TENOR

INTERNATIONAL CONFERENCE
ON TECHNOLOGIES FOR MUSIC
NOTATION & REPRESENTATION

Hamburg University for Music and Theater (HfMT)
Hamburg, Germany

Sixth International Conference on Technologies for Music Notation and Representation

2020/2021 Online Edition

Edited by:
Rama Gottfried, Georg Hajdu, Jacob Sello,
Alessandro Anatrini, and John MacCallum.

ISBN 978-3-00-066930-9

International Conference on Technologies for Music Notation and Representation TENOR 2020/21
Hamburg, Germany (Online Edition)

Proceedings published by: Hamburg University for Music and Theater

Editorial coordination: Rama Gottfried

Logo design: Nicolas Taffin, updated for 2020/21 by Janina Luckow

Introduction

The Covid-19 pandemic brought about challenges to academia not seen before throughout the past decades. Conferences as well as classes, concerts and other types of events needed to be cancelled or moved to the online domain. Fortunately, networking technology has developed to such a degree that the challenges were met to a considerable extent. Last year, we decided to postpone the conference by a year as we had hoped that the pandemic would have subsided by now. These hopes were largely dashed, but we will be using robust and convenient technologies to overcome the obstacles imposed by the Internet, such as JackTrip for the choir concert on Tuesday night.

I would like to thank my team for their involvement in getting this conference off the ground, my thanks also go out to our partners at the HAW Hamburg who have made a conscious decision to overlap their Klingt gut! conference with us, and to the many undeterred participants who will be joining from their respective locations around the globe. I hope that by next year will be able to see each other in person again. I would also like to acknowledge the sponsors of this year's conference: The federal ministry of education and research (BMBF), the Hamburg authority for science, research, equality and districts (BWFGB), via their Innovative Hochschule and Landesforschungsförderung programs, the TENOR Network via the Canadian SSHRC program and the HfMT for providing the infrastructure for this memorable event.

Georg Hajdu
TENOR 2020/21 Conference Chair

TENOR 2020/21 Organizing Committee

Georg Hajdu (General Chair)
Rama Gottfried (Paper Chair)
Gregory Beller (Performance Chair)
Jacob Sello (Performance Chair)

Scientific Committee

Andrea Agostini (Conservatorio Giuseppe Verdi di Torino, Italy)
Miriam Akkermann (TU Dresden, Germany)
Alessandro Anatrini (HfMT Hamburg, Germany)
Constantin Basica (CCRMA, Stanford University, USA)
Jonathan Bell (Aix-Marseille University, France)
Gregory Beller (IRCAM, France)
Jean Bresson (Ableton, Germany)
Pierre Couprie (Institut de Recherche en Musicologie (IReMus), Université Paris-Sorbonne, France)
Aaron Einbond (City University of London, UK)
Jose Miguel Fernandez (IRCAM, France)
Dominique Fober (Grame, France)
Jason Freeman (Georgia Institute of Technology, USA)
Daniele Ghisi (IRCAM, France)
Jean-Louis Giavitto (IRCAM - UMR STMS 9912 CNRS, France)
Rama Gottfried (HfMT Hamburg, Germany)
Karim Haddad (IRCAM, France)
Georg Hajdu (HfMT Hamburg, Germany)
Richard Hoadley (Anglia Ruskin University, UK)
Alexander Refsum Jensenius (University of Oslo, Norway)
David Kim-Boyle (University of Sydney, Australia)
Kristin Kuldkepp (Hamburg University of Applied Science, Germany)
John MacCallum (HfMT Hamburg, Germany)
Thor Magnusson (Music Department, University of Sussex, UK)
Kilian Schwoon (University of the Arts Bremen, Germany)
Aigerim Seilova (HfMT Hamburg, Germany)
Mattias Sköld (KMH Royal College of Music in Stockholm, KTH Royal Institute of Technology, Sweeden)
Daniel Spreadbury (Steinberg Media Technologies GmbH, Germany)
Pierre Alexandre Tremblay (University of Huddersfield, UK)
Jeffrey Trevino (California State University, Monterey Bay, USA)
Paul Turowski (University of Liverpool, UK)
Aaron Wyatt (Edith Cowan University, Australia)

TENOR Steering Committee

Chair: Jean Bresson (Ableton, Berlin, Germany)
Sandeep Bhagwati (Concordia University, Montréal, Canada)
Dominique Fober (GRAME, Lyon, France)
Richard Hoadley (Anglia Ruskin University, Cambridge, UK)
Cat Hope (Monash University, Melbourne, Australia)

Table of Contents

TENOR 2020

TENOR 2021

CROWDED STAVES — RULES FOR SEMANTICS AND STYLE OF CONVENTIONAL MULTIPLE-VOICES-PER-STAFF MUSICAL NOTATION

Markus Lepper
semantics GmbH Berlin
post@markuslepper.eu

Michael Oehler
Hartmuth Kinzler
Osnabrück University
michael.oehler hkinzler @uni-osnabrueck.de

Baltasar Trancón
semantics GmbH Berlin
baltasar@trancon.de

ABSTRACT

In many variants of Common Western Notation (CWN) more than two voices can be notated together in one staff. Reading this kind of multi-voice notation implies complicated parsing decisions, taken by the trained musician's brain in most cases non-knowingly. This article makes them explicit, supposing a theoretical maximum of information retrieval and formal consistency.

1. INTRODUCTION

When talking about musical notation, very different viewpoints can be sensible. One of these is "notation as a precise encoding", which means that the graphic components of the notated piece of music can be translated by a mathematically well-defined process into the elements of a semantic model, and vice versa. While this approach is obviously not *fully* appropriate to most empirical situations, it can never be *totally absent*, because it represents an abstract and idealistic "basic idea of notation" as a perfect encoding system.

The method of *mathematical re-modeling* as applied in our SemPart project creates compound mathematical models, intended to mimic by a precise mathematical operation the empirically and historically given processes of encoding and decoding music notation. Thus these models *define the semantics* of notation, and the grid spanned by their variants offers precise nomenclature for *semantic and stylistic differences*.

We concentrate on the *Common Western Notation (CWN)*, which has evolved since the 17th century, has been successfully adapted since then in many variants to new developments in composition, practice and theory of music, and is nowadays still in vivid development.

The created models basically consist of (a) a comparatively simple mathematical structure representing the graphical appearance (= "external representation" = "syntax sphere"), namely the graphical components' classes and the syntactical rules for their combination, (b) a collection of arbitrarily complex mathematical objects (values, vectors, sets, relations, functions, constraints) representing the intended musical parameters (= "semantic sphere"), (c) a mathematical function mapping syntax to semantics (= "parsing function") and/or (d) a function mapping semantics to syntax (= "writing function").

For all the technical details and philosophical implications of this approach see [1], where we applied this method to the notation of musical dynamics. From these principles are relevant for this article:

(A) In no case there is "one single true semantic model". Instead, we always found a multitude of similar models (or to say: one model which can be widely parameterized) which represent different usages from contemporary or historical practice. These differences range from slightly different flavors to opposite and incompatible approaches. The intention of the SemPart project is to present the different models to the domain specialists of historic or systematic musicology, music pedagogy, music performance practice, editing, printing, computer programming, etc., to let them select according to their needs, and so offer to all of them a nomenclature for talking about common features and differences in a more precise way.

(B) While the historic evolution and the social processes are the main forces which have defined the syntax and semantics of music notation, this is completely excluded from the modeling. The SemPart method is an *ahistoric* one, which refers to examples from history only for inspiration and for an a posteriori verification by application.

(C) There are important *empirical* results about the reading process of music notation, e.g. [2], [3], [4] and [5]. The meta-study [6] evaluates 15 studies on eye movement "to support the crafting of more well-founded research hypotheses and the more systematic design of experiments". Applying methods from psychology and neurology, all these are completely complementary to our approach, which extracts by abstract methods an idealistic information contents which can only serve as the theoretical maximum of retrievable information. One could say we define a proposal for the "Abstract Internal Representation", as it is graphically symbolized by the question mark in Figure 1 of [4], and which is explicitly *not* part of their work. E.g., we define a parsing algorithm only to define the abstract information contents, not as a concrete model of the empirical process of reading.

(D) The re-modeling approach presented in this paper also differs fundamentally from the well-known task of *voice separation*, see [7], [8] (both also for surveys), [9], [10] and many others. Those papers cover a concrete data-processing task, required for automated music notation, analysis and retrieval, and the different solutions take into account very different aspects of perception, precision, historic rules, technical representation, etc. In contrast, the SemPart project addresses a more basic level and simply tries to concretize (the different variants of) the mental and cultural *definitions* which are applied when notation is read or written by human actors. This concretization is also given in form of an *operational semantics*, i.e. an algorithm which must be executed to yield results, but this algorithm is only the means for the analyses, not their purpose.

2. PROCESSING PIPELINE FOR PARSING MULTI-VOICE STAVES

2.1 Outline of the problem

This paper applies the fundamental modeling technique of the SemPart programme to the task of recognizing more than two voices notated in the same staff. The idiosyncrasies of the historical evolution of CWN made this parsing process much more complicated and thus mathematically interesting than it appears at the first glance: In daily practice it is nearly always executed by any musician without further difficulties, but also without realizing the complicated intellectual processes required. The following algorithms do nothing more than making these processes visible.

For the following discussion, a *voice* is a sequence of *adjacent notated events* in time. Each event can be a *sounding event (=sound)* or a *pause* and has one particular *duration*, measured in some musical time system. [1]

Even the oldest known example of Western Notation writes two voices in one staff (*Musica Enchiriadis*, using text syllables for note heads, see page 57r of [11].) Much later, in *Ars Nova*, stem direction was used for voice indication [12].

Later more than two note heads simultaneously appeared in course of notating instruments which are capable of (sporadic) chord playing. This case is *not* covered in this article.

Here we look only at more than two parallel, independent and contiguous voices sharing one staff. In practice, these appear (a) in orchestra scores, when more than two "voices" in the sense of "note text for one melody instrument" are comprehended in one staff, and (b) in notes for keyboard instruments, where "voices" in the sense of abstract "voice leading" can spontaneously come into existence and vanish again. (Case (b) comprises also some string instruments like guitar, harp, hackbrett etc.)

Esp. in piano literature, starting with the pre-classics like C.Ph.E. Bach and D. Scarlatti, complicated combinations of these "keyboard voices" have been realized. Here we do not discuss what these voices are intended to represent, whether they are meant to be heard by the listener, how they should be played by the musician or whether they live only on a pure conceptual level, etc.

Here we are interested only in the mere graphical notation and how this is decoded or "parsed" into a data set which identifies the *relation from events to voices(= to voice names)*. For this we feed one single staff with its contents, called *original staff*, through a pipeline of processing steps.

Figure 1. Different note heads for filtering voices

2.2 Vertical filter

The first step separates the information from different layers in the original staff if they are immediately distinguished by a graphical attribute. In course of the more and more complicated voice textures, especially in piano literature, some authors used such attributes for a first clarification of the voice leading. The most common means are to use a *smaller size of note heads* for marking one particular voice. Also different *note head forms* are possible, like cross or diamond. Figure 1 shows an example from a 20th century piano sonata.

In modern scores, for instance if presented on a digital display, also *different colors* are easily possible.

If such a graphical attribution is present, this step separates the contents of the original staff into separate homogeneous staves according to these attributes, called *filtered staves.* In case that horizontal gaps do result, these have to be filled by additional pauses. The next steps are applied to each of these filtered staves separately in basically the same way as to a homogeneous original staff.

[1] For this article we assume without loss of generality this time system to be metrical duration, with the conventional rational numbers as duration values, and with the conventional graphical duration symbols. Nevertheless, the algorithms presented in the following would work also on arbitrarily different bases.

2.3 Horizontal segmentation

In an orchestra score, when several melody instruments share a staff, the identity of the instruments is established by some *explicit voice order declaration*. It lists the names of the instruments in one particular sequential order, which shall be mapped by the reader to the vertical order of the simultaneously executed events.

In many cases this explicit declaration is not constant over the whole duration of the score but changes by *explicit (voice order) re-declaration*. For our algorithm, a staff can be cut at such a point and the resulting segments treated separately, if this declaration mentions all existing voices. Otherwise its evaluation is much more complicated and part of the parsing process, as explained below.

In keyboard music the voices are (normally) not named or declared explicitly and can come into existence and vanish again spontaneously. Here we must cut between segments with different numbers of voices and treat them separately. [2] The naming of the voices (which is needed by the algorithm) is thus implicit and synthetic, like "`voice1`", "`voice2`", etc.

Contrarily, in keyboard fugues like "Das Wohltemperierte Klavier" the number of overall voices is given by the title like "Fuga à 5", which can be expanded to a canonical initial voice declaration "`v1, v2, v3, v4, v5`". [3]

2.4 Input data of the algorithms

Finally we apply parsing to each staff data which results from vertical filtering and horizontal segmentation. The task is to map each note head to a voice name, given the sequence of note heads and an initial voice order declaration.

We model the notation text as a sequence of sets of note heads. The type K in module *perLineas* in Table 1 [4] represents one note head as a tuple of its duration (as a rational number), pitch (natural number for the vertical position in the staff), "fine x" (the horizontal order of the note heads which belong to the same "chord" / time point), the direction of the stem, a flag whether it is a pause, and optionally the start coordinate of an incoming voice leading line. Such a coordinate from Ptx is a tuple of score position (as defined below), vertical position (=pitch) and fine x value and thus identifies a note head unambiguously.

The complete input data to the parsing algorithm is of type N. It includes a sequence of sets of note heads from $\mathbb{P}K$. All note heads in such a set are played synchronously. The sets follow with non-zero delay in temporal order; the index into this sequence is called *score position*, is taken from $\mathbb{N}_1$ and corresponds to the sequence of passing musical time points. [5]

At each combination of score position and voice, there is exactly one note head (not tied to its predecessor) or pause symbol, iff the preceding event in this voice ends at the time point represented by this position. These voices are called *(currently) active (at this score position)*; the other voices are *overlapping*. Thus the main task is to create at any given score position a bijective map between its note heads and its active voices.

Beside these sets, each score position can carry two lists of voice names from V for voice re-declaration, see section 4.1 below. So N represents the above-mentioned *syntax sphere* of the translation model. The *semantic sphere* is represented by the data type R which represents the result of the parsing algorithm, see section 4.3 below.

Since traditionally the "first oboe" sounds higher than the "second", the declaration sequence of voice names from left to right is mapped from top to bottom to the note heads, which corresponds to the physical pitch of the sounding events. This results in the *current (physical) voice order (=CVO)*. The CVO is a status data which reigns the assignment of note heads to voice names. It can be altered by subsequent *explicit re-declarations*, but also by voice leading lines and by other necessities resulting from the parsing process. Therefore it must be passed through all functions of our parsing algorithm as an explicitly modeled datum.

3. ISO-RHYTHMIC VOICES

First we analyze the situation in iso-rhythmic settings, i.e. without overlapping voices. The results of voice crossing analysis can easily be transferred to the more complex case of hetero-rhythmic voices, but doing so for pause sharing is left to future work.

[2] A change in the visible number of voices can also result from pause sharing and should be treated accordingly, see section 3.1.

[3] In practice, two different and disjoint declarations for both staves of the keyboard staff must be generated. This problem is not treated in this article, – all algorithms here treat one single staff only. Furthermore, they may appear more voices than mentioned by the title, typically in a coda section.

[4] The mathematical notation is based on the well-proven Z notation [13], with only few extensions for brevity. See the appendix for details.

