2001, no 2, <http://www.mtosmt.org/issues/mto.01.7.2/mto.01.7.2.cook.html>.

[11] For example, in *Variations I* (1958) and *Variations II* (1961). The *Song Books* (1970) and the *Concert for piano and orchestra* (1958) compile numerous other examples of Cage's research on notation.

INTEGRATING MACHINE LEARNING WITH DSP FRAMEWORKS FOR TRANSCRIPTION & SYNTHESIS IN COMPUTER-AIDED COMPOSITION

Brandon Lincoln Snyder and Marlon Schumacher
Institut für Musikinformatik und Musikwissenschaft
der Hochschule für Musik Karlsruhe
Karlsruhe, Germany
brandon.snyder@stud.hmdk-stuttgart.de
schumacher@hfm-karlsruhe.de

ABSTRACT

In this paper we present applications integrating two classic machine learning methods into a Computer-aided Composition environment with the specific purpose of notating, organizing and synthesizing audio from large sets of sound data. We present a modular sample replacement engine driven by a classification method, and a texture synthesis application employing a clustering method. The applications are designed and presented with a particular focus on modularity and extensibility, with the goal of providing flexible options for integration into existing *OpenMusic* projects. Therefore, in addition to presenting the methodolgy behind our applications, we also highlight the modular aspects of their structure along with several functions for performing transient detection, Mel-frequency cepstrum analysis, and probability vector calculation.

1. INTRODUCTION

The development of computer software for compositional applications has a long tradition and today's computer-aided composition (CAC) environments provide users with powerful frameworks in which multiple types of media (instrumental, electronic parts, spatialization, gesture, etc.) can be created, represented and manipulated in an integrated fashion [1]. Machine learning (ML) methods open up a large number of possibilities for such environments, particularly because these methods can create multi-dimensional networks of relationships across large amounts of data [2]. In this paper we outline two applications for sound analysis, notation and synthesis driven by ML. The ML methods involved here are classic amongst current literature for ML-based audio applications, and are designed with clear access points for altering and extending them. We were particularly interested in these applications being modular. That is, for them to be easy to take-apart, build-up, and integrate into existing CAC projects. Built using a combination of the *OM-SoX* external library as well as custom functions within the *OpenMusic* environment, these applications are able to easily be incorporated into a user's larger CAC workflow.

In this paper, we first situate these applications within the wider context of machine learning-based and corpus-based synthesis and notation tools. Highlighting the modular aspects of the project, we show several custom functions that are implemented at different stages of the sound analysis, notation, and synthesis process. Then, we walk through the two applications, describing the techniques and algorithms involved, as well as show audio examples of the applications' output (available on the accompanying website). [1] Finally, we discuss the benefit, potential impact, and limitations of these applications, and speculate on further directions this project could take.

1.1 Related Works

Our interest in analyzing and querying large amounts of audio brought us near several methods in corpus-based concatenative analysis and synthesis. Tools such as the Caterpillar system [3], Audioguide [4], and OM-Pursuit [5] [2] are significant examples from this field, and the research behind those tools has informed our own decisions around what audio features, distance functions, and software to use. Though these aforementioned tools are effective and flexible, we have found in our experiences with students and professionals that they can be perceived as a bit of a 'black box' to artists with beginner to intermediate knowledge in these methods. The input and output of these tools are clear, while the internal algorithm is not immediately so. Thus, we sought to provide a modular toolbox within a popular visual composition environment, in a way that not only makes clear their internal processes, but also that is easy to adjust and integrate into existing *OpenMusic* projects.

[1] https://edu.marlonschumacher.de/audio-ml-4-cac/
[2] http://www.music.mcgill.ca/marlonschumacher/software-contributions/om-pursuit/