[5] In the Z formalism, which is the basis for our modeling, indices of sequences start with the number One(1).[13] Thus the encoding of the notes in Figure 2 is a sequence of sets of tuples. starting with
$\langle\{(1/4, e'', 1, \perp, \texttt{true}, \perp), (1/4, e', 1, \perp, \texttt{true}, \perp)\},$
$\{(1/4, e'', 1, \uparrow, \texttt{false}, \perp), (1/4, e', 1, \perp, \texttt{true}, \perp)\},$
$\{(1/4, e'', 1, \uparrow, \texttt{false}, \perp), (1/4, e', 1, \downarrow, \texttt{false}, \perp)\},$
$\ldots\rangle$
and the top line of Figure 2 holds the data
$\langle\{(3/8, g'', 1, \uparrow, \texttt{false}, \perp), (3/4, e'', 2, \uparrow, \texttt{false}, \perp),$
$(3/4, c'', 2, \uparrow, \texttt{false}, \perp), (7/8, c', 1, \downarrow, \texttt{false}, \perp)\},$
$\{(1/8, g', 1, \uparrow, \texttt{false}, \perp)\},$
$\{(1/2, g', 1, \uparrow, \texttt{false}, \perp)\},$
$\{(1/4, d'', 1, \uparrow, \texttt{false}, \perp), (3/4, h', 1, \uparrow, \texttt{false}, \perp)\},$
$\{(1/8, d', 1, \downarrow, \texttt{false}, \perp)\},$
$\{(1/2, d'', 1, \uparrow, \texttt{false}, \perp), (1/2, d'', 1, \uparrow, \texttt{false}, \perp),$
$(1/2, g', 1, \uparrow, \texttt{false}, \perp), (1/2, e', 1, \downarrow, \texttt{false}, \perp)\},$
$\ldots\rangle$
The right part in the second row of Figure 5 holds
$\langle\{(1/8, c'', 1, \uparrow, \texttt{false}, \perp), (1/4, a', 2, \uparrow, \texttt{false}, \perp)\},$
$\{(1/8, h', 1, \uparrow, \texttt{false}, \perp)\},$
$\{(1/4, a', 1, \uparrow, \texttt{false}, \perp), (1/8, e', 2, \uparrow, \texttt{false}, (2, h', 1))\},$
$\{(1/8, d', 1, \uparrow, \texttt{false}, \perp)\},$
$\ldots\rangle$

module perLineas
$C = \{\uparrow, \downarrow, \perp\}$
// stem direction: up, down, not present
$\mathrm{Ptx} = \mathbb{N}_1 \times \mathbb{N}_1 \times \mathbb{N}_1$
// coordinates: score position, pitch, fine x
V *// given Type: Voice names*
$K = \mathbb{Q} \times \mathbb{N}_1 \times \mathbb{N}_1 \times C \times \mathsf{bool} \times \mathsf{optPtx}$
// note head = duration, pitch, fine x,
// stem, is pause, start of incoming lead line
$N = \mathsf{seq}(\mathbb{P}K \times \mathsf{iseq}V \times \mathsf{iseq}V)$
// score data = note heads and voice decls.
$R = V \twoheadrightarrow \mathsf{iseq}\mathrm{Ptx}$
// result, maps voice names to the
// coordinates of their note heads

$\mathrm{lexSquash} : (A \twoheadrightarrow \mathsf{seq}\mathbb{Z}) \to \mathsf{iseq}A$
// sort the key elements by the mapped values,
// lexicographically.
$\mathrm{sortH_X}, \mathrm{sortH_Y} : \mathbb{P}K \to \mathsf{iseq}\ K$
$\mathrm{sortH_X}(k) = \mathrm{lexSquash}(k \lhd \pi_3)$
$\mathrm{sortH_Y}(k) = \mathrm{lexSquash}(\lambda x : k \bullet (-1 * \pi_2(x), \pi_3(x)))$

module *perLineas.pausae
import perLineas
$d : \mathbb{N}_1$ *// number of voices active in* $\widehat{k}$
$\widehat{k} : \mathsf{iseq}K$
$\forall i, j \bullet i < j \implies \pi_2(\widehat{k}(i)) \geq \pi_2(\widehat{k}(j))$
$\mathrm{pausae}(\widehat{k}) = \{i : \mathbb{N}_1 \mid \pi_5(\widehat{k}(i)) = \mathit{true}\}$
$\mathrm{pSequ}(\widehat{k}) = \mathrm{pausae}(\widehat{k}) \setminus \mathit{succ}(\!|\mathrm{pausae}(\widehat{k})|\!)$
$\mathrm{pAdj}(\widehat{k}) \iff \#\mathrm{pausae}(\widehat{k}) > \#\mathrm{pSequ}(\widehat{k})$
$\mathrm{nonCommunes}(\widehat{k}) \iff d = \#\widehat{k}$
$\mathrm{maxCommunes}(\widehat{k}) \iff \neg\mathrm{pAdj}(\widehat{k})$
$\mathrm{defectis}(\widehat{k}) \iff d > \#\widehat{k} \wedge \mathrm{pAdj}(\widehat{k})$
$\iff \neg(\mathrm{nonCommunes}(\widehat{k}) \vee \mathrm{maxCommunes}(\widehat{k}))$
$\mathrm{ambig}(\widehat{k}) \iff \#d > \#\widehat{k} \wedge \#\mathrm{pSequ}(\widehat{k}) > 1$

Table 1. Note heads and sharing of pauses.

3.1 Sharing of pauses

The historical development process of CWN brought up many rules for special cases. These make the parsing process non-orthogonal and complicated. Two of them are

nota.perLineas.pausae.communes: Multiple pauses of the same duration in adjacent voices *may* be represented by one single graphical symbol.

nota.perLineas.pausae.maxCommunes: Multiple pauses of the same duration in adjacent voices *must* be represented by one single graphical symbol.

The first rule is implied in the second. Module *perLineas.pausae* in Table 1 shows the definitions of the most important respective properties, when $\widehat{k}$ is an injective list of note heads and pause symbols, sorted by vertical height.

To recognize (=parse) sharing of pauses, the total number of intended voices must be known by some preceding explicit voice declaration. This is modeled by d in that Table.

The set pausae contains all those indices where a pause stands; the set $\mathrm{pSequ}(\widehat{k})$ all those where a maximal group of adjacent pauses starts, e.g. $\{1,2,4\} \setminus \{2,3,5\} = \{1,4\}$. The property $\mathrm{pAdj}(\widehat{k})$ indicates that there are adjacent pause symbols. Figure 2 shows minimal examples for characteristic combinations:

	a)	b)	c)	d)	e)	f)
pAdj	☒	☐	☐	☒	☐	☒
nonCommunes	☒	☒	☐	☐	☐	☐
maxCommunes	☐	☒	☒	☐	☒	☐
defectis	☐	☐	☐	☒	☐	☒
ambig	☐	☐	☐	☐	☒	☒
$d =$	2	2	2	3	4	5

Figure 2. Top part check boxes indicate the properties of the note example below, given a particular voice count d.

nota.perLineas.pausae.maxCommunes: All occurring maximal groups of adjacent pauses have been replaced by one single pause symbol.

nota.perLineas.pausae.nonCommunes: No pair of adjacent pauses has been replaced by one single pause symbol.

nota.perLineas.pausae.defectis: At least one pair of adjacent pauses has been replaced by one single pause symbol but another pair has not.

nota.perLineas.pausae.ambig: There are multiple pause symbols which can be expanded to the eliminated pause symbols.

The minimal voice number for ambiguity is indeed four: One non-pause voice is required to separate the two pause symbols; one of these stands for two voices. To combine this with *defectis* requires a further voice with a non-replaced pause symbol.

These attributes play an important role in practice. The piano reduction contained in Wolfgang Graeser's edition of Bach's "Die Kunst der Fuge" BWV 1080 [14] consequently follows nonCommunes, which implies ¬ambig and thus makes voice leading as clear as possible. [6] In contrast, Walter Denhard's edition of "Das wohltemperierte Klavier I" [15] changes frequently: Fuga I à 4 starts with ambig, but Fuga III à 3 with ¬maxCommunes ∧ nonCommunes. Fuga IV à 5 starts with maxCommunes ∧ ¬ambig. [7] The second half of measure 7 starts a new declaration segment of only three voices (see section 2.3 above), because only these are notated without any accompanying pause. Similar in measure 21, when the pauses of voice 2 suddenly

[6] For these remarks we consider the two staves of the keyboard notation as one single staff. The formally correct treatment of this case is complicated and out of scope in this paper.

[7] Reading both staves into one additionally yields defectis, which is an artefact.

disappear while those of voice 5 are visible. Contrarily, measure 35 switches to ¬maxCommunes, showing adjacent note symbols. Starting with measure 77, even ledger lines are employed for these.

3.2 Crossings of iso-rhythmic voices

One of the main concerns in many contexts is a lightweight but unambiguous way to indicate *temporal voice crossings*. This means that the physical top-down order of the sounding notes of the voices contradicts the declared voice name order (= CVO), but only for a short time and without the need for a full voice order re-declaration.

According to the historic evolution, all voices with *stem up* notes are read as a multiplication (or splitting) of one original "upper voice" or "first voice", and all *stem down* notes correspond to a "second voice". Therefore a convenient but limited means for voice crossing is the *stem direction*. Given a certain explicit voice declaration $\langle v_1, v_2, \ldots, v_n \rangle$, and a set of n note heads in which $u < n$ are stem-up, then the first voice names $\langle v_1, \ldots, v_u \rangle$ are assigned to the stem-up note heads, from top to bottom, and the stem-down note heads are mapped to $\langle v_{u+1}, \ldots, v_n \rangle$, again from top to bottom.

In an iso-rhythmic setting, voice crossings can be expressed between the lowest voices with up-stem and the highest voices with down-stem, but not among voices with the same stem direction. Figure 3 shows all possible situations for four-voice chords.

The number of all expressible voice crossings is calculated as follows: Given a chord of n note heads, all with stems and with heads on different heights (= different pitches), and u of them shall be stem-up. Then there are $\binom{n}{u}$ possibilities to assign stem directions. $u = 0$ and $u = n$ do not allow any voice crossing, so we get

$$\sum_{u=1}^{n-1} \binom{n}{u}$$

for the number of possible assignments. For each u, there is one combination which is not a crossing, namely the first in each group in Figure 3. For the number of all possible permutations (original order plus all crossings) we must thus subtract $n - 2$. All other permutations are indeed different, what can be shown as follows: for any given u, the voices u and $u + 1$ (counted from the top) have different stem directions and thus are crossed in all combinations but the very first (see Figure 3). For all $m \neq u$ the voices m and $m + 1$ have the same stem direction and thus never cross. So all permutations are different.

Further normalization yields

$$\sum_{u=1}^{n-1} \binom{n}{u} - n + 2 = \sum_{u=1}^{n-1} \binom{n}{u} - n + \binom{n}{0} + \binom{n}{n}$$

$$= \sum_{u=0}^{n} \binom{n}{u} - n = 2^n - n$$

The table in Figure 3 shows the absolute and relative numbers of expressible voice crossings in the iso-rhythmic case.

4. PARSING OF FREE RHYTHMIC VOICES

In hetero-rhythmic settings more voice crossings are possible: At each score position only the active voices (as defined above) get a new note head. [8] So arbitrary voice crossings can be notated between an active and an overlapping voice, see Figures 4 and 7. Additionally, for more flexible voice crossings, it may be sensible to allow *changes of the stem direction*, which makes parsing significantly more complicated.

Voice crossings between active voices can be indicated by explicit *voice leading lines*. These connect visibly the latest preceding note head of a particular active voice with a note head at the current score position and declares graphically that both note heads belong to the same voice, see Figure 5.

4.1 Voice (re-)declarations

Initial voice declarations (at the beginning of a staff or a segment) must mention all present voices. They can come in two formats: (a) as one list of voice names for all voices, or (b) as two separate lists for up- and down-stem voices, written above and below the staff. In any case, the respective numbers of names and voices must match. In case of the single list, the leading names are matched to the up-stem voices and the trailing names to the down-stem voices. The version with two lists can contain slightly more information when the staff begins with stemless notes, because it declares the voices in the second list as "nominally down-stemmed", see Section 4.7 below.

A re-declaration stands at a particular score position in the midst of a segment. It comes in similar formats: (a) one list for both stem directions or (b) two lists for up-stemmed and down-stemmed each, or (c) one list only for up-stemmed or (d) one list only for down-stemmed heads. These formats must be recognizable by the position(s) and length(s) of the list(s). Furthermore, the declaration can address different targets: (A) All voices appearing in the segment, or (B) only the voices with a note head at this score position, or (C) only the active voices (=only the heads not tied to its predecessors).

Re-declarations either (α) change the CVO, or (β) they redundantly only confirm it and show it to the reader for convenience. These are **two fundamental different semantics**, because the latter can be erased from the notation without changing its meaning. This results in twenty-four(24) combinations for the types of voice re-declaration, all of which can be found in practice. Figure 4 shows some examples.

[8] In the following we write "note head" for shortness, also when "note head or pause symbol" is meant.

n	$n!$	$2^n - n$	expressible/total
2	2	2	1
3	6	5	0.833
4	24	12	0.5
5	120	27	0.225
6	720	58	0.0805
7	5040	121	0.0224

Figure 3. Voice crossings expressible by stem directions

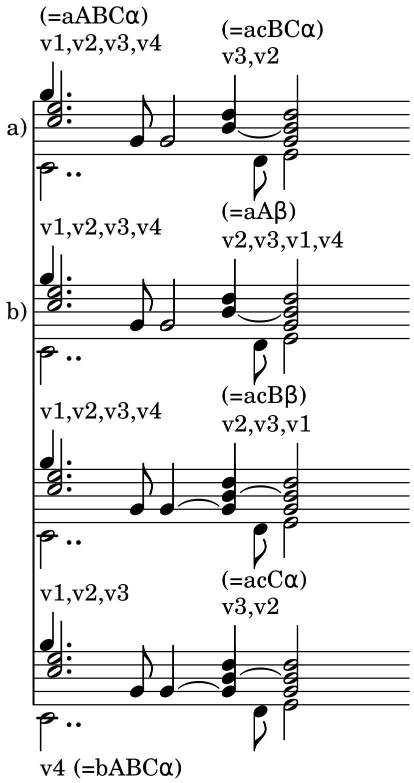


Figure 4. Voice (re-)declarations and crossings

Ambiguities can only arise between cases cA vs. aB, cA vs. aC, and cB vs. aC, and can in most cases be resolved by the nominal stem direction of the named voices.

4.2 Preparatory steps

To apply our algorithm to concrete examples from practice, some *preparatory transformations* must be applied. These model the extraction of *additional information* for voice parsing (e.g. from beams and slurs) which is done by the trained musicians brain unknowingly. We realize it by adding explicit voice lead lines, see Figure 5. Additionally all *ties* are removed by replacing e.g. the combination 𝅗𝅥 ‿ 𝅘𝅥𝅮 by a single (artificial) note head with a duration value of "5/8". [9]

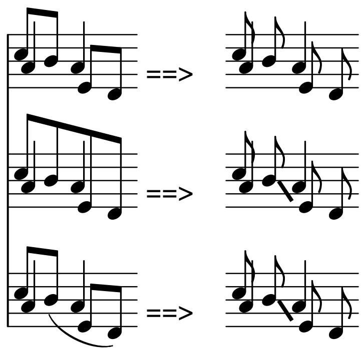


Figure 5. Preparatory adding of voice lead lines from beams and slurs (not formalized in this paper)

4.3 Parsing

The following tables show different versions of a parsing algorithm. This is done in a *modular* way of specification. Tables 2 and 3 show the fundamental and auxiliary functions, common to all versions. Table 4 gives the more simple variant which treats filtered segments containing only one stem direction; Table 5 shows the more complex version in which each voice may arbitrarily change its stem direction. Both algorithms come in two different flavors, *suonaeProximae* and *solumExplicitum*, as explained below. Result is R, which gives for each voice the sequence of the coordinates of all note head contained in that voice. [10]

The parsing function calc($n : N$) calls the real computation function step(..) [11] with this input data, the

[9] Indeed, the problem of "ties vs. slur" has most intricate cases, philosophically and mathematically most complex. This is left out in this article. E.g., the function $\mathrm{assign}_{\mathrm{D}}(..)$ in Table 2, which treats explicit re-declarations, does not treat those of type "B" but only "C", so it does not need to know about incoming ties.