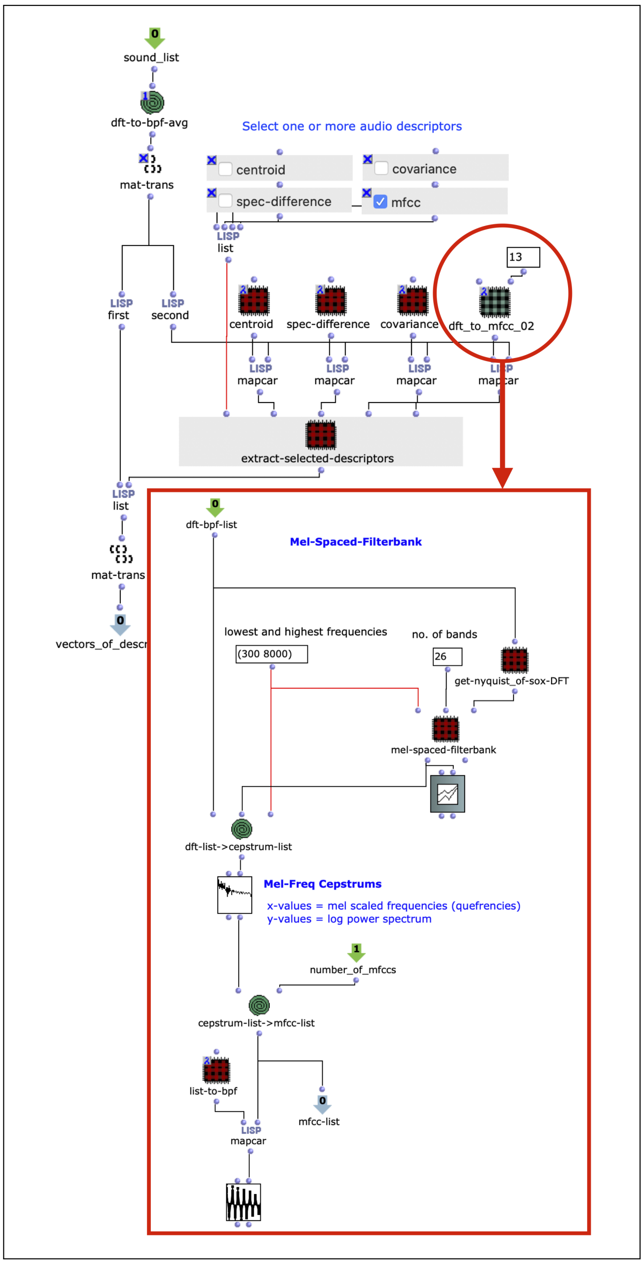


Figure 1: Interchangeable audio descriptors are set as patches in lambda mode. Here, a patch extracting 13 MFCCs is being used.

In terms of ML methods in CAC, our project shares similarities with OM-AI [6], an external library for *OM#* [3] that also seeks to provide tools to composers interested in ML. In order to maintain the modular nature of our applications, the ML algorithms and audio descriptor analysis functions are built as abstractions consisting of *OpenMusic* objects and simple custom LISP functions. Consequently, a user may change or even switch out these abstractions to suit their needs. Figure 1 shows our audio descriptor analysis engine with each descriptor as a patch in lambda mode. Featured in our application are Mel-frequency cepstrum coefficients (MFCCs), as well as spectral centroid, spectral difference, and covariance descriptors. Any number of these descriptors can be chained together to be incorporated into the ML process.

Likewise with our ML algorithms, Figure 2 shows a clustering algorithm in the form of an *OpenMusic* program, with the distance function implemented as a patch in lambda mode. This was an important structural decision to the project, as it not only gives a user a clear view of how the ML functions operate within the *OpenMusic* patch, it also allows for modularity within the actual ML algorithms. This would be useful for a user who might need to use a different distance function to evaluate their data, or who may need to pre-process or clean up their audio sets in some way before entering certain stage of the ML algorithm.

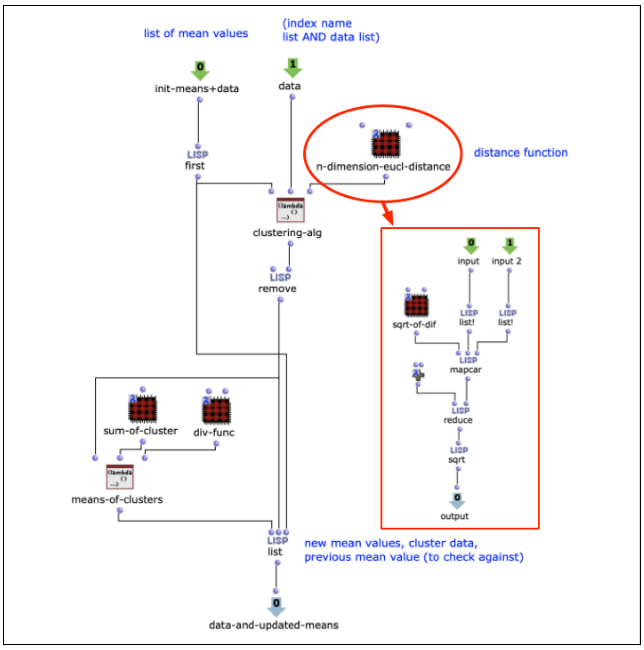


Figure 2: Interchangeable distance function. For certain audio sets, special functions such as a distance matrix or tree distance may be more suitable than Euclidean distance [3].

In order to provide the required DSP functionalities within *OpenMusic* and for flexible integration into existing workflows we employed *OM-SoX*, a framework for audio analysis and synthesis, bundled as an external library to be loaded into *OpenMusic*.

1.2 *OM-SoX*

OM-SoX is a programmable, modular analysis-synthesis framework for algorithmic audio processing in *OpenMusic*, developed by the second author. We used its analysis functions to derive audio descriptors, and detect and classify transients. We also used its processing functions to synthesize sound from that analysis data. While *OM-SoX* has been a popular tool used for audio processing [7], sound synthesis [4] and notation [5] (see Figure3), there so far

[3] https://github.com/cac-t-u-s/om-sharp

[4] http://www.music.mcgill.ca/marlonschumacher/arts/6-fragments-on-one-act-of-cleaning-a-piano

[5] http://notation.afim-asso.org/doku.php/evenements/2013-06-27-etude-notation1

seem to have been less notable applications for incorporating its analysis functionalities into compositional workflows. Included in our applications are functions that use *OM-SoX* to extract cepstrums, Mel-frequnecy cepstrum coefficients, spectral centroid, spectral difference, covariance, and detect transients. We hope this project might provide inspiration for possible musical applications of the analysis engine in *OM-SoX*.

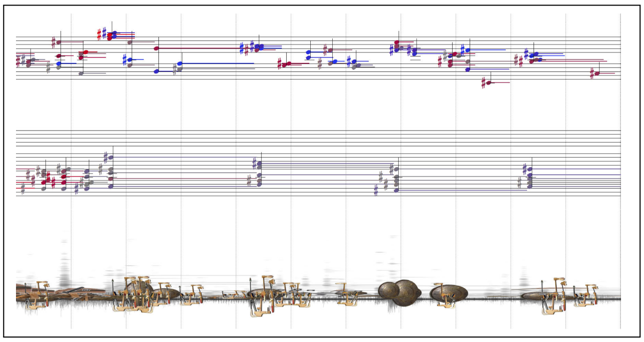


Figure 3: Figure shows the use of *OpenMusic* & *OM-SoX* for the creation of a mixed-paradigm notation in a *Maquette* object.

2. TWO MACHINE LEARNING APPLICATIONS FOR COMPUTER-AIDED COMPOSITION

With the overall scope of this project introduced, we will now present the two applications and the ML methods that drive them.

2.1 Modular Drum Sample Replacement

Our first technique uses a supervised classification method to create a modular sample replacement engine. In our example *OpenMusic* patch, we use three objects for mediating the sample replacement process: a *multi-seq* [6] notates the detection and classification of transients in the audio, a *BPF-library* notates the multi-sound-set sample replacement process, and a *maquette*-based sample sequencer transcribes new audio with the replacement samples.

The multi-seq, an object that notates data in the form of traditional western musical staves, sorts the segments of a given instrument-type (i.e. snare, kick, hi-hat) into its own staff (see Figure 4). This 'score' serves firstly as a transcription of the transients detected and classified by our *OM-SoX* functions. The score's second purpose is that of a script, dictating, in partnership with the BPF-library, the construction of a new sequence of samples.

In the BPF-library, each BPF corresponds to a transient which will be reconstructed from a pool of possible replacement samples. The specific replacement sample used is determined by a probability vector, which is sampled from the BPF.

In our accompanying audio examples there are three separate sample replacement libraries used: an acoustic

[6] https://support.ircam.fr/docs/om/om6-manual/co/Poly-Multi-Editor.html

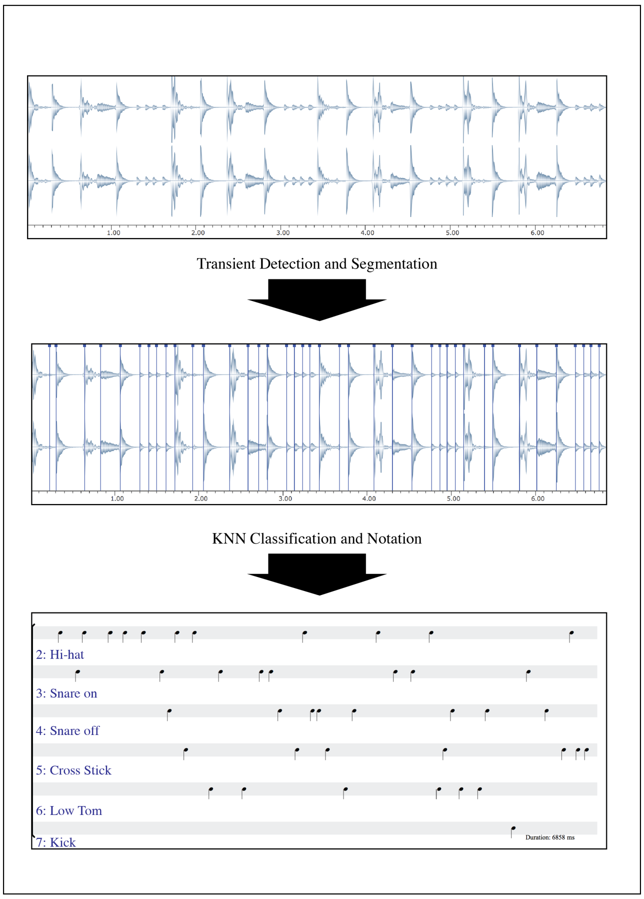


Figure 4: A sound is trimmed into individual segments, which are then classified and notated by instrument-type on a *multi-seq* object.

drumkit, an electronic drumkit, and a set of field recordings. Thus, each BPF called for each note from the multi-seq is sampled into three points. If the number of sample libraries were four, then four would be the sample-value of the BPF weighted vector. As illustrated in Figure5 the weighted vector is calculated from a three-point and four-point sampling. In the accompanying example audio, an interpolation between two BPFs is executed such that the beginning of the resynthesized drumloop consists exclusively of acoustic drum samples (sample replacement library 1), see Figure 6 for the specific interpolation. Gradually, this balance shifts so that at the end of the loop, there is an equal likelihood that a note from the multi-seq would call the electric drumkit or the field recordings. (sample replacement libraries 2 and 3). This BPF interpolation is a simple technique employing a high level visual interface for controlling the complexity of large sets of audio samples.

The multi-seq and BPF interfaces then lead to a maquette-based sequencer (see Figure 7), where the classified audio is represented again, this time with the final replacement samples (designated by color). Here, the transients notated and classified in the multi-seq can now be handled as actual audio samples, and can controlled for even further sound synthesis.