[10] Since coordinates represent the score position order internally, the modeling as sequences per voice is redundant, but practical; semantically sets would suffice.

[11] We write "$f(..)$" to refer to a function by its name, while omitting the full type signature.

module $*\mathrm{perLineas.extrahere}[H]$ // *H = auxiliary data = running status, parameterizable = CVO*
import perLineas
$\mathrm{read} : N \times \mathrm{Ptx} \nrightarrow K$ // *extract note head at given coordinate from score data*
$\mathrm{store} : R \times (V \times \mathbb{N}_1 \times K) \rightarrow R$ // *assign voice at score position to note head*
$\mathrm{calc} : N \nrightarrow R$ // *main function = analyze score data and initial voice declaration*
$*\mathrm{initH} : N \nrightarrow H$ // *initialize running status; defined by importing module*
$\mathrm{step} : N \times \mathbb{N}_1 \times (V \nrightarrow \mathbb{Q}) \times R \times H \nrightarrow R \times H$ // *score data, score position, voice ends, result, running status*
$\mathrm{assign_W} : \mathbb{N}_1 \times \mathbb{P}V \times \mathbb{P}K \times \mathbb{P}K \times \mathsf{seq}V \times \mathbb{P}V \times R \times R \times H \nrightarrow R \times H$
// *score pos., active voices, heads with/without lead, re-declaration, active voices, old/current result, aux state*
$\mathrm{assign_F} : \mathbb{N}_1 \times \mathsf{seq}V \times \mathsf{seq}K \times R \times H \times \mathbb{P}V \times R \nrightarrow R \times H$
// *score position, active voices, note heads, result, running status, old active voices, old result*
$\mathrm{assign_D} : \mathbb{N}_1 \times \mathbb{P}V \times \mathbb{P}V \times \mathbb{P}V \times \mathbb{P}V \times \mathsf{seq}K \times \mathsf{seq}V \times R \times H \times \mathbb{P}V \times R \nrightarrow R \times H$
// *score pos., overlap/tied/led/active voices, sorted heads, voice decls, result, old CVO, up-stem voices, old result*
$*\mathbf{phase2} : \mathbb{N}_1 \times \mathbb{P}V \times \mathbb{P}K \times \mathsf{seq}V \times \mathbb{P}V \times R \times R \times H \nrightarrow R \times H$
// *score pos., active voices, heads without lead, re-declaration, active voices, old/current result, aux state*
$*\mathbf{finalize} : \mathbb{N}_1 \times R \times H \times \mathbb{P}V \times R \nrightarrow R \times H$
// *score position, result, running status, old active voices, old result*
$\mathrm{lastOrd} : R \times \mathbb{P}V \rightarrow \mathsf{iseq}\ V$ // *sort the set of voices according to their latest occurrence in the result*
$*\mathrm{aboveVK}_{\_,\_}, *\mathrm{equalVK}_{\_,\_} : R \times \mathbb{P}V \rightarrow (V \leftrightarrow K)$
// *result × stem dir → relation whether a overlapping voice is above/on equal height with a note head,*

$$\mathrm{read}(p, (m, t, x)) = k \iff k \in \pi_1(p(m)) \ \wedge k = (\_, t, x, \_, \_, \_)$$

$$\mathrm{store}(r, (v, m, (\_, t, x, \_, \_, \_))) = r \oplus \{v \mapsto r(v) \frown \langle (m, t, x) \rangle\}$$

$$\frac{\mathrm{initH}(p) = (v, u)}{\mathrm{calc}(p) = \pi_1(\ \mathrm{step}(p, 1, (\mathsf{ran}v) \times \{0\}, (\mathsf{ran}v) \times \{\langle\rangle\}, (v, u)))}$$

$$\frac{\begin{array}{c} q_0 = \mathsf{min}(\mathsf{ran}\, q) \quad \widehat{v} = q^{-1}(\!|\{q_0\}|\!) \quad \widehat{k} = \pi_1(p(m)) \quad d = \pi_2(p(m)) \\ \#\widehat{v} \neq \#\widehat{k} \implies \texttt{error}(\text{“}\textit{Numbers of expected and notated note heads differ}\text{”}, m) \\ k_W = \{k \in \widehat{k} \mid \pi_6(k) \neq \bot\} \quad k_F = \widehat{k} \setminus k_W \\ \#k_W = 1 \ \wedge \ \#k_F = 0 \implies \texttt{warning}(\text{”}\textit{Redundant voice leading line}\text{”}, m) \\ (r', h') = \mathrm{assign_W}(m, \widehat{v}, k_W, k_F, d, \widehat{v}, r, r, h) \\ q' = q \oplus \lambda v : \widehat{v} \bullet q_0 + \pi_1[\mathrm{read}(p, \mathit{last}(r'(v)))] \end{array}}{\mathrm{step}(p, m, q, r, h) = \mathrm{step}(p, m+1, q', r', h')}$$

$$m > \#p \implies \mathrm{step}(p, m, \_, r, h) = (r, h)$$

$$\frac{\begin{array}{c} k_W = \pi_6(k) : \mathrm{Ptx} \quad \exists v_W \bullet \mathit{last}(r(v_W)) = k_W \\ v_W \notin v \implies \texttt{error}(\text{“}\textit{Leading line comes from non-active voice}\text{”}, k) \\ r' = \mathrm{store}(r, (v_W, m, k)) \quad v' = v \setminus \{v_W\} \end{array}}{\mathrm{assign_W}(m, v, \{k\} \cup k_W, k_F, d, v_0, r_0, r, h) = \mathrm{assign_W}(m, v', k_W, k_F, d, v_0, r_0, r', h)}$$

$$\mathrm{assign_W}(m, v, \{\}, k_F, d, v_0, r_0, r, h) = \mathbf{phase2}(m, v, k_F, d, v_0, r_0, r, h)$$

Table 2. Common and auxiliary functions for parsing two or more voices per staff

next score position to process (initially $m = 1$), a map q of type $V \nrightarrow \mathbb{Q}$ which gives the next time point for each voice (initially all point to 0), and the result accumulator, which is initially empty.

The data type H is a parameter specific for the variants of the algorithm. It threads additional auxiliary status data through all function calls. Its initial value is delivered by initH(..), which must be defined accordingly in the importing module. [12] For brevity of the algorithm, the score must carry a complete voice order declaration of Type aAα with the very first event, and (for the double stemmed case) the nominal stem direction must be visible from the very first note heads.

The function step(..) calculates the next time point to process ($= q_0$) as the lowest value in the map q, and the set $\widehat{v}$ of all voices active there. $\widehat{k}$ are the note heads at the next score position m, and d the voice name redeclaration there, which at most score positions will be empty. The set $\widehat{k}$ is divided into those note heads at which an explicit voice leading line arrives ($= k_W$, from German “Stimm*W*eiser”) and the *F*ree rest ($= k_F$). First $\mathrm{assign_W}$(..) is called to process k_W. Afterwards the two variants of **phase2**(..) (in Tables 4 and 5) call the common functions $\mathrm{assign_D}$(..) if there is an explicit voice re-declaration, or $\mathrm{assign_F}$(..) if not. Afterwards, the specific function **finalize**(..) updates the running status H.

[12] Therefore the head of the module is marked with an asterisk, meaning “not a complete schema in the sense of Z”.

$$\mathrm{assign_F}(m, \langle v\rangle \frown \alpha, \langle k\rangle \frown \beta, r, h, v_0, r_0) = \mathrm{assign_F}(m, \alpha, \beta, \mathrm{store}(r, (v, m, k)), h, v_0, r_0)$$

$$\mathrm{assign_F}(m, \langle\rangle, \langle\rangle, r, h, v_0, r_0) = \mathbf{finalize}(m, r, h, v_0, r_0)$$

$$\frac{A = \lambda v \in \mathsf{dom} r \cap \widehat{v} \mid k = \mathit{last}(r(v)) \bullet (-1 * \pi_2(k), \pi_1(k), \pi_3(k))}{\mathrm{lastOrd}(r, \widehat{v}) = \mathrm{lexSquash}(A)}$$

$$\frac{r' = \begin{cases} \textbf{if } \#d = \#v3 + \#v4 & \textbf{then } \mathrm{zip}(m, v_4, \mathrm{lastOrd}(r, v_3), \widehat{k}, d, r, U) \\ \textbf{else if } \#d = \#v2 + \#v3 + \#v4 & \textbf{then } \mathrm{zip}(m, v_4, \mathrm{lastOrd}(r, v_2 \cup v_3), \widehat{k}, d, r, U) \\ \textbf{else if } \#d = \#v1 + \#v2 + \#v3 + \#v4 & \textbf{then } \mathrm{zip}(m, v_4, \mathrm{lastOrd}(r, v_1 \cup v_2 \cup v_3), \widehat{k}, d, r, U) \\ \textbf{else} & \mathtt{error}(\textit{“wrong length of voice declaration list”}) \end{cases}}{\mathrm{assign_D}(m, v_1, v_2, v_3, v_4, \widehat{k}, d, r, h, U, r_0) = \mathbf{finalize}(m, r', h, v_3 \cup v_4, r_0)}$$

$\mathrm{zip}(m, \widehat{v}, \langle v\rangle \frown \alpha, \langle k\rangle \frown \kappa, \langle d\rangle \frown \delta, r, U)$
// score pos., voices to allocate, fixed voices, heads to allocate, declarations, result, nominal upstem voices

$$= \begin{cases} \textbf{if } (v \text{ aboveVK}_{r,U}\ k) \textbf{ then} \begin{cases} \textbf{if } v = d & \textbf{then } \mathrm{zip}(m, \widehat{v}, \alpha, \langle k\rangle \frown \kappa, \delta, r, U) \\ \textbf{else} & \mathtt{error}(\textit{“declaration contradicts overlap/lead”}, d, v) \end{cases} \\ \textbf{else } (v \text{ equalVK}_{r,U}\ k) \wedge v = d & \textbf{then } \mathrm{zip}(m, \widehat{v}, \alpha, \langle k\rangle \frown \kappa, \delta, r, U) \\ \textbf{else if } d \in \widehat{v} & \textbf{then } \mathrm{zip}(m, \widehat{v} \setminus \{d\}, \langle v\rangle \frown \alpha, \kappa, \delta, \mathrm{store}(r, (d, m, k)), U\ ) \\ \textbf{else} & \mathtt{error}(\textit{“Voice for note head is not active”}, d, k) \end{cases}$$

$$\mathrm{zip}(m, \{\}, \langle v\rangle \frown \alpha, \langle\rangle, \langle d\rangle \frown \delta, r, \_) = \begin{cases} \textbf{if } v = d & \textbf{then } \mathrm{zip}(m, \{\}, \alpha, \langle\rangle, \delta, r, \_) \\ \textbf{else} & \mathtt{error}(\textit{“declaration contradicts overlap/lead”}, \end{cases}$$

$$\mathrm{zip}(m, \{\}, \langle\rangle, \langle\rangle, \langle\rangle, r, \_) = r$$

Table 3. Common and auxiliary functions, continued.

module perLineas.extrahere. $\boxed{\text{suonaeProximae} / \text{solumExplicitum}}$. $\boxed{\text{X} / \text{Y}}$
import perLineas.extrahere[$\mathsf{iseq}\,V$] *// parameter H used to pass through the CVO*

$$\frac{v = \pi_2(p(1))}{\mathrm{initH}(p) = \textbf{if } (\#v = \#\pi_1(p(1)) \textbf{ then } v \textbf{ else } \mathtt{error}(\textit{“Error in initial voice declaration”})}$$

$\mathbf{phase2}(m, v, k_{\mathrm{F}}, d, v_0, r_0, r, h)$ $\quad$ $// \#k_{\mathrm{F}} = \#v$

$$= \begin{cases} \textbf{if } d = \langle\rangle & \textbf{then } \mathrm{assign_F}(m, \mathit{squash}(h \rhd v), \mathrm{sortH}_{\boxed{X/Y}}(k_{\mathrm{F}}), r, h, v_0, r_0) \\ \textbf{else} & \mathrm{assign_D}(m, \mathsf{ran} h \setminus v_0, \{\}, v_0 \setminus v, v, \mathrm{sortH}_{\boxed{X/Y}}(k_{\mathrm{F}}), d, r, h, \{\}, r_0) \end{cases}$$

$$\mathbf{finalize}(m, r, h, v_0, r_0) = (r, \boxed{\mathrm{lastOrd}(r, \mathbb{N}_1) / h \mathbin{⨾} (\mathbf{ID}_{\mathbb{N}_1} \oplus (\mathrm{lastOrd}(r_0, v_0)^{-1} \mathbin{⨾} \mathrm{lastOrd}(r, v_0)))}\ )$$

$v \text{ aboveVK}_{r,\_}\ k \iff \pi_2(\mathit{last}(r(v))) > \pi_2(k)$
$v \text{ equalVK}_{r,\_}\ k \iff \pi_2(\mathit{last}(r(v))) = \pi_2(k)$

Table 4. Parsing multiple voices with all the same stem direction

On return from $\mathrm{assign_W}(..)$, the function step(..) calculates for all active voices their next time points in the map q': $\mathit{last}(r'(v))$ is the latest score coordinate recognized as part of v; $\mathrm{read}(p, \mathit{last}(r'(v)))$ reads the event at this position of the input score; $\pi_1(..)$ extracts the duration. Then step(..) calls itself recursively, until the input data from N is exhausted.

In $\mathrm{assign_W}(..)$, the variable k steps through the note heads k_{W}; the start coordinate of the voice leading line arriving there ($= \pi_6(k)$) must come from a note head which exactly ends at the current time point. So the voice assigned to it must be active ($v_w \in v$). It is assigned to k by calling the auxiliary function store(..) and is removed from the set v. [13]

The function $\mathrm{assign_F}(..)$ simply assigns the voices in its second argument to the note heads in its third, in that sequential order provided by the caller.

The function $\mathrm{assign_D}(..)$ gets in $v_1..v_4$ the sets of voices which are overlapping/tied/with lead-lines/still to allocate, and in d the declaration text. It recognizes cases aA, aB and aC, as defined in Section 4.1, by comparing the cardinalities of the sets. [14] It always

[13] The set of all initially active voices and the old result, before evaluating the voice leading lines, are additionally passed through by the function arguments v_0 and r_0, because they are needed later by one of the algorithm's variants.

[14] For brevity, cases b to d are not supported and case B is not called by the other modules: for brevity, ties are not modeled in our data. The algorithm treats cases α and β uniformly and

holds that $\#v_4 = \#\widehat{k}$. It calls zip(..), which iterates synchronously over three lists and one set. It gets as many declarations in d as voices in $\widehat{v} \cup v'$, and as many un-assigned voices in $\widehat{v}$ as heads to assign in $\widehat{k}$. It can assign the next note head (in the respective sorting order) to the next entry in the declaration, if this voice is in the set $\widehat{v}$ (= still unassigned). Otherwise it checks whether the topmost assigned voice (=overlapping, with lead lines, etc.) is in sync with the declaration. When the topmost note head and the topmost assigned voice are on the same height, both alternatives for their nominal order are considered. The set U of all nominal up-stemmed voices is only needed to compare the height of the current note head and the current assigned voice: in the double-stem case each nominally up-stemmed voice is infinitely higher than any down-stemmed note head.

4.4 Parsing only one stem direction

The first and more simple case is to process only all voices with one particular stem direction, see Table 4. The arguments for $\mathrm{assign}_\mathrm{F}(..)$ are simply the currently active voices in the sequential order of the CVO and the sorted noted heads. ($h \triangleright v$ is the "range restriction", which selects from the sequence h all maplets which point into v, and *squash*(..) compactifies this to a sequence.) The note heads are all those with no arriving lead line, sorted according to the selected method, see next section.