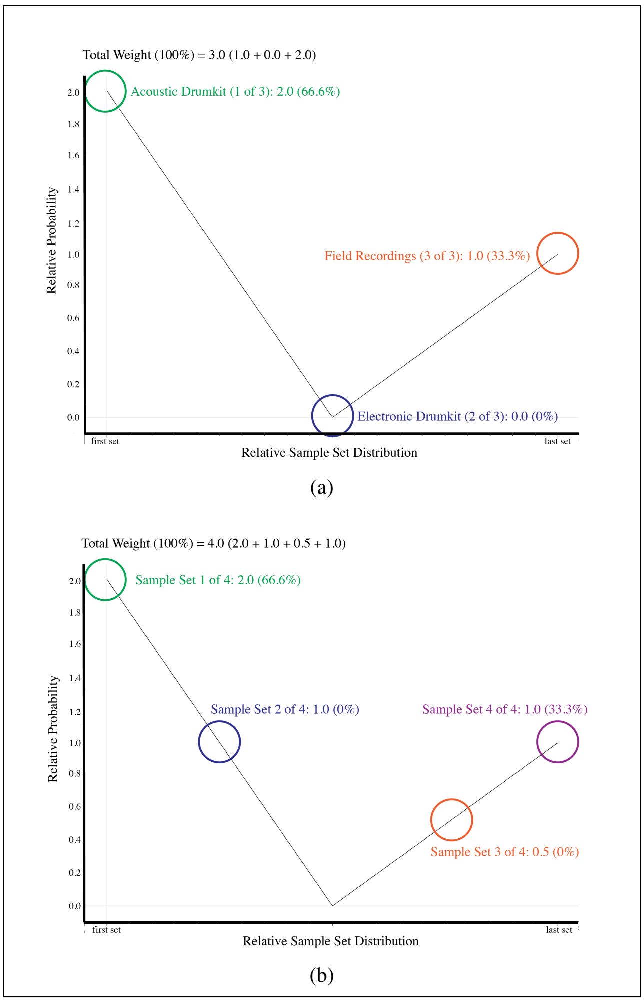


Figure 5: The x- and y-axis of the probability vector BPFs are in relative units. The probability values (y-axis) are relative to the total sum of all the indices values, and the distribution of proability values along the function is evenly distributed across the x-axis.

2.1.1 Preparing Audio for Classification

In this example, audio of live-performed drums is segmented into its individual attacks according to a transient detection method in *OM-SoX*. We track changes in amplitude between small grains to detect transients in the audio (see Figure 8. In our examples, we take the average amplitude of the audio. However, it is possible here to also detect transients along a specific band of the frequency spectrum.

Each segment is compared against a training set of already-classified drum hits using a k-nearest neighbors classification algorithm. The algorithm assigns each segment a class-ID, designating what drum or cymbal the segment contains. The class-ID of this segment sorts it into one of seven staves in a multi-seq (seven, per the number of class-IDs learned by the training set), altogether constructing a rough [7] transcription of the drum loop.

[7] The multi-seq in its current state notates only a class-ID, an onset time, and a duration for seven possible class-IDs. A higher level AMT tool would likely employ hundreds, accounting for dozens of different parameters including dynamics and timbre.

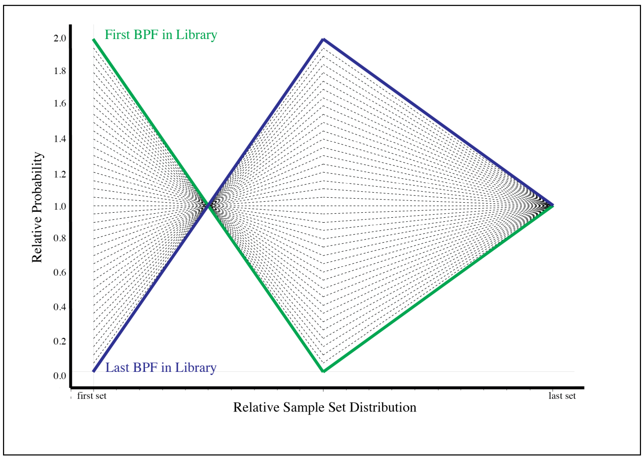


Figure 6: The gradual interpolation from the first BPF to the last is a simple technique that results in a complex mixture of many different sample libraries.

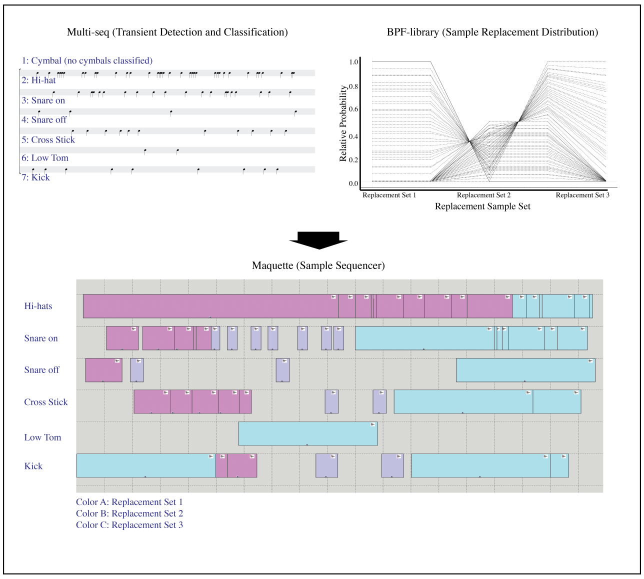


Figure 7: The maquette notates the results of the sample replacement process controlled by the multi-seq and BPF-library.

2.1.2 Classification Method

The method of transient classification used is a k-nearest neighbors algorithm. This is a classic supervised ML classification method, one that has been applied already for many years to audio applications. For readers who may be experienced in *OpenMusic* but unfamiliar with ML methods, this section (and the later section on k-means clustering) dives into the specifics of how this ML algorithm works.