4.5 Two methods of note head sorting

For the sorting of the note heads (see Tables 4 and 5, for calling $\mathrm{assign}_\mathrm{D}(..)$ and $\mathrm{assign}_\mathrm{F}(..)$) there are two different methods: sortH_X sorts by the "fine x" horizontal position only; sortH_Y sorts by pitch = vertical position, and subordinately by fine x position, if necessary. Both functions are defined in Table 1.

These two methods imply **fundamentally different ways to express short-term voice crossings**: The method *perAltitudinemCaputis* uses sortH_Y and assumes that the physically lower voice is mentioned later in the CVO. The horizontal position is used as secondary criterion, only in cases of equal height. This method is widely used and allows sharing of stems and note heads (see below Section 5).

The method *perCaudaeSequentiam* uses sortH_X and defines the sequential order by the x position of the stem only. Therefore it can easily express short-time voice crossings, see Figure 6. It is not found as frequently as the preceding variant, but can also be found in practice.

4.6 Two strategies of changing the CVO

The four modules combined in Table 4 also differ in the strategy the CVO is finally affected by the crossings

can easily be enhanced for detecting and signaling them.

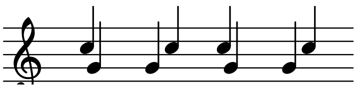
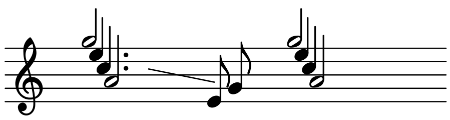
a) 

Figure 6. Voice crossing by x position of stems: voice one goes c-g-c-g, vs. voice two with g-c-g-c



suonaeProximae	1	1	1
(=physical)	2	**4**	4
	3	**3**	3
	4	**2**	2
solumExplicitum	1	1	1
(=nominal)	2	**3**	3
	3	**2**	2
	4	4	4

Figure 7. Physical vs. explicit-only changes of the CVO. Voice names below notes, changes in **bold**.

between active and overlapping voices: *perLineas.extrahere.suonaeProximae* takes the complete finally resulting *physical* order (calculated by $\mathrm{lastOrd}(r, \mathbb{N}_1)$, see the left part of the last box in the Table) as the new CVO, which will reign the parsing step at the subsequent score position.

Contrarily, *perLineas.extrahere.solumExplicitum* lets only explicit changes (by voice order re-declaration or by voice lead lines) change the CVO. The calculation is more complicated: $\mathrm{lastOrd}(r_0, v_0)$ is the sequential order of only the free voices, before any processing/assignment at this score position had been started. $\mathrm{lastOrd}(r, v_0)$ is the same after all evaluation. So $\mathrm{lastOrd}(r_0, v_0)^{-1} \fatsemi \mathrm{lastOrd}(r, v_0)$ is the permutation of these voices, a mapping from voice name to voice name. This permutation is now expanded to a mapping which is the identity on all other voices and then applied to the old CVO. (This is the only place where the function parameters v_0 and r_0 are needed.)

The effects of both strategies are illustrated in Figure 7: The last chord has different voice assignments depending on the CVO calculated after processing the next-to-last score position. These are again **two fundamentally different semantics** which must be declared or found out when talking about a given text. [15]

4.7 Parsing both stem directions

Here the parameter H is set to $\mathsf{iseq} V \times \mathbb{N}_0$. The CVO is the sequence of voice names, first the up-stemmed, then the down-stemmed, and the additional value (called u in the following) is the index after which the latter start (=the count of up-stemmed voices).

[15] Of course, *suonaeProximae* is only sensible in combination with *perAltitudinemCaputis*. In *perCaudaeSequentiam* the height of notes does not influence the CVO anyhow. See Section 5 for a survey on the sensible combinations of strategies.

$$\text{module perLineas.extrahere.duplex.}\boxed{\text{suonaeProximae}\ /\ \text{solumExplicitum}}\ .\ \boxed{X/Y}$$
$$\text{import Lmn.nota.perLineas.extrahere}[\mathsf{iseq}V \times \mathbb{N}_0]$$

$$\frac{K = \pi_1(p(1)) \quad v = \pi_2(p(1)) \quad k_{\square\in\{\uparrow,\perp,\downarrow\}} = \{k \in K \mid \pi_4(k) = \square\}}{\mathrm{initH}(p,v) = \begin{cases} \textbf{if } \#v \neq \#K & \textbf{then } \mathtt{error}(\textit{“Error in initial voice declaration”}) \\ \textbf{else if } \#k_\perp = 0 & \textbf{then } (v, \#k_\uparrow) \\ \textbf{else} & \mathtt{error}(\textit{“Nominally up-stemmed voices not visible.”}) \end{cases}}$$

$$\frac{\begin{array}{c} k_{\square\in\{\uparrow,\perp,\downarrow\}} = \{k \in k_{\mathrm{F}} \mid \pi_4(k) = \square\} \qquad \widehat{k} = \mathsf{anyPerm}(k_\uparrow) \frown \mathrm{sortH}_{\boxed{X/Y}}(k_\perp) \frown \mathsf{anyPerm}(k_\downarrow) \\ k_{\mathrm{U}} = \mathrm{sortH}_{\boxed{X/Y}}(\widehat{k}(\!|\{1..u\}|\!)) \qquad k_{\mathrm{D}} = \mathrm{sortH}_{\boxed{X/Y}}(k_{\mathrm{F}} \setminus \mathsf{ran}(k_{\mathrm{U}})) \end{array}}{\begin{array}{c} \mathbf{phase2}(m, v, k_{\mathrm{F}}, d, v_0, r_0, r, h) \\ = \begin{cases} \textbf{if } d = \langle\rangle & \textbf{then } \mathrm{assign}_{\mathrm{F}}(m, \mathit{squash}(h \rhd v), k_{\mathrm{U}} \frown k_{\mathrm{D}}, r, (h,u), v_0, r_0) \\ \textbf{else} & \mathrm{assign}_{\mathrm{D}}(m, \mathsf{ran}h \setminus v_0, \{\}, v_0 \setminus v, v, k_{\mathrm{U}} \frown k_{\mathrm{D}}, d, r, h, h(\!|\{0..u\}|\!), r_0) \end{cases} \end{array}}$$

$$\frac{\begin{array}{c} v_\uparrow = \{x \in v_0 \mid \pi_4(\mathit{last}(r(x))) = \uparrow\} \quad v_\downarrow = \{x \in v_0 \mid \pi_4(\mathit{last}(r(x))) = \downarrow\} \\ v_{\mathrm{U}} = h(\!|\{0..u\}|\!) \quad v_{\mathrm{D}} = (\mathsf{ran}\, h) \setminus v_{\mathrm{U}} \quad v'_{\mathrm{U}} = (v_{\mathrm{U}} \cup v_\uparrow) \setminus v_\downarrow \quad v'_{\mathrm{D}} = (v_{\mathrm{D}} \cup v_\downarrow) \setminus v_\uparrow \quad u' = \#v'_{\mathrm{U}} \end{array}}{\begin{array}{c} \mathbf{finalize}(m, r, (h,u), v_0, r_0) = (r, (\boxed{\mathrm{lastOrd}(r, v'_{\mathrm{U}}) \frown \mathrm{lastOrd}(r, v'_{\mathrm{D}})\ /} \\ \boxed{/\ \mathrm{rotate}(h, v_0, \#(v_{\mathrm{U}} \cap v_0), u, u') \mathbin{⨟} (\mathbf{ID}_{\mathbb{N}_1} \oplus ((\mathrm{lastOrd}(r_0, v_0)^{-1} \mathbin{⨟} \mathrm{lastOrd}(r, v_0))\ ))}, u')) \end{array}}$$

$$\mathrm{rotate} : \mathsf{iseq}V \times \mathbb{P}V \times \mathbb{N}_1 \times \mathbb{N}_1 \times \mathbb{N}_1 \to \mathsf{iseq}V$$
// CVO, active voices, number of active & orig. up-stem voices, old/new number of up-stem voices

$$\frac{\begin{array}{c} v_{\mathrm{P}} = (\mathit{squash}(h \rhd v))(w + u' - u) \qquad p = h^{-1}(v_{\mathrm{P}}) \quad \textit{// max index to rotate} \\ s = \textbf{if } u' > u \textbf{ then } \{(u+1)..p\} \lhd h \ \textbf{ else } \ \{p..u\} \lhd h \\ s' = \begin{cases} \textbf{if } u = u' & \textbf{then } \mathbf{ID}_V \\ \textbf{else if } u' > u & \textbf{then } \mathit{squash}(s)^{-1} \mathbin{⨟} (\mathit{squash}(s \rhd v) \frown \mathit{squash}(s \mathbin{⩥} v)) \\ \textbf{else} & \mathit{squash}(s)^{-1} \mathbin{⨟} (\mathit{squash}(s \mathbin{⩥} v) \frown \mathit{squash}(s \rhd v)) \end{cases} \end{array}}{\mathrm{rotate}(h, v, w, u, u') = h \mathbin{⨟} (\mathbf{ID}_V \oplus s')}$$

$$v \ \mathrm{aboveVK}_{r,U}\ k \iff (v \in U \wedge \pi_3(k) = \downarrow) \vee ((v \in U \vee \pi_3(k) \neq \uparrow) \wedge (\pi_2(\mathit{last}(r(v))) > \pi_2(k)))$$
$$v \ \mathrm{equalVK}_{r,U}\ k \iff (\pi_4(k) = \perp \vee (v \in U \iff \pi_4(k) = \uparrow)) \qquad \wedge (\pi_2(\mathit{last}(r(v))) = \pi_2(k))$$

Table 5. Parsing more than two voices with changing stem direction

Each voice is always treated either as “nominally up-stemmed” or “nominally down-stemmed”, even if the current note head (or pause symbol) does not carry a visible stem. The membership in these two groups is only altered if this is *unavoidable*, i.e. unambiguously indicated by the graphical input.

The algorithm in the modules *perLineas.extrahere.duplex. suonaeProximae/solumExplicitum* in Table 5 is basically the same as in *perLineas.extrahere.suonaeProximae/solumExplicitum* in Table 4. Main difference is that the criterion whether a note head is higher than a particular voice or higher than another note head is additionally affected by the stem direction: All up-stem voices and up-stem heads are infinitely higher than all down-stem voices and down-stem heads. This rule comes into play when sorting note heads, comparing note heads with voices (see the new definition of $\mathrm{aboveVK}_{\_,\_}$) and calculating the new CVO. For example, the physical variant *suonaeProximae* (see the left part of the last box in Table 5) extracts the physical order of all up-stem voices and appends that of the down-stemmed, so that physical crossings between both groups do not affect the CVO.

4.8 Voices changing the stem direction

The algorithm assigns the note heads to the active voices in the above-mentioned order: all up-stemmed precede all down-stemmed. But the numbers need not match: There can be more or less nominally up-stem voices in the currently active voices than there are up-stem note heads. The algorithm applies the minimal necessary change of direction assignments on the fly.

Only in case *solumExplicitum* special means must be taken by prepending the permutation delivered by rotate(..), see Figure 8: The y-axis means increasing pitch, with up-stem notes infinitely higher than down-stems; all horizontal lines are overlapping voices; the four eighth notes at score position T_1 (1 up and 3 down) shall proceed to the four eighth notes at T_2 (3 up and 1 down). There are explicit voice leading lines for v1 and v8 (see the solid lines), the proceedings of v4 and v6 follow from the algorithm (see the dotted

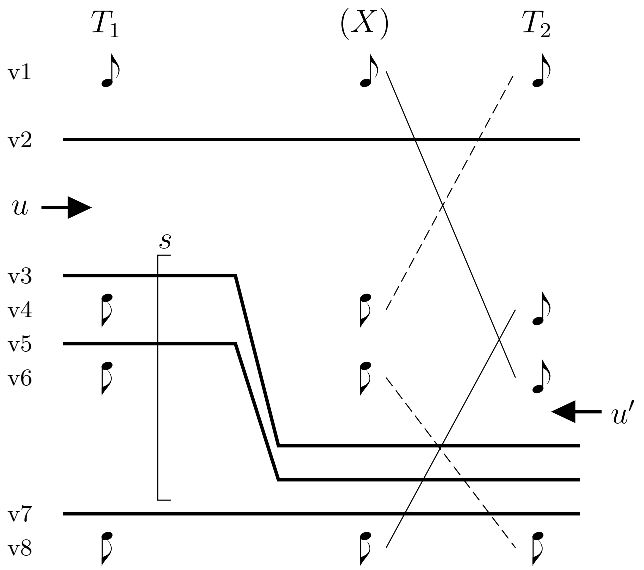


Figure 8. Preparatory rotation of voice names to enable the necessary change of stem directions

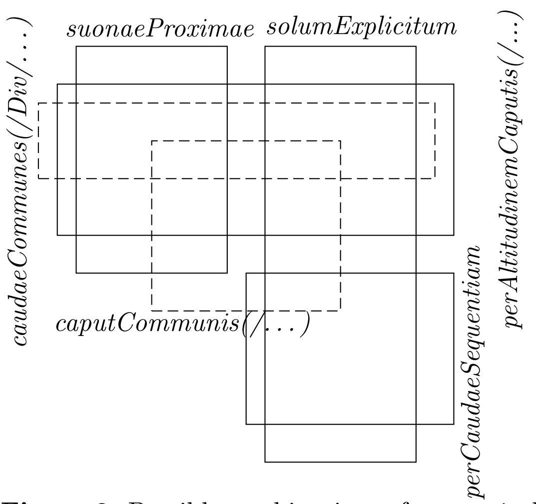


Figure 9. Possible combinations of *semantical* strategies (solid boxes) vs. graphical representations = *stylistic* variants (dashed boxes)

lines). The algorithm *perLineas.extrahere.solumExplicitum* simply applied the resulting permutation of the voice names to the CVO. But here the value u (= the number of up-stemmed voices) changes, and the lowest up-stemmed voice (v1 after the permutation) must precede the highest down-stemmed voice (v3) in the CVO. Therefore the graphically indicated permutation rotate(..) (as defined in Table 5) of all voices in $s = \{v3, v4, v5, v6\}$ is prepended. This leads to the intermediate situation (X). Now u can be replaced by u', changing the number of nominally up-stemmed voices, and the permutation of voice names can be applied as in the simple case.

5. GRAPHICAL APPEARANCE OF MULTI-VOICE STAVES. POSSIBLE COMBINATIONS OF STRATEGIES

The strategy *perLineas.caudaeCommunes* allows the note heads of different voices with the same score position, the same stem direction and the same head form to share the graphical representation of the stem. Frequently found is also the more lenient variant *perLineas.caudaeCommunes.div* allowing stem sharing for note heads of different forms (i.e. quarters and halfs), and the more restricted *perLineas.caudaeCommunes.idemPuncta* requiring the same head form and the same number of prolongation dots [16, pg.55]. While the parsing method *perAltitudinemCaputis* is used, these transformations are purely graphical and can be introduced or removed without changing the information contents. This is not longer true with *perAltitudinemCaputisVelSequentiam* and *perCaudaeSequentiam*. A further wide-spread restriction is *perLineas.caudaeCommunes.trabsCompleta*: If the stems are connected to a *beam*, than either all or none of the notes of the two voices under this beam share their stems.

Vice versa, *perLineas.caputCommunis* allows two voices with stems in different directions to share a note head, if score position, vertical position, duration (including prolongation dots) and accidentals are the same.

More lenient is */nota.vox.perLineam.caputCommunis .punctaMixta*, which allows different numbers of prolongation dots. This can be found in practice (Beethoven, piano sonata op14/1, 1.mvmt., m.7pp,[17]) but is not always accepted in text books ([16, pg.55, point(7)]).

Figure 9 shows the possible combinations of these graphical strategies with the different semantical strategies defined in this paper.