First, a library of already-labeled drum samples has 'trained' the *OpenMusic* patch to classify a sound in one of seven different sound types (this is the 'training set'). Then, a segment of input audio (from the 'testing set') is analyzed (converted from audio signal to a vector of numbers) and compared against the data in the training set (see Figure 9). Here, the classification algorithm assigns a class-ID to the testing set vector according to

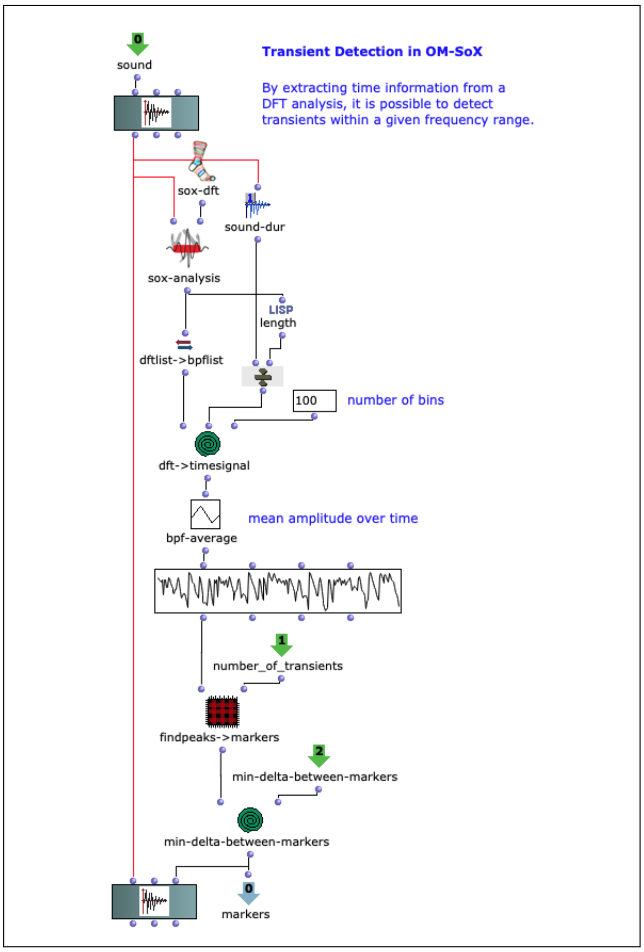


Figure 8: The amplitude is derived from a DFT analysis, making it possible to detect transients within specific frequency bands.

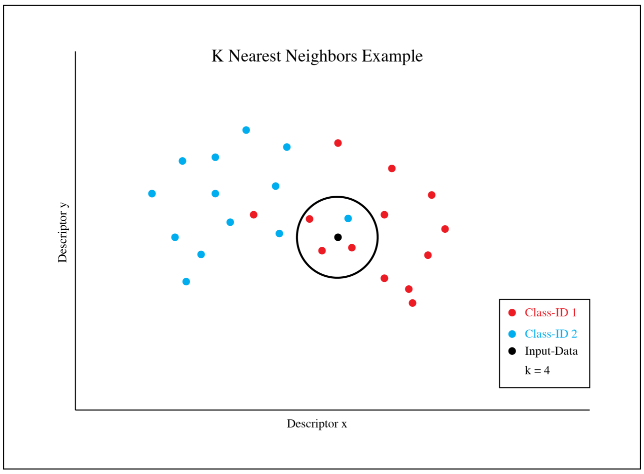


Figure 9: A data point is tested against a training set of labeled data to determine what class it belongs to.

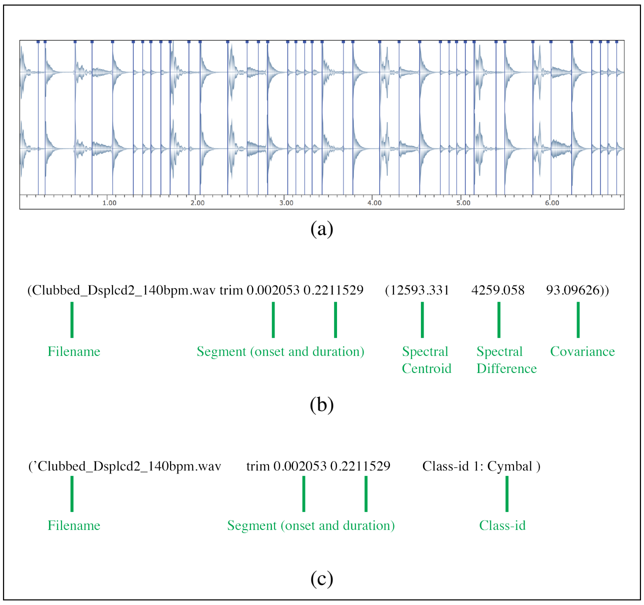


Figure 10: (a) The segmented sound before analysis. (b) The vector that references the sound segment. (c) The assigned class-ID after classifying the vector.

the class-ID's of the nearest vectors from the training set. This method is considered 'supervised' because there is a training set which the author of the algorithm has arbitrarily designated what class-IDs correspond to what data. An important corollary to this is that the act of classifying means that the algorithm is returning a discrete value (the class-ID. For this method there is no result that is "in-between" one sound type and another.

This algorithm does not handle any actual audio signal. Before any data is processed, it is first analyzed and assigned a number of values that describe it. These are called audio descriptors, and they are often derived from spectral information in the signal. In our project, we used the discrete Fourier transform (DFT) analysis to extract spectral centroid, spectral difference, covariance, and Mel-frequency cepstrums of the audio signal. Figure 10 compares a sound to its vector and eventual classification.

2.1.3 *Classification Training Phase*

We train our *OpenMusic* patch on a relatively small set of audio: 105 classified drum samples (fifteen samples of seven different drum sound-types), assigning MFCC data to each of them. The audio used in the testing set does not overlap with the training set. However, it does come from the same recording session. Having a common source instruments and microphones between training and testing sets is an important necessity when the training set is small. This helps ensure that the classification algorithm is not influenced by characteristics of the audio signal not relevant to the application. For example, the frequency response of a microphone, the spectral content of the background noise, and the specific timbre and tuning of the drums recorded all influence the audio's signal and consequential MFCC data.

Our decision to use a small training set aligns with our interest in keeping a low barrier of entry for users new to ML methods. While these applications also function with large training sets [8], such sets typically carry their own sets of concerns. Every data set brings different results to an

[8] such as MedleyDB, OpenMIC-2018, URBAN-SED, Urban Sound Datasets, for example.

algorithm, and thus it becomes especially important to interrogate the audio features each of these sets are curated around and annotated with [8].