6. REFERENCES

[1] M. Lepper, M. Oehler, H. Kinzler, and B. Trancón, "Diminuendo al bottom —- clarifying the semantics of music notation by re-modeling," *plosOne*, 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0224688

[2] M. A. Cara, "Anticipation awareness and visual monitoring in reading contemporary music," *Musicae Scientiae*, vol. 22, no. 3, pp. 322–343, 2018.

[3] R. Kopiez, C. Weihs, U. Ligges, and J. I. Lee, "Classification of high and low achievers in a music sight-reading task," *Psychology of Music*, vol. 34, no. 1, pp. 5–26, 2006.

[4] D. Schön, J. L. Anton, M. Roth, and M. Besson, "An fMRI study of music sight-reading," *Neuroreport*, vol. 13, no. 17, pp. 2285–2289, 2002.

[5] A. Stenberg and I. Cross, "White spaces, music notation and the facilitation of sight-reading," *Scientific reports*, vol. 9, no. 1, p. 5299, 2019.

[6] M. Puurtinen, “Eye on music reading: A methodological review of studies from 1994 to 2017,” *Journal of Eye Movement Research*, vol. 11, no. 2, pp. 1–16, 2018.

[7] J. Kilian and H. H. Hoos, “Voice separation - A local optimization approach,” in *ISMIR 2002, 3rd International Conference on Music Information Retrieval, Paris, France, October 13-17, 2002, Proceedings*, 2002. [Online]. Available: http://ismir2002.ismir.net/proceedings/02-FP01-6.pdf

[8] N. Guiomard-Kagan, M. Giraud, R. Groult, and F. Levé, “Comparing voice and stream segmentation algorithms,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, M. Müller and F. Wiering, Eds., 2015, pp. 493–499. [Online]. Available: http://ismir2015.uma.es/articles/180_Paper.pdf

[9] E. Chew and X. Wu, “Separating voices in polyphonic music: A contig mapping approach,” in *Computer Music Modeling and Retrieval: Second International Symposium, CMMR 2004, Esbjerg, Denmark, May 26-29, 2004, Revised Papers*, ser. Lecture Notes in Computer Science, U. K. Wiil, Ed., vol. 3310. Springer, 2004, pp. 1–20. [Online]. Available: https://doi.org/10.1007/978-3-540-31807-1_1

[10] S. T. Madsen and G. Widmer, “Separating voices in MIDI,” in *ISMIR 2006, 7th International Conference on Music Information Retrieval, Victoria, Canada, 8-12 October 2006, Proceedings*, 2006, pp. 57–60.

[11] H. von Werden, *Musica Enchiriadis*. Bamberg: Staatsbibliothek, 1000, vol. Var. 1, https://zendsbb.digitale-sammlungen.de/db/0000/sbb00000078/images/index.html?id=00000078&seite=115&bibl=sbb.

[12] W. Tappolet, *La notation musicale et son influence sur la pratique de la musique du moyen age a nos jours*. Neuchâtel: Ed. de la Baconnière, 1947.

[13] J. M. Spivey, *The Z Notation: A reference manual*, ser. International Series in Computer Science. Prentice Hall, 1988.

[14] J. S. Bach, *Die Kunst der Fuge BWV 1080*. Wiesbaden: Breitkopf und Härtel, 1924, *edt.* W. Graeser.

[15] ——, *Das Wohltemperierte Klavier I BWV 846-869*. Wien: Wiener Urtext Edition, 1977, *edt.* Walter Denhard.

[16] K. Hader, *Aus der Werkstatt eines Notenstechers*. Wien: Waldheim-Eberle, 1948.

[17] L. van Beethoven, *Klaviersonanten I*. München: Henle Verlag, 1952, *edt.* B.A. Wallner.

A. MATHEMATICAL NOTATION

The employed mathematical notation is fairly standard, inspired by the Z notation [13]. For leaner notation, we add some overloading. Important constructs are:

$\mathbb{P}\ A$	Power set, the type of all subsets of the set A, incl. infinites.
$a \setminus b$	The set containing all elements of a which are not in b.
$A \times B$	The product type of two sets A and B, i.e. all pairs $\{c = (a, b) \mid a \in A \wedge b \in B\}$.
π_n	The nth component of a tuple.
$A \rightarrow B$	The type of the *total* functions from A to B.
$A \nrightarrow B$	The type of the *partial* functions from A to B.
$A \leftrightarrow B$	The type of the relations between A and B.
$a \mapsto b$	An element of a relation; simply another way to write (a, b)
$\mathsf{dom}\ a, \mathsf{ran}\ a$	Domain and range of a function or relation.
$S \lhd R$	$= R \cap (S \times \mathsf{ran}\ R)$, i.e. domain restriction of a relation.
$R \rhd S$	$= R \cap (\mathsf{dom} R \times S)$, i.e. range restriction of a relation.
$R \mathrel{⩥} S$	$= R \setminus (\mathsf{dom} R \times S)$, i.e. negative range restriction of a relation.
$f \llparenthesis s \rrparenthesis$	The image of set s under function or relation f
r^{-1}	The inverse of a relation
$\mathbf{ID}_A$	$= \{a \in A \bullet (a \mapsto a)\}$, the identity relation.
$r \fatsemi s$	The composition of two relations: the smallest relation s.t. $a\ r\ b \wedge b\ s\ c \Rightarrow a\ (r \fatsemi s)\ c$. (first apply r, then apply s)
$r \oplus s$	Overriding of function or relation r by s. Pairs from r are shadowed by pairs from s: $r \oplus s = \big(r \setminus (\mathsf{dom} s \times \mathsf{ran} r)\big) \cup s$
$\mathsf{seq}\ A$	The type of finite sequences from elements of A i.e. maps $\mathbb{N}_1 \nrightarrow A$ with a contiguous range $\{1..n\}$ as its domain.
$\mathsf{iseq}\ A$	The type of injective finite sequences from elements of A
$squash(a)$	Turns any partial function $\mathbb{N}_1 \nrightarrow A$ into a $\mathsf{seq}\ A$ by compactifying the indices.
$last(a)$	The last element in a sequence
$\langle\rangle$	The empty sequence.
$\alpha \frown \beta$	Concatenation of two lists.
$\#a$	The magnitude of a set (=number of contained elements).

Functions are considered as special relations, i.e. sets of pairs, like in “$f \cup g$”.

NEXUS: LIVE NOTATION AS A HYBRID COMPOSITION AND PERFORMANCE TOOL

Professor Sam Hayden
Trinity Laban Conservatoire of Music and Dance
s.hayden@trinitylaban.ac.uk

Professor Mieko Kanno
Sibelius Academy
University of the Arts Helsinki
mieko.kanno@uniarts.fi

ABSTRACT

The NEXUS live notation system, the latest product of the Hayden-Kanno collaboration, contrasts with their previous projects which utilised live DSP and synthesis. NEXUS is first discussed in the contexts of a comparison of Kanno's experience of performing solo violin works involving the live generation of music in both the audio and symbolic domains, and the affordances of Common Practice Notation for generative music. As with previous Hayden-Kanno projects, the main goal is the creation of a musical work which is fluid and spontaneous, both in its global form and specifics of detail, yet maintains a sonic consistency and coherence, but now in the symbolic domain. The implications of performer reading and interpretation for system design are explored. The second half of the paper outlines the main functions of the Max patch, how GMN code is generated for rendering as CPN in INScore during the performance, and, of the performer GUI which constrains the stochastic processes underlying the generation of specific musical parameters, general textual characteristics, and global formal shaping. The challenge was to formalize Hayden's compositional procedures so the generated notations retain a musical identity and interest, whilst leaving space for Kanno's interpretative decisions and being technically simple enough to be sight-readable. The uses of the system for hybrid performance and compositional applications are discussed, and some directions for future development.

1. INTRODUCTION

1.1 The *NEXUS* project: antecedents

As part of an ongoing collaboration involving the artistic potentials and *affordances* (after Gibson) [1] of new music technologies, composer Sam Hayden and violinist Mieko Kanno extend their practice-based research into the area of 'live notation', used as both a real-time composition and performance tool. Their previous Arts and Humanities Research Council (AHRC, United Kingdom) funded research collaborations resulted in two works for electric violin (*Violectra*[1]) and interactive computer: *schismatics II* (2010) and *Adaptations* (2011) [2]. *schismatics II* involved seven movements of fixed notation (in a fixed order) with real-time DSP. *Adaptations* involved various short modules of fixed notation performed in an indeterminate order, also with real-time DSP, to which the computer *adapted* to the performance using Nick Collins' machine listening and learning system (ll~ object) [3]. The patches were designed to run autonomously but can be used as a digital 'instrument' with input from a (human) performer. Given Hayden and Kanno work primarily with Common Practice Notation (CPN), a domain which, according to Legard and Morgan '…remains a necessary symbolic language for composers to communicate their intentions to performers'[4], the *NEXUS* project takes Hayden and Kanno's collaboration in a new direction, the emphasis being on the live generation of CPN, where previously it was live audio processing. As a result of having worked in a composer-performer collaboration for more than 10 years, this project allows detailed consideration of where the boundaries between composer and performer lie in an artistic practice that uses technology-mediated CPN.

2. AESTHETIC COMPARISONS: DIFFERENT APPROACHES TO LIVE GENERATION OF SONIC/SYMBOLIC MATERIAL

2.1 New perspectives on live notation?

One might legitimately ask what 'new' perspectives are on offer from the *NEXUS* project to the field of live notation. The technology is well-established (Max + INScore), as are the use of generative algorithms such as Markov Chains and other stochastic methods. The aesthetic goal of having the notation be different in every performance yet maintain a coherent identity, and the common issues and solutions around live notation (e.g. making the music easy to sight-read by imposing constraints and combining pre-generated and live material) are also familiar territory.

[1] Violectra is a range of custom made, skeletal-frame, electric violins, violas, cellos, and vio-basses, designed and made by David Bruce Johnson (Birmingham, UK, since 1992); http://violectra.co.uk/violin/

Nevertheless, we consider that there are three key questions at the heart of the project that require exploration:

- Why we find the combination of CPN and generative music an attractive creative option.
- Which are the effects of *reading* and *performing,* i.e. symbolic interpretation, on the part of the performer, on the system's aesthetic function(s).
- What are the risks to which this system exposes Hayden and Kanno, in terms of their established professional practice.

Our answers to these questions shed light on the nature and significance of this project, both in terms of the development of the Max programming and the actualisation of the work as performance; an examination of the relationship between system and performance offers a unique perspective to the live notation research communities.

2.2 Common Practice Notation (CPN)

In order to examine the creative potentials of Common Practice Notation (CPN) and generative music, it is useful to first discuss the musical affordances of CPN per se. While 'sight-reading' frequently refers to the reading 'on-the-fly' of CPN, the use of notation in music improvisation (technology-mediated or otherwise) involves a much wider range, including images, graphics, text and other metaphorical stimuli. It is often assumed by practitioners that in a live performance setting involving the *reading* of material, the use of non-standard notation is more creative and 'open' than CPN, engendering more freedom and promoting more creativity and structural coherence by the provision of 'cues'. Conversely, the reading of CPN is often assumed to involve less creativity and be more prescriptive, promoting (more) exact reproduction of the written properties by its commonly understood specific rules for reading, with its requirement for expert skill in sight-reading. Yet, CPN has been used to promote more varied readings, precisely because of its widespread use for many centuries in Western art music. Harris emphasises the importance of the quasi-linguistic understanding of CPN's representational aspects (emphasis by the authors):

'Music reading depends on <u>understanding the language</u>, instantly recognizing the symbols and knowing exactly what they mean. You need to know the different keys, spot recurring melodic patterns, really understand how rhythms go and develop an instinct for fingering.' [5]

The musical 'language' in traditional Western art music consists of varied parameters, including pitch, rhythm, key, speed, phrasing, character, mood, timbre, historical style, genre-specific style, instrument-specific idioms, expressive rhetoric, technical ease, and others. Sight-reading triggers a whole set of *learned skills* automatically, combining all of the information pertaining these parameters that come with this rich language. The sight-reading skill has much in common with a style recognition skill traditionally required as part of a tonmeister training where learners identify a compositional style at sight, from an open page of an unknown score.

The sight-reading skill of professional performing musicians allows them to read off more information than that which is given on the page (or screen). They infer, anticipate, and guess. The gathering of an excess amount of information from the visual information, and making it available in performance, is at the heart of excellent sight-reading skill. Music notation, in this sense, is designed as stimuli to inspire spontaneity. You may ask: what about the level of specificity that is a feature of CPN? The question points to what this specificity is *for*, otherwise, in music performance. While one-to-one correspondence is important, one of the purposes of CPN is to communicate the music to the performer, as oppose to communicating the written properties that make up the music. Much like other notation types, it functions to inspire the reader/performer's imagination. Paradoxically, the specificity of the information in CPN has the capacity to trigger a vast range of references beyond written information itself, precisely because of its specificity.

An approach towards music notation as a triggering interface for the performer's imagination may seem overly utilitarian. Yet, an excellent capacity to read off the page a varied range of information both visible and invisible and set it in motion, is a hallmark of good musicianship. In discussing scores in CPN, pianist Ian Pace comments that he prefers "to see scores as the means for channelling performers' creative imagination in otherwise unavailable directions, rather than as an obstacle [to limit the imagination]" [6]. Kanno and Hayden also view using CPN with generative algorithms as the means for channelling performers' and composers' creative imaginations. We wish to take advantage of the wealth of references that CPN is capable of communicating, as well as of our expertise in handling it as composer and performer in the context of the new creative affordances of live notation.

2.3 The effects of *reading* and *performing*

The practice of reading CPN makes it clear that there is a temporal as well as conceptual hiatus between *reading* and *performing*, a situation that remains in the digital domain. Much like poetry reading, reading aloud is a performance unlike silent reading which is devoid of communication potential. Live notation involving human performers is never precisely 'live' in a sense of temporal simultaneity between the display and performing. There is always a time-lag, however small that may be. There are instructive comparisons to be made with generative musical works involving solo violin, performed by Kanno, where computer-mediation enters the symbolic domain (including CPN). We see performer interpretation taking a more significant role in terms of the *structural* actualisation of the work. The works described below deploy this hiatus between reading and performing using original approaches.

Georg Hajdu's *Ivresse '84* for violin, laptop quartet and electronic conductor (2007), a piece which involves computer mediation in *both* the sonic and symbolic domains, is a case in point, a piece that Kanno has performed in collaboration with the composer. From the point of view of the violinist, various notational fragments (Hajdu calls them 'measures') taken from the first of Cage's *Freeman Etudes* (1977-80) [7] are recombined algorithmically during the performance, in combination with audio samples from other performances and spoken text from a recorded interview the composer undertook with violinist János Négyesy in 2007. Hajdu describes the notation:

'For each of the 20 sections, a stochastic process chooses among a range of measures and recombines them into a new structure, which is sight-read by the performer. (This approach, of course, assumes familiarity with the material.)' [8]

In a significant sense, the notation *is* 'live', in as much as the stochastic ordering of the notational fragments generated during each performance is necessarily unique. However, the material is not sight-read in a literal sense, given Kanno (in this case) pre-prepares the Cage extracts in advance of the performance, even if the specific ordering that will occur is not known. She half-recognizes the material as it appears during performance, knowing the language, but not necessarily exactly where the particular fragment comes from within the *Etude*. She retains a level of familiarity with the material, yet has a necessarily context-specific interpretative flexibility (e.g. how fast/slow, what pauses in between, whether to 'phrase' over some fragments, etc). It gives a different *meaning* to how time passes within this music: Hajdu's live notation system influences how she listens to the ongoing sonic landscape and how she may contribute to it structurally. Compositionally, what the violinist plays is designed to influence the laptop quartet and the other sonic materials, and to have a significant impact on the formal actualisation. In both Hayden's *NEXUS* and Hajdu's *Ivresse '84*, the act of reading and performing the score (i.e. the symbolic aspects of the piece) by a live performer reduces the hierarchical primacy of the system as a structural determinant of the performance; placing much more responsibility with the live violinist who performs the 'language' of the given CPN by connecting disparate notated musical objects into a coherent formal shape.

Hajdu's example is a case of live generation of familiar material. A further comparison can be made between works involving real-time generation of the notation *during* the (live) performance and live performances of scores that are completely algorithmically generated but the scores' material exists *before* the performance (so the material is pre-prepared). The latter type of notation engages the performer in much the same way as most fixed notation scores do, except in the sense that the score can be generated anew for every performance occasion. In this sense, the generated scores are 'performances' of the system which is the work: such conceptual framework introduces a new perspective for interpretation by the performer.

Michael Edwards' *hyperboles 2* for violin and live electronics (2015) exists in an interesting position *between* live notation and fixed score [9]. The score is generated *prior* to each performance, leaving time for the soloist to pre-prepare the work as if it were a fixed score, but for that specific performance only. Here, Edwards stretches the temporal displacement between algorithmic generation and live actualisation. *Hyperboles 2* presents the option of notational re-generation by Kanno herself, through a user interface, by setting criteria (in domains such as the 'terrain', duration, and some pitch characteristics), but again, this is something that is done *prior* to the performance. In practice, Kanno regenerates scores for *hyperboles 2* as many times as she needs, until she finds a score that she likes to play, for each performance occasion. The duration is a critical parameter in decision-making; for example, as she considers a one-hour performance requiring different material to a 15-minutes performance, she changes the parameters in the user interface, until she finds a satisfactory result. Her decision-making is informed by the expressive potential in performance that she observes in the generated notation. In other words, her reading of the language expressed in the notation informs her performing potential.

Repeated generations of scores provides another insight for the performer. Kanno starts to see a correspondence between the chosen parameters and scores in terms of expressive potential. Here, we mean general 'musical' aspects such as the character, atmosphere, and overall mood of the music rather than specific material properties. She recognises a musical *style,* in addition to a compositional style, resulting from the structural choice. This foresight knowledge is crucial in dealing with works with multiple generations. Andrew R. Brown also observes, while discussing his work *Appearances*, how programming constraints on the generative systems enable familiarity with the material for the performer, even if specific details are unique (an idea that Hayden embedded in the design of *NEXUS* and was observed by Kanno as the user):

'…because the nature of the generated music is tightly constrained, the more familiar the performer is with the processes of the algorithm, either as a result of analysis, explanation or experience performing it, the more comfortable they become with the stochastic nature of the work.' [10]

2.4 Comparing formal paradigms: the performer + live audio / live notation + the performer

In Hayden's own *schismatics II* for e-violin and computer (2010), the computer-processed live audio, resulting from the performance of a fixed (common practice) notated score, is based on stochastic algorithms and machine listening to control live sampling, envelope-based sample playback, convolution, delays and other live DSP modules, combined to produce a computer-generated *doppelgänger* which shadows the sound of the live violin. *schismatics II* uses a model long-established by many works developed at IRCAM, such as Boulez's *Anthèmes 2* for violin and live electronics (1997), which involves a fixed score with prepared electronics which are live-triggered, and also live-processed electronics, using real-time score-following techniques [11]. Another example in this category is James Wood's *Autumn Voices* for violin and electronics (2001), a BBC commission written for Kanno, whose sonic landscape is based on the spectral analysis of violin recordings of Kanno [12]. In this case, pre-prepared electronics are live-triggered and spatialized by the composer in combination with the live violin sound.[2]

This category of the live violin + live audio has a number of common characteristics. First, the outcome varies from one occasion to another while retaining the majority of the details consistent across all possible performances. Second, the uniqueness of the outcome at each performance occasion lies for the most part in the electronics and *not* on the part of the performer. This second point is significant when considered the role of the violin sound in these compositions: the interactivity on the part of the violinist is limited, even though the violin sound itself is central to the conception and production of the electronic part. For example, the performer's role in *schismatics II* is limited to making subtle adjustments of balance and timing in their reading of a fixed notated object in relation to the live computer part: interactivity is more a feature of the computer's relationship to the soloist than vice versa. In this sense, the interactivity is an additional feature in the established practice of electroacoustic music, which gives nuance to the outline though not challenging the role of the composer as the provider of the sonic design, and that of the performer as the interpreter of this design.

When the 'reading' is centred on computer-generated notation, interpretation becomes inexorably tied to the generative aspects of the work and system design. The nature of what such systems do is transformed and reconfigured as the computer becomes an *active* site for performer interpretation, as opposed to the more *reactive* role it plays in *schismatics II*, requiring the performer to realise entirely the sound of a generative work. The computer is no longer assigned the role of generating an actualised sonification of data, instead offering a symbolic 'proposal' to the performer to 'complete' a sonic object whose outline is suggested by the *NEXUS* system. The notation is necessarily incomplete, Kanno not only having many choices of musical details (e.g. tempo, articulation and dynamics) of any particular generated notation, but also choices which affect the overall formal contouring of the piece during the performance, where her anticipation of the larger-scale parametric changes in phrase-length, phrase direction, register, density of events and so on feedback into her 'on-the-fly' interpretative decisions. This aesthetic 'incompleteness' in the symbolic domain has fundamental implications for the conception of the programming: the goal is no longer to create a coherent sonified digital musical 'agent', as if equivalent to a human musician; rather the goal becomes the creation of a symbolic musical space with enough information to *enable* the performer to actualise both the details and the formal trajectory of the work as a performance. The system is designed to present a suggestion of a musical structure, but one where there is space left (what Hayden terms a 'symbolic deficit', see section 4.4) for real-time performative decisions. Otherwise, there would be no need for a human performer and a MIDI, or other synthesised output would suffice.

This category of live notation + the performer brings some risks to both the composer and performer of Western art music. The composer hands over control of a significant portion of actual material outcomes to the generative system and performer. The performer, on the other hand, may have limited access to their 'toolkit' that makes up their usual professional practice. For example, the generated notations may call for very little actions. In other words, the notations may trigger very little learned (instrumental) skills. It then challenges the performer's musicianship to find a musical solution on the spot, which is also a skill of an experienced performer. The risks describe the conceptual challenges on the norms of composition and performance in Western art music. This is because the experimental practice we conduct involves learning, but more significantly, some 'unlearning' of the norms and standards that have been fundamental in our respective careers as professional composer and performer. The shift from the category of the performer + live audio to that of live notation + the performer seems ultimately to correspond to the shift from 'additive learning' experience to that of 'unlearning in order to learn'. It has involved revision of what Hayden and Kanno take for granted, respectively as composition and performance.

2.5 Live notation: implications for the system

The combination of live-generated *and* pre-prepared materials is an option open to the performer using the *NEXUS* system in order to practically enable more complex (pre-prepared) material and simpler (sight-readable) materials, the aesthetic benefit being more variety in the material, and more variety of strategy within a performance. The fact that live generation occurs at a higher level of symbolic abstraction beyond the direct sonification of live DSP (actually *two* levels of abstraction higher, given the patch first generates GMN code, which is then rendered as CPN in INScore) has implications for system design and aesthetic functions. The system is primarily concerned with the splicing of strings of text (the building-blocks of GMN code). As such, this process represents a step 'out of time',

[2] Kanno's other collaborations include John Hails' *La Pastora* for violin and live electronics (2007), the computer functioning as a complex delay system, where linear 'found' material originating in folk song is stretched and presented canonically using combinations of precise numerical ratios [13]; Dimitris Papageorgiu's *deti* (2017), composed as part of Kanno's AHRC-funded project 'Modelling a virtual violin' [14], which uses live-generated DSP and triggered samples (according to the 'scrolling score' principle, where samples are triggered at certain points in the notated score on a sequencer timeline) [15], the notational system focussing on parametric specifications of physical actions (often involving the 'decoupling' of left-hand and right-hand violin techniques); and Stylianos Dimou's *For Violectra* (2018) which combines fixed notation with live electronics, featuring sophisticated real-time granulation of the violin timbres [16].

however brief, between data generation and sonification by the live performer, a temporal displacement that is exaggerated in Michael Edwards' works. In the *NEXUS* system, Hayden perceives a clearer mediated 'trace' of his compositional intentions in the performance results than is the case in *schismatics II*: the constraints of the symbolic domain were shaped entirely by Hayden's programming decisions which were all, in effect, aesthetic compositional decisions, as opposed to being the sonic result of stochastic combinations of pre-existing live synthesis methods.

Whether notated material is generated live, pre-prepared, or somewhere in between, the implications to the system are that many more parametric decisions in the symbolic domain, usually taken by the composer, are handed to the performer. As the *NEXUS* system is designed to create notational objects that are necessarily incomplete, Kanno is required to view the performance potentials of such decisions, mediated through her own instrumental technique, embedded experience and sedimented performance histories. The specific sonic realisation of a generated notation, the anticipation of its place in a larger-scale form (see section 3.6), its formal (dis)connection with previous/subsequent notations and so on, are examples of compositional decisions and local/global criteria that Kanno is constantly aware of during performance, aspects that the system is designed to enable rather than determines.

3. *NEXUS*: THE PROTOTYPE

3.1 The *NEXUS* project: context

Begun in 2016, the *NEXUS* project uses Max with INScore to generate score fragments algorithmically in real-time, to be sight-read by Kanno during the performance. An earlier Max6 prototype was presented to the SOUND WORK seminar (Orpheus Instituut, Ghent, 2016) [17]. The current Max7 version involves no DSP, existing only in the symbolic domain. There has been extensive research into the many applications of real-time synthesis and DSP in artistic practices, to which Hayden and Kanno have already contributed. Dominique Fober observes that real-time algorithmic composition using *notation,* or other symbolic musical representations (as opposed to interactive/generative systems which use MIDI, real-time synthesis and machine learning e.g. Karlheinz Essl's algorithmic music generator *Lexikon-Sonate* [18], George Lewis' interactive improvisation system *Voyager* [19], Thor Magnusson's IXI software [20] and IRCAM's OMAX system [21]), remains comparatively under-explored in interactive music:

'Today, new technologies allow for real-time interaction and processing of musical, sound and gestural information. But the symbolic dimension of the music is generally excluded from the interaction scheme.' [22]

Indicative examples of compositions utilising live notation range from the SuperCollider-generated CPN of Richard Hoadley's *triggered* [23] to the generative graphic scores of Andrea Valle's *Dispacci dal fronte interno* [24]. Various interactive notation-based composition systems exist, including the Bach project [25] and Maxscore [26], both Max-based applications, and computer-assisted composition environments such as Opusmodus [27], OpenMusic [28], Escore [29] and the Active Notation System [30]. The goal of the *NEXUS* project is to create a system that 'works' as a *performance* outcome and is not only a demonstration of digital music techniques or a music programming investigation. The *aesthetic* aim is the creation of a live *notated* musical work which is fluid and spontaneous, both in its specifics of detail and global form, yet maintains a sonic consistency and identity. This was also the aim of *schismatics II* where the computer-processed live *audio* was always different within defined constraints; a coherent sound world with different specifics of detail and a consistent global form [31].

Hayden's motivation for the *NEXUS* project originated in his speculations about the extent to which his compositional ideas in the 'fixed' symbolic domain of CPN could be mediated by generative music technologies, to be 'completed' by Kanno's interpretation, and still result in a coherent (co-)authored work. *NEXUS* represents a formalisation, in simplified form, of Hayden's recent compositional methods, such as interpolations between atonality and diatonicism, inharmonicity and harmonicity, rapid gestures and stasis; the use of stochastic rhythms and pitch sequences, and large-scale formal contouring involving progressions of density and register. *NEXUS* is both a *compositional* output from Hayden and a performative output from Kanno, but one where the 'territories' of composition and performance are conceived in a different way, redefining what Hayden and Kanno do.

3.2 *NEXUS* and GUIDO Music Notation (GMN)

The prototype *NEXUS* patch live-generates and combines event-lists of pitch, duration and register information with randomly selected GMN 'tags' [32] representing standard CPN aspects (e.g. meter, clefs, beams, dynamics, and some articulation classes). It involved more string formatting (via the *sprintf* object) than was undertaken in previous Hayden-Kanno projects, in order to generate complete lines of GMN code, rendered as CPN in INScore. Figure 1 is an example of GMN code with its associated CPN.

```
/ITL/scene/myscore set gmn "[\\meter<\"4/8\"> \\clef<\"g\">
\\beam ( \\slur ( g1/8 c1/16 e1/16 a&0/16 c&1/8 c1/16 ))]"
```

Figure 1. GMN to CPN example

Given the *NEXUS* system does not use the more metaphorical notations often utilised in pieces involving generative scores, such as images, graphics or text, there is a more direct (less arbitrary) representational relationship between the generative aspects - what is rendered from

GMN - in terms of properties: what the performer sees as CPN, what is played by the performer, and what the listener hears in performance.

3.3 The performer Graphical User Interface

The *NEXUS* system functions somewhere between an instrument/performance system and a composing tool. It has a modular construction which generates live notational fragments of varying lengths and complexity, a process triggered by the performer, using the GUI (see Figure 2).

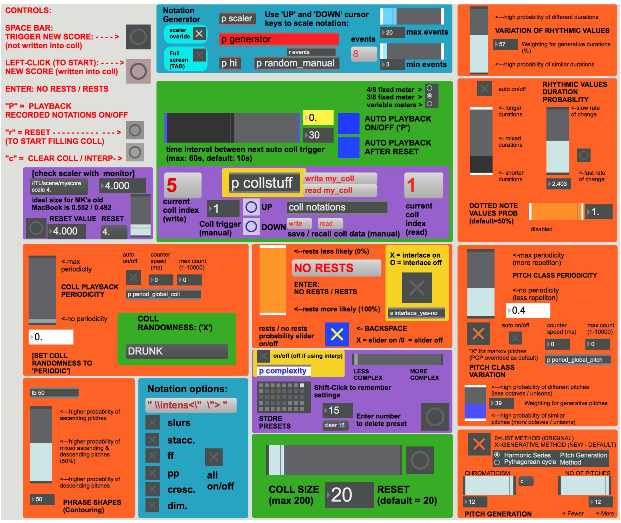


Figure 2. Prototype performer GUI (detail)

The performer can alter the GUI settings to influence various musical parameters:

- *Pitch generation method*
- *Rhythm generation method*
- *Variation of registers*
- *Variation and periodicity of pitch classes*
- *Variations of rhythmic values (length, similarity)*
- *Probability of rests and dotted rhythms*
- *Phrase shape, contour and phrase direction*
- *Max/min number of 'events' ('event' = rest or note)*
- *Interpolation between 'initial' and 'target' settings.*

3.4 Generative musical parameters: pitch-classes

Underlying pre-compositional decisions were involved in the patch design: e.g. a selection of pre-composed pitch-class sets (messages containing pre-defined numerical lists) were built into the initial programming to facilitate the controlled random generation of 12-TET pitch materials. A later development was the addition of the option to generate pitch-sets algorithmically using Markov Chains via some abstractions from Essl's Real Time Composition Library (RTC-lib 7.1) [33], making selections from the harmonic series or the Pythagorean cycle, facilitating pitch-fields on a continuum between chromatic atonality and quasi-diatonicism (see Figure 3). Markov Chains can also be applied to the domains of rhythm and register, according to user choice via the performer GUI.