2.1.4 Classification Testing Phase

Audio from the testing set is analyzed and converted to vectors of audio descriptor data in the same manner as the training set. The testing set data is processed with a k-nearest neighbors (knn) classification algorithm. In summary, a knn algorithm compares the distance between the given testing set vector and a number (k) of the nearest vectors from the training set. The effectiveness of a knn algorithm depends on how this distance is calculated.

We used an n-dimension euclidean distance function for our knn algorithm (as well as for our other, unsupervised clustering algorithm). Drawing from research on corpus-based concatenative synthesis [3], we determined that an n-dimension euclidean distance function was the most appropriate function for our specific goals of classifying and querying audio based on DFT analysis. Our *OpenMusic* patches include functions for extracting four audio features (MFCCs, spectral centroid, spectral difference, and covariance). However, it is possible for any number of audio descriptor functions to be incorporated. For example, any of *OM-SoX*'s 33 built-in audio descriptor analysis functions can also be used. Referencing classification methods used by Artemi-Maria Gioti [9], we used MFCCs in our examples because of their usefulness in classifying the timbre of audio.

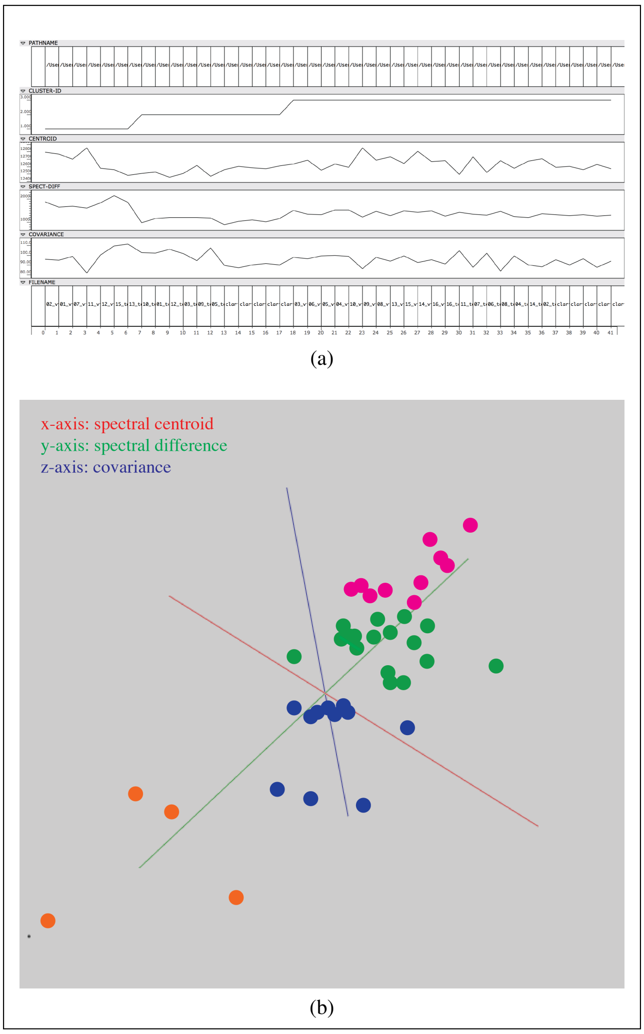
2.2 Texture Synthesis

Our second technique uses an unsupervised clustering method to synthesize sound textures from a large library of unclassified audio (we refer to texture synthesis as the reconstitution of certain sound characteristics via smaller preexisting sound elements, see e.g. [10] for an overview of different approaches). A sound library of any size is analyzed, clustering the individual sounds into groups based on one or more audio descriptors. These clusters are presented in a *class-array* and a *3DC-library*, two *OpenMusic* objects that can provide alternative representations of multiple parameters of a sound all at once (see Figure 11). This allows a user to view and organize large amount of data for texture synthesis. The synthesis process is executed through a collection of sorting and processing functions in which a user is capable of both orchestrating broad gestures involving many sounds, and arranging individual sounds.

2.2.1 Curating Sounds for Clustering

The results of this clustering method are influenced primarily by what audio is curated at the start. This method is 'unsupervised', meaning that the input of the audio is compared against itself. There is no external 'training-set' of data by which the processing of the input data is supervised. Rather, the clustering method analyzes a set of data and makes a network of connections between each sound.

This clustering method analyzes a three-dimensional vector of audio descriptors (spectral centroid, spectral difference, and covariance). We found that grouping multiple different descriptors such as these can lead to unexpected and interesting clusters. The idea here is that the sounds are not simply sorted along a single parameter (pitch, noise, harmonicity). But rather are grouped along a metric that combines multiple parameters.



(a)

(b)

Figure 11: (a) A *class-array* object illustrating the pathname, cluster-id, audio features, and segmentation information of the input audio library. (b) A 3DC-library visualizing the four clusters of sounds in a three-dimensional space.

We found that the most interesting results of this method often come when the designated number of cluster groups is different than the number sound-types in the input library. This "encourages" the algorithm to find similarities between sounds that are not from the same sound type. For example, with a 3-mean cluster method (i.e. the audio is clustered into three groups), an audio library consisting of cymbals, clarinet multiphonics, and field recordings clearly groups itself according to its sound type. Sound-type (or, timbre) has such a strong influence over our three audio features. However, this same three-mean cluster method, when applied to only clarinet multiphonics suggests for us an unorthodox, but interesting, way of grouping this relatively homogeneous set of sounds into three groups.