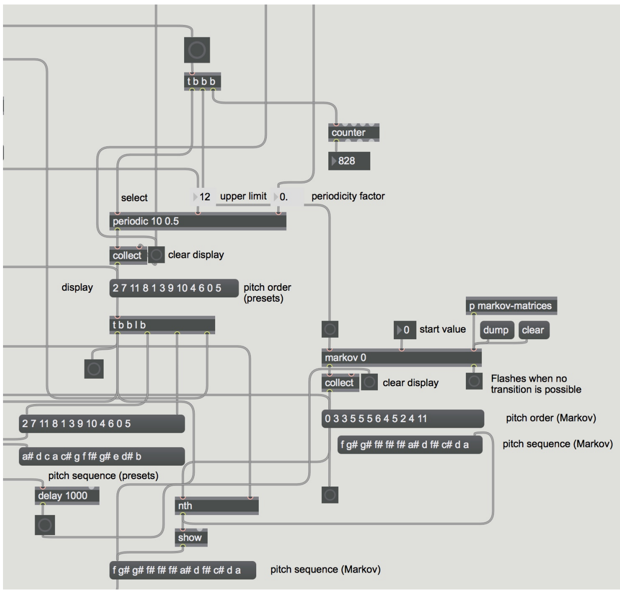


Figure 3. Pitch-class set generation sub-patch (detail)

3.5 Generative musical parameters: duration

To make the live-generated music sight-readable, Hayden limited the possible set of subdivisions of the beat to simple duplet ratios (no tuplets), although dotted rhythms are possible, and the semiquaver (16th note) is the shortest possible rhythmical unit. A duration series is generated with the pitch-sets in a sub-patch called 'generator' (see Figure 4). Using stochastic processes, the patch then splices these pitch-sets and rhythmical series together into an event list, distributing events across registers according to the GUI controls. This sequence can be interleaved with another randomized series of rests or pitches. The event list forms the basis of a line of GMN code, excepting the 'tags' added at the end of the generative process.

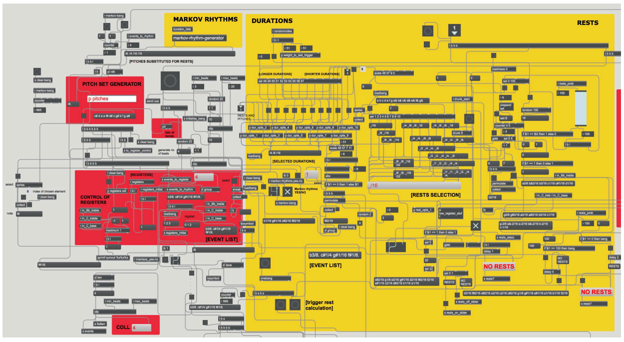


Figure 4. 'Generator' sub-patch (detail)

3.6 Saving generated notations: the coll

A useful feature of the system, with both performative *and* compositional applications, is that it can save and recall generated materials, using the *coll* object. Figure 5 shows an example of a coll ('notations') within which each complete line of GNM code is stored and recalled. The automated control of timings triggering the recall of saved materials can be adjusted, or, saved notations can be recalled manually in combination with newly generated materials.

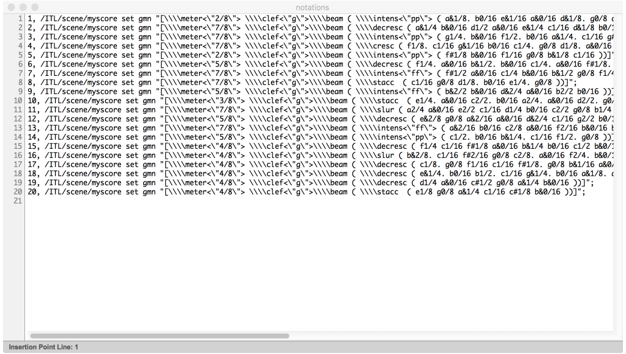


Figure 5. Coll of GMN code (detail)

These functions arose from Kanno's desire to have a mixture of familiar and new notations when interpret-ing the material on-the-fly. Hayden and Kanno became aware of the interpretive distinction between live-gen-eration (of not-so-new material) and live-notation (of newly generated material). They use this variety in or-der to produce multiple levels of attention on the part of the performer. To provide performance flexibility, an option was added to splice 'simple' (i.e. live-generated) and 'complex' (i.e. pre-saved) colls into a single coll, in order to combine simpler (live) materials and more com-plex (pre-prepared) materials (see section 2.5). The splic-ing of coll indices can either be randomised, or, the order of generation preserved if the interpolation system is used (see section 3.7). This was deemed necessary to utilise fully the expressive skills of the performer and realise more fully the compositional potentials of the system (see section 4).

3.7 Global formal shaping: interpolation functions

Most musical parameters defined in the patch can either be given fixed (numerical) values or can transform gradually *within* each successively generated notation (single coll index). However, creating effective larger-scale formal transformations *between* successive coll indices has been an important development of this project. To achieve this end, a *pattr* system was implemented to interpolate values between user-defined 'initial' and 'target' global settings (see Figure 6). Almost every musical parameter (in numerical form), as defined on the GUI, is connected to the *pattr* system so a linear, exponential or user-drawn table interpolation can be selected, across a user-defined coll size, to give a large-scale transformation of multiple parameters simultaneously. Michael Edward's *hyperboles 2* (see section 2.3), has parallels with Hayden's *NEXUS* system, although Kanno's potential compositional interventions in *hyperboles 2* affect more global criteria. When performing *NEXUS,* Kanno can also influence the local detail of the music more directly (as well as global formal shaping), through the various parametric controls available on the GUI connected to the interpolation system. This aspect of the system crucially enables Kanno to *anticipate* the likely global direction and formal contouring according to the chosen parameters and interpolation type and adjust her performance of each notation accordingly.

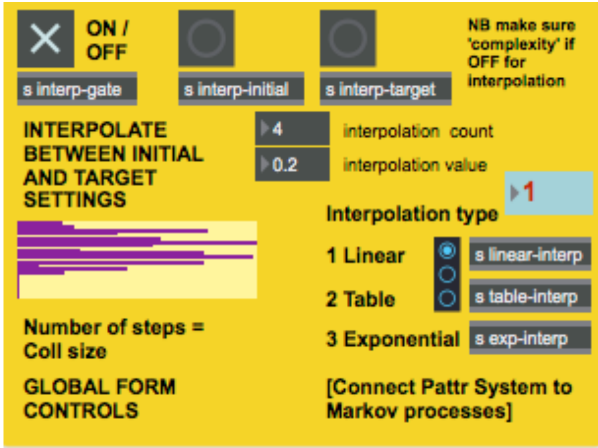


Figure 6. Interpolation controls in main GUI

4. *NEXUS*: CHALLENGES AND SOLUTIONS

4.1 The function(s) of 'live' generation in making the piece: why 'live' notation?

The following section outlines some of the practical challenges and solutions in response to questions raised so far in the development and uses of the *NEXUS* system. An innovation afforded by live notation systems is that the performer has the means to change the generative parameters of a notated composition *during* its performance, while remaining in the symbolic domain. Nevertheless, 'live-ness' in computer music is an oft-debated point by John Croft amongst others [34]. The extent to which the 'real-time' generative approach is 'live' is a moot point, discussed at length by Simon Emmerson [35]. It is not 'in-time' in the sense that the production of the musical data and its sonification by the human performer are *not* simultaneous when material is generated in the symbolic, not audio, domain. What can be said for certain is this approach can guarantee the uniqueness of each performance (although colls can be saved and recalled: see section 3.6), and the uniqueness of performer responses, which is nevertheless based on defined musical objects resulting from concrete constraints. The question of the identity of the *NEXUS*-generated piece and/or performance inevitably arises, and whether the computer should be regarded as an instrument or a compositional tool (or both): computer-mediated 'live' notation somewhat deconstructs this traditional binary division in Western art music. The answer to this question lies in the artistic *uses* of the system. Part of the aesthetic of the piece (in fact, an infinite set of pieces), is that each performance is unique, created by defined algorithms which have infinite variations yet are highly constrained. Many of the generative processes are automated versions of Hayden's 'out of time' formalised compositional methods when writing fixed scores but are simplified to enable sight-reading. Likewise, through experience, it has become something akin to an instrumental extension for Kanno as she learns to anticipate the general effects of changing certain parameters as a means of directing the live performance. The idea of a *hybrid* compositional and performance tool is therefore apt.

4.2 How is notation generated so that the system creates the symbolic material Hayden-Kanno want to use to make music?

Notation is one of the most influential communication tools in classical music, and the knowledge and skills embedded in its use amongst composers and performers remain significant. Hayden and Kanno are interested in how the computer could complement creatively their existing expertise in their handling of notation, to the extent that the relationship between live computer-generated CPN and human performer interpretation intrinsic to the project expands their understanding of the potential of 'text' in music. The *NEXUS* system has been developed iteratively, after feedback from Kanno's experimentation, gradually constraining the parameters until notated material emerges that is both musically convincing and sight-readable (constraint of register was an important factor). Programming decisions are also aesthetic decisions, affecting directly the notational outcomes of the system. Yet the material is necessarily incomplete (Hayden's 'symbolic deficit'), in particular regarding timbre and articulation, and 'needs' the interpretation of a human performer to 'complete' its transformation into sonic musical material.

The affordances of Max-enabled live notation for the performer's interpretative spontaneity are the main points of investigation, as is finding a useful definition of what Hayden and Kanno mean by 'musical' in this digitally-mediated context. The point at which the outcomes can be regarded as 'music' is an aesthetic judgement: e.g. Hayden's use of Markov Chains, for more weighted probabilities in relation to pitch, rhythm and register, has significantly helped to achieve this 'musicality' from the points of view of composer and performer alike. Hayden and Kanno observed that the use of Markov Chains reinforces the linear melodic character of the generated material by suggesting a sequential direction, thus helping Kanno to create phrasing (what Essl calls 'controlled randomness', where the sonification of weighted probabilities means less general scattering of pitches) [36]. The use of Markov Chains to control registers made the generated notations more 'musical', given the importance for the violin of register for melodic shape, phrasing and tone quality, in the sense of having more differentiation and perceived cumulative flow, whilst being less predictable in overall character. There is a need for a certain balance between predictability and surprise in the generated material, in order for it to bear some sense of 'musicality'. With too much predictability, the composition-performance becomes a pastiche exercise; if there is too little predictability, it becomes too random and reduces Kanno's role to merely being a translator. The generated notation has to 'give' something, i.e. 'speak' to the performer, enabling and inviting them to make sense of it, which in turn requires the performer to have a certain familiarity with the visual 'language': it cannot be *completely* different each time.

4.3 How do computer-generated notation and performer interpretation contribute to 'expressivity'?

In the process of computer-generated CPN fragments becoming musically 'expressive' through performance interpretation, determining the contributions of the patch and of the performer is not a straightforward question. To begin to answer this, Kanno makes an important distinction between 'properties' and 'behaviours' of musical material; 'properties' are the combined statistical/numerical processes that generate the material (the various parameter sliders and settings on the GUI and associated internal algorithms), whereas 'behaviours' are the perception of the nature of the musical object or entity itself, in totality or *gestalt*. As a performer, Kanno is less worried about *how* the musical object has been generated, but more focusing on *what* is generated. From a composition point of view, this is relatable to a distinction between generative *processes* (multi-variable algorithms internal to the patch) and generative *results* (the notation *as a musical entity*).

Kanno and Hayden consider the potential multiple action possibilities arising from notation material (one definition of 'expressiveness') as an important aspect of this project. They are concerned mostly with the material's 'behaviour', more than its 'property'. The statistical properties of the material determine possible 'behaviours', but 'behaviours' themselves have so much more 'expressiveness' than the material (or its said 'property'). During the early stages of the project, Kanno thought she was going to select (or find a rule for selecting) action possibilities from notation (in a Cage-like procedure). However, what Kanno does now is to recognise and amplify material 'behaviours' observable within notations, an aspect that the interpolation system has significantly enhanced. Kanno selects or gives perspective to simultaneously appearing 'behaviours' according to how the performance is going.

Excepting 'minimal' music, the *results* of compositional processes are usually heard, as opposed to direct perception of the processes themselves. When performing, musicians don't usually think about *how* the material has been made, but more *what* the material is and how to interpret it expressively/musically. The *NEXUS* project makes Kanno consider *compositional* parameters more than she usually would when interpreting fixed notations, given she can manipulate what are more usually called *compositional* parameters at any point. Many decisions are deliberately left to her, including choices of tempi, details of dynamics and articulation classes, as well as the (dis)continuity between successive generated notations. It is not Hayden and Kanno's aspiration for the system to become an autonomous virtual composer nor performer, so the notational outputs from the system are necessarily incomplete.

4.4 The symbolic deficit: what is the relationship between generated notation and live interpretation?

Given the generative CPN material is deliberately lacking in detail, the 'symbolic deficit' means that much of the sonification decisions are in the hands of Kanno's musicianship. Her focus is to shape the material, make it into a performance that 'works' and is in some way 'musical', relying heavily on performance intuition and experience. Much of the performer's decision-making will be around parameters not defined by the programming: phrasing, timbre, articulation, tempo, intermediate dynamics (beyond ff and pp) and so on. There is a controlled quasi-improvised process taking place where Kanno reacts to the notations in the moment: the system is an invitation for performance. On the one hand, the CPN was necessarily simplified from Hayden's usual practice (his fixed scores are *much* more complex) whilst still embedding something of his compositional ideas in the programming. On the other, it gives Kanno the ability to influence the outcome of a performance that is nevertheless very constrained, given the composer-defined limits on what can happen.

4.5 Symbolic generation and interpretation: 'in-time' or 'out-of-time' actualisation?

To use Xenakis' famous distinction, there are both 'in-time' and 'out-of-time' applications of the *NEXUS* system [37]. Although we have focused on a performance use which is as close to being as 'in-time', 'live' or 'real-time' as possible (minimum time-lag between generation and interpretation), one can use the system to generate material *any* time before the performance, an 'out-of-time' compositional application. One could, as Michael Edwards does with his 'slippery chicken' software, generate notation algorithmically, to create a fixed score to be rehearsed in preparation for a later performance [38]. Each generation of the piece would still be unique so it would be a valid approach as long as that version was not repeated. Hayden and Kanno *decided* to use the *NEXUS* prototype to combine pre-prepared and live-generated materials for variety of output and performance strategy, but this is not a given. These are aesthetic decisions, adopting a more 'experimentalist' approach which requires performance uniqueness and unpredictable (but not unknown) outcomes: it is the *controlled* randomness that interests Hayden and Kanno, in between the aesthetics of Cage and Xenakis.

5. FUTURE DEVELOPMENTS

5.1 Initial phase of the project

The first phase of the project has been a technical consolidation of the current Max7 prototype, which focused on improving the efficiency and effectiveness of the generation of the GMN data rendered as notation in INScore, the underlying stochastic mechanisms which generate the GMN data, the GUI design, organization of sub-patching, functioning of parametric controls, and fixing fundamental notation issues (e.g. beaming, groupings of rests and beats, dotted rhythms, clefs and transposition, visual formatting, completion of incomplete bars with rests and so on).

5.2 Current phase of the project

The current phase of the project is optimizing the *musical* affordances of the GUI and generated CPN *for the performer*, finding solutions to the issues of maintaining *musical* interest and variety, while retaining playability, through combinations of sight-readable (live) and more complex (pre-prepared) materials. This has included implementing more variability within the idea of 'complexity' from a perceptual point of view, more variety in the possible lengths and characters of generative notations, more efficient handling of larger-scale global formal control of musical parameters (via the interpolation system), and a more flexible generative approach to pitch-class set and duration series creation, selection and succession (via Markov Chains). The next part of this phase will be the notational implementation of violin-specific techniques (e.g. double-stops, harmonics, microtones, and articulations).

5.3 Final phase of the project

The final phase of the project will be to make the *NEXUS* system interactive by implementing the real-time analysis of sonic descriptors, e.g. timbral and temporal features of the live violin signal, by using Max externals such as Nick Collins' ll~ object (see section 1), Tristan Jehan's library (pitch~, loudness~, brightness~, noisiness~, bark~, analyzer~, shifter~, segment~, beat~) [39], the iana~ object (Todor Todoroff) [40], the yin~ object (Norbert Schnell, implementing the Cheveigné and Kawara model) [41], the *FTM/Gabor* object library (IRCAM) [42] [43], fiddle~ and bonk~ objects (Miller Puckette) [44], or the *Zsa.Descriptors* libraries (Mikhail Malt and Emmanuel Jourdan) [45], in order for the computer to influence decisions about the generation of future notations in a feedback situation. We are also considering adding some live DSP, so the patch generates sound which is related in pitch, rhythm and timbre to; (a) the generated notation; and/or (b) the played sound from the live violin, as counterpoint to the live violin sound itself. A further development could be to implement a network connection to enable the coordination of multiple generative notations between different computers, allowing an ensemble of live musicians to use the system, whether synchronized or in free time.