2.2.2 *Clustering Method*

The k-means clustering algorithm is a classic method for unsupervised machine learning. In our context, a clustering algorithm is used to group the input sounds. A 'mean' in this context refers to the mean value of a given cluster. For example, a 3-mean clustering algorithm will return 3 clusters (see Figure 12). The k-means clustering method is an iterative 4-step process in which 1) a number (k) of random 'means' are generated and each sound from our input is assigned to its nearest mean value. 2) This forms a number (k) of groups. The 'centroid' of a given group is calculated and that becomes a new mean value of that group. 3) step 1 is repeated with the new mean. The vectors are grouped anew, according to their distance from the closest of the new means. And then step 2 is done, evaluating the centroid of these new groups and generating a new set of k means from that. 4) After multiple iterations, the groups eventually settle into an unchanging set. Further iterations only produce the same clusters and means. This indicates that the process is complete.

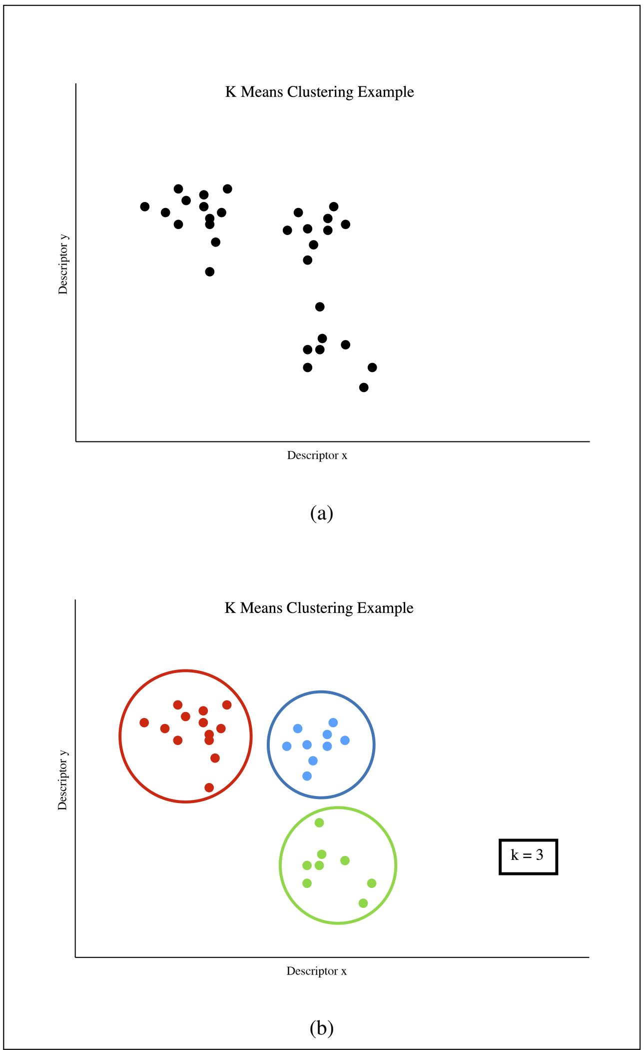


Figure 12: (a) A collection of unlabeled data (b) That same data clustered around three 'means'.

This algorithm may return different results each time it is executed, even when using the same input data. This is due to the random means generated in step 1. For our purpose of organizing sound against multiple parameters at once, we found it preferable to have an algorithm that provided varying outputs upon each return. Similar to using seeds in pseudo-random functions, this k-means clustering algorithm could potentially be controlled more tightly through the use of seeds for the randomly generated mean. Despite that, by representing and indexing the returned clusters in a 3DC-library and class-array object, a user is able to easily evaluate the clusters, allowing them to make adjustments to their input library and k-value, and iterate the clustering algorithm further until a desired grouping is found. Any need to save or replicate a particular output from the clustering algorithm is able to be saved by exporting the class-array as an OM-instance.

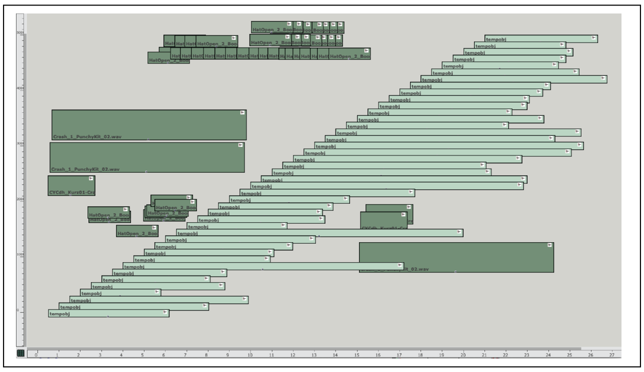


Figure 13: Texture synthesis with a *maquette* object. The cluster-sorted pitched-instrument sounds are the temp boxes in the maquette, while the sound boxes denote hand-placed cymbal samples. In this example, the y-axis does not influence the sound.

2.2.3 *Sorting and Calling Sounds for Synthesis*

Once clustered and stored in the class-array object, the sounds are selected and organized in a maquette, where they can be subsequently edited and synthesized. There are a number of functions for organizing audio in the maquette. Firstly, the order of the clustered sounds can be sorted by cluster-id, as well as by any other dimension in the class-array (e.g. covariance). Once sorted, any number of cluster groups can be called from the class-array to be placed in the maquette. The selected groups of sounds are then arranged in the maquette according to a list of onsets. Finally, it is also possible to hand-pick and arrange individual sounds in the texture, via a pop-up menu that indexes all the sounds in the patch. This is a useful tool for balancing out broad algorithmic gestures with micro, 'hand-written' ones. Figure 13 illustrates the accompanying example output audio: a collection of saxophone, clarinet, and violin sounds are sorted into three cluster groups. This creates a gradual nuanced harmonic texture that ebbs between noise and harmony. This homogeneous texture is then punctuated by individually placed cymbal sounds called by the pop-up menu.

3. CONCLUSIONS

Our hope is that these applications inspire and lower the barrier of entry for composers interested in incorporating ML methods into their own CAC workflows. Particularly, we hope that the modularity in these applications helps users develop new aesthetic approaches to ML methods. A variety of interfaces common in the *OpenMusic* environment were incorporated into the applications as forms of higher-level, complementary representations for controlling synthesis processes. These interfaces can be seen as a form of notation not only in a symbolic format (i.e. the multi-seq and maquette), the class-array can be understood as a tabulated notation, and the 3DC- and BPF-libraries as a form of geometrical notation. This variety aligns with the modular intentions of this project.

This project is still in its early stages, with several possible directions for further development. One current limitation of these applications is that they do not break out of the medium of sound synthesis. While there are notation-based parts in these applications, they are limited to mediating the journey of beginning with input sounds and ending with new sounds. Functions for score synthesis would be a clear step forward and would provide one further level of integration for a composer's CAC environment. A score synthesis function would be particularly useful for the simultaneous notation and synthesis of electronic parts.

If the applications were evaluated purely on their accuracy of timbre classification, then the use of classic ML methods can also be considered a limitation. For example, incorporating few-shot object detection into the classification application would likely increase its accuracy and efficiency in transcribing drum hits.

Acknowledgments

The first author would like to thank the second author, who has not only provided generous advisement, support, and resources for this research, but also developed several of the software tools used as well as theoretical models that supported the foundation of this project. The first author also thanks the students he works with at the Institute for Music Informatics at the Hochschule für Musik Karlsruhe. Their interest has provided fresh insight in developing the modular nature of this project's software design. The applications presented in this paper are available to the community for download, along with accompanying audio. [9]

The sounds for training, testing, and clustering were curated from: Mark Guiliana Drums by Loop Loft (drums) [10], Musicradar.com (drums) [11], Freesound.org (field recordings) [12] [13] [14] [15] [16], Marcus Weiss (tenor saxophone) [11], and Gerhard Krassnitzer (clarinet) [12]. The violin harmonics were produced by the first author.

4. REFERENCES

[1] M. Schumacher, "A Framework for Computer-Aided Composition of Space, Gesture, and Sound. Conception, Design, and Applications." Ph.D. dissertation, McGill University, Music Technology Area, Schulich School of Music, Jun. 2016.

[2] R. A. Fiebrink and B. Caramiaux, *The Oxford Handbook of Algorithmic Music*. Oxford University Press, 2018, ch. The Machine Learning Algorithm as Creative Musical Tool.

[3] D. Schwarz, "Corpus-Based Concatenative Synthesis," *Signal Processing Magazine*, pp. 92–104, 2007.

[4] B. Hackbarth, N. Schnell, and D. Schwarz, "Audioguide: a Framework for Creative Exploration of Concatenative Sound Synthesis," Tech. Rep., Mar. 2011.

[5] M. Schumacher, "Ab-Tasten: Atomic Sound Modeling with a Computer-Controlled Grand Piano," in *The OM Composer's Book: Volume 3*, J. Bresson, G. Assayag, and C. Agon, Eds. Éditions Delatour France / IRCAM—Centre Pompidou, 2016, pp. 341–359.

[6] A. Vinjar, "OMAI: An AI Toolkit For OM#," in *Linux Audio Conference*, Bordeaux, France, November 2020, pp. 12–15.

[7] J. Garcia, J. Bresson, M. Schumacher, T. Carpentier, and X. Favory, "Tools and Applications for Interactive-Algorithmic Control of Sound Spatialization in OpenMusic," https://hal.archives-ouvertes.fr/hal-01226263, Karlsruhe, Germany, 2015, [Online; accessed 5-January-2022].

[8] J. Salamon, "What's Broken in Music Informatics Research? Three Uncomfortable Statements," https://slideslive.com/38917449, Long Beach, California, USA, June 2019, [Online; accessed 5-January-2022].

[9] A. M. Gioti, "Imitation Game: Real-time Decision-making in an Interactive Composition for Human and Robotic Percussionist," https://www.researchgate.net/publication/340979353_Imitation_Game_Real-time_Decision-making_in_an_Interactive_Composition_for_Human_and_Robotic_Percussionist, New York, USA, June 2019, [Online; accessed 5-January-2022].

[9] `https://edu.marlonschumacher.de/audio-ml-4-cac/`

[10] `https://www.thelooploft.com/products/mark-guiliana-drums?variant=959418489`

[11] `https://www.musicradar.com/news/drums/1000-free-drum-samples`

[12] `https://freesound.org/people/Jack_Master/sounds/384466/`

[13] `https://freesound.org/people/Grupo2SONIDO/sounds/255214/`

[14] `https://freesound.org/people/doxent/sounds/386009/`

[15] `https://freesound.org/people/man/sounds/29580/`

[16] `https://freesound.org/people/laribum/sounds/353025/`

[10] D. Schwarz, “State of the Art in Sound Texture Synthesis,” in *International Conference on Digital Audio Effects*, Paris, France, 2011, pp. 221–231.

[11] M. Weiss and G. Netti, *The Techniques of Saxophone Playing*. Bärenreiter, 2010.

[12] G. Krassnitzer, *Multiphonics für Klarinette mit deutschem System*. ebenos, 2002.

Les éditions de PRISM 2022

9 791097 498030