6. CONCLUSION

The project has raised fundamental questions about the past, present, and future significance of CPN in technologically-mediated composition and performance. The exploration inherent in the development of the *NEXUS* system has involved: (a) an interrogation of the relationship between the underlying technical aspects of the system design and (pre)compositional programming decisions that determine the processes of how material is generated in the symbolic musical domain; (b) the evaluation of resultant generated CPN fragments as a symbolic language appropriate for making music; and (c) how such generated symbolic material becomes musically *expressive* through performer reading and the implications of interpretation for system design. The current phase of the project has focused on two technical developments: (1) implementing higher-level interpolation controls to enable larger-scale formal transformations; and (2) implementing a more musical control of linear contouring via Markov Chains. Such ‘higher-level controls’ shape the global transformation of the notated materials through the gradual interpolation between user-defined ‘initial’ and ‘target’ states (or presets), where previously, successive colls were separate, discrete and non-connected entities. This necessitated the finer tuning and calibration of multiple control parameters. The more musical control of linear contouring, phrase direction and phrase shaping was achieved by mapping Markov Chains to successions of pitch-classes, rhythmic units and registers. This proved to be a more flexible and musically intuitive approach to the (performer-defined) constraints enacted upon the algorithmic generation of materials. We continue to investigate controlling the larger-scale transformation of such constraints over time in relation to perceived ‘musical’ performance *outcomes*. Within these processes, it has been fundamental to calibrate the system to identify enough space in the generated CPN to enable Kanno to do something *interpretative* so that the final work engages her skills and musicianship, rendering a live performance as more than the sum of its parts.

7. REFERENCES

[1] J. J. Gibson, *The Ecological Approach to Visual Perception*. Houghton Mifflin Harcourt, 1979, Boston.

[2] S. Hayden and M. Kanno, “Towards Musical Interaction: Sam Hayden's Compositions for E-Violin and Computer”, in *Contemporary Music Review*, 2013, vol. 32, no. 5, pp. 485-498.

[3] N. Collins, “LL: Listening and Learning in an Interactive Improvisation System,” *Listening and Learning System* (ll~), 2010. https://composerprogrammer.com/re-search/ll.pdf

[4] P. Legard and N. Morgan, “Re-conceptualizing Performance with ‘Active’ Notation”, in *Proc. from Interdisciplinary Centre for Computer Music Research seminar*, p.1, 2007. https://www.academia.edu/1640744/Reconceptualizing_Performance_with_Active_Notation

[5] P. Harris, *Improve your sight-reading!* London: Faber Music, 2008, p.4.

[6] I. Pace, “The New State of Play in Performance Studies,” in *Music and Letters*, 2017, p. 285. Also quoted in F. J. Schuiling, “Notation Cultures: Towards an Ethno-musicology of Notation,” in *Journal of the Royal Musical Association*, 2019, p.432.

[7] J. Cage, *Freeman Etudes* – Books 1/2 for Violin solo (Etudes 1-16), 1977-1980.

[8] G. Hajdu, *Ivresse ’84* for violin, laptop quartet and electronic conductor, 2007. http://georghajdu.de/compositions/ivresse-84/

[9] M. Edwards, *hyperboles 2* for violin and computer, 2015. https://michael-edwards.org/work.php

[10] A. R. Brown, “Generative Music in Live Performance,” *Australasian Computer Music Conference*, Brisbane, Australia, ACMA, 2005, p. 24.

[11] P. Boulez, *Anthèmes 2* for violin and live electronics, 1997. https://www.universaledition.com/pierre-boulez-88/works/anthemes-2-1150

[12] J. Wood, *Autumn Voices* for violin and live electronics, 2001. http://dro.dur.ac.uk/375/

[13] J. Hails, *La Pastora* for violin and live electronics, Composition Portfolio, Written Commentary (Durham University), 2007, pp. 34-39. http://etheses.dur.ac.uk/1322/1/1322.pdf

[14] M. Kanno, “Modelling a virtual violin,” AHRC project ref: AH/D501784/1 (RCS), 2015-16. https://gtr.ukri.org/projects?ref=AH/N003691/1

[15] D. Papageorgiu, *deti* for e-violin and electronics, 2017. https://soundcloud.com/dimitris_papageorgiou/deti

[16] S. Dimou, *For Violectra* for e-violin and electronics, 2018. https://stylianosdimou.com/music/for-violectra-2018

[17] SOUND WORK: *Composition as Critical Technical Practice*, Orpheus Instituut, Ghent, 2016. https://orpheusinstituut.be/en/news-and-events/sound-work-composition-as-critical-technical-practice

[18] K. Essl, *Lexikon-Sonate*: infinite realtime composition for computer-controlled piano, 1992-2016. http://www.essl.at/works/Lexikon-Sonate.html

[19] G. E. Lewis, “Too Many Notes: Complexity and Culture in Voyager,” in *Leonardo Music Journal*, vol. 10, 2000, pp. 33-39.

[20] T. Magnusson, “The IXI Lang: a Supercollider Parasite for Live Coding,” in *Proc. of ICMC*, Huddersfield, 2011. http://www.ixi-software.net/thor/ixi_lang.pdf

[21] G. Assayag, M. Chemillier and G. Bloch, OMAX Software Environment, IRCAM, 2018. http://repmus.ircam.fr/omax/home

[22] D. Fober, Y. Orlarey and S. Letz, “INScore: An Environment for the Design of Live Music Scores,” in *Proc. of the Linux Audio Conference*, CCRMA, Stanford University, 2015, p.47.

[23] R. Hoadley, *triggered* (for various performers) with experimental sculptural performance interfaces, 2011. http://rhoadley.net/comp/triggered/

[24] A. Valle, *A Dispacci dal fronte interno* for feedback system including ad libitum strings, printer and live electronics, 2012. https://vimeo.com/55360995

[25] A. Agostini and D. Ghisi, Bach – Computer-Aided Composition in Max, 2016. http://www.bachproject.net/

[26] G. Hajdu and N. Didkovsky, "MaxScore: Current State of the Art," in Proc. of ICMC, Ljubljana, 2012, pp. 156-162.

[27] N. Morgan and J. Podrazik, Opusmodus 2.0 – Parametric Music Composition System, 2020. https://opusmo-dus.com/

[28] C. Agon, G. Assayag and J. Bresson, OpenMusic – Visual programming environment for Computer Assisted Composition (IRCAM: Music Representation research group), 1998-2018. http://forumnet.ircam.fr/product/openmusic-en/

[29] M. Alcorn and C. McCelland, "Escore – Real-time Notation in Interactive and Live Electronic Performance Environments," in *Proceedings of the Live Algorithms Group seminars,* 2005.

[30] P. Legard and N. Morgan, "Re-conceptualizing Performance with 'Active' Notation," in *Proceedings from Interdisciplinary Centre for Computer Music Research seminar*, 2007.

[31] S. Hayden and M. Kanno, "Towards Musical Interaction: Sam Hayden's *schismatics* for e-violin and computer (2007, rev. 2010)," in *Proc. of ICMC*, Huddersfield, 2011, pp. 486-490.

[32] H. Holger and K. Hamel, The Guido Music Notation Format, Version 1.0, 2020. http://guidolib.sourceforge.net/doc/GUIDO-Music Notation Format.html

[33] K. Essl, Real Time Composition Library (RTC-lib 7.1), 2016. http://www.essl.at/works/rtc.html

[34] J. Croft, "These on liveness," *in Organised Sound,* vol. 12/1, 2007, pp. 59-66.

[35] S. Emmerson, "Live versus real-time," in *Contemporary Music Review,* 1994, vol. 10/2, 95-101.

[36] K. Essl, "Algorithmic Composition," in *Cambridge Companion to Electronic Music*, N. Collins and J. d'Escriván (eds.), Cambridge: CUP, 2007, p.123.

[37] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition*, second, revised English edition, NY: Pendragon Press, 2001, pp.180-200.

[38] M. Edwards, "An Introduction to Slippery Chicken," 2012. http://www.michael-edwards.org/sc/media/sc-paper-short.pdf

[39] T. Jehan, *Creating Music by Listening*, PhD Thesis in Media Arts and Sciences, Massachusetts Institute of Technology, September 2005.

[40] T. Todoroff, E. Daubresse and J. Fineberg, "IANA: A Real-Time Environment for Analysis and Extraction of Frequency Components of Complex Orchestral Sounds and its Application within a Musical Realization," in *Proc. of ICMC*, 1995, Banff, Canada, pp.292-293.

[41] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," in *Journal of the Acoustical Society of America*, 2002, vol. 111, pp. 1917-1930.

[42] N. Schnell et al., "FTM Complex Data Structures for Max/MSP," in *Proc. of ICMC*, 2005, Barcelona, Spain, 2005.

[43] N. Schnell, and D. Schwarz, "Gabor, multi-representation realtime analysis/synthesis," in *Proc. of the Int. Conference on Digital Audio Effects* (DAFx'05), Madrid, Spain, 2005.

[44] M. Puckette and T. Apel, "Real-time audio analysis tools for Pd and MSP," in *Proc. of ICMC*, 1998, pp. 109-112.

[45] M. Malt and E. Jourdan, "Zsa.Descriptors: a library for real-time descriptors analysis," in *5th Sound and Music Computing Conference*, Berlin, Germany, 2008. pp.134-137.

REPRESENTATION OF HARMONIES ON THE HARMONIC WHEEL

Luis Nuño

ITACA Institute
Polytechnic University of Valencia (Spain)
lnuno@dcom.upv.es

ABSTRACT

The Tonnetz is a useful tool for representing musical excerpts or full pieces containing mainly major and minor triads. However, when a musical composition contains dissonant triads or higher-order chords, it can only give a limited representation of it. The Harmonic Wheel is a physical tool that combines a Tonnetz transformed into a polar grid with a plastic disc containing the lines that define the major, harmonic and melodic minor scales, together with the scale degrees and the symbols of the corresponding seventh chords. This way, it allows to represent a large variety of musical works, including both triads and seventh chords, as well as to find the chords that are common to different keys. To show its main characteristics and advantages, several examples are given from different musical styles. In all cases, the representations obtained are simple and compact, and therefore easy to memorize, which makes the Harmonic Wheel a powerful and versatile tool for analyzing and composing music, as well as providing an efficient mnemonic notation.

1. INTRODUCTION

The Tonnetz is a graphic representation of musical notes and their consonance relationships, that is, the consonant intervals (perfect fifth, major and minor thirds) and the consonant triads (major and minor) formed by them. There are two relevant representations of it: the Oettingen/Riemann and the Douthett and Steinbach, which are dual [1, 2]. The Oettingen/Riemann Tonnetz (Figure 1) is a triangular lattice, where the notes are at the vertices and the consonant triads on the triangles, while the Douthett and Steinbach's Tonnetz (also called Chicken-Wire) is a hexagonal lattice, where the notes are on the hexagons and the consonant triads at the vertices. In both cases, the edges represent the consonant intervals, which in turn define (in different ways) the P, L and R operations, which stand for parallel, leading-tone exchange and relative, respectively. They are defined for consonant triads and, for example, P maps C major to C minor, L maps C major to E minor and R maps C major to A minor, and vice versa [1].

Both Tonnetze are infinite in a plane, that is, in a 2-dimensional (2D) space. However, they are periodic in 3 directions, so, by choosing any 2 of them, we can obtain an alternative representation on a torus, which is a finite surface in a 3-dimensional (3D) space.

Generalizations of the Tonnetz to include other (dissonant) intervals and trichords or higher-order chords, such as tetra- or pentachords, lead to more complex geometries, which require 3 or more space dimensions [3 – 6].

Both the analysis and composition of some kinds of musical pieces can be greatly simplified by representing their notes and harmonies on a Tonnetz, mainly when they only include major and minor triads. Logically, 2D graphs are simpler and easier to use than those requiring 3 or more space dimensions.

In this respect, some examples of representing musical excerpts on a 2D Tonnetz are given in [7], which only contain major and minor triads. They consist of binary and ternary combinations of PLR operations and correspond to nineteenth century music by Brahms, Schubert, Beethoven, Verdi and Wagner. Other examples, from mid-twentieth century Jazz and Latin repertoire, are given in [8], where most harmonies are seventh chords. Therefore, in order to represent them on a 2D Tonnetz, the seventh was omitted in the dominant seventh chords and the root was omitted in the half-diminished chords.

The Harmonic Wheel [9] is a practical 2D Tonnetz, where one of the axes is re-oriented so that the notes of a major key form a rectangle, thus resulting in a full rectangular grid, which is then transformed into a polar one. The final graph is an annulus, which is finite in a plane, thus keeping the advantages of both the planar and the toroidal Tonnetze. Additionally, the regions corresponding to the major, harmonic and melodic minor scales, together with the scale degrees and the seventh chords associated to them, are also indicated, which facilitates the representation of harmonies of tonal pieces.

To show its main characteristics and advantages, first a couple of examples on diatonic modulation are given, which deal with finding the pivot chords. Then, the harmonies of five musical excerpts and pieces are represented on the Harmonic Wheel: an excerpt by Beethoven included in [7], a well-known tonal song, a Coltrane's composition included in [8] (but represented there on a chromatic circle instead of a Tonnetz), a piece whose harmonies are based on Béla Bartók's axes, and a song including modulations.

In all cases, the representations obtained are simple and compact, and therefore easy to memorize, which makes this representation system an efficient mnemonic notation. It is worth pointing out that some of the examples here presented are, in some cases, studied in the 12 keys, so having such a mnemonic notation is greatly helpful. In fact, the Harmonic Wheel, together with other similar tools, are part of a subject of a Master on Music and Scenic Arts in a Doctorate Program.

2. HARMONIC WHEEL

Figure 1 shows the Oettingen/Riemann Tonnetz, where the notes are assigned to the vertices and the major and minor triads to the triangles. The horizontal lines represent the perfect fifths and the two oblique lines the major and minor thirds. Additionally, a region containing the 12 major and 12 minor triads just once is marked with a dashed line.

In Figure 2, the notes in that region belonging to the C major key are marked with circles, except the tonic, which is marked with a rectangle. Then, the two oblique sides are re-oriented to become vertical (Figure 3), so that the notes of a major key form a rectangle (the 3 notes of the C major key outside the rectangle are in fact repeated on it) and, therefore, the whole grid becomes rectangular. Furthermore, the 6 consonant triads belonging to the C major key are inside the rectangle and the C major and A minor triads are in its centre. As well, each pair of relative triads forms a smaller rectangle, which is assigned the corresponding key signature. This way, each triad also represents the centre of a major or natural minor key.

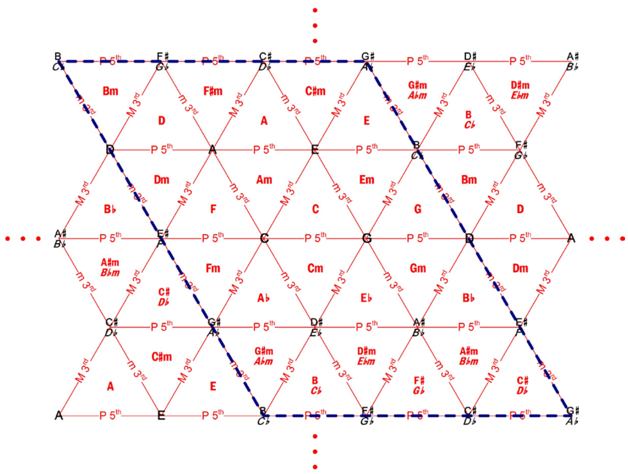


Figure 1. Oettingen/Riemann Tonnetz and a region with the 12 major and 12 minor triads.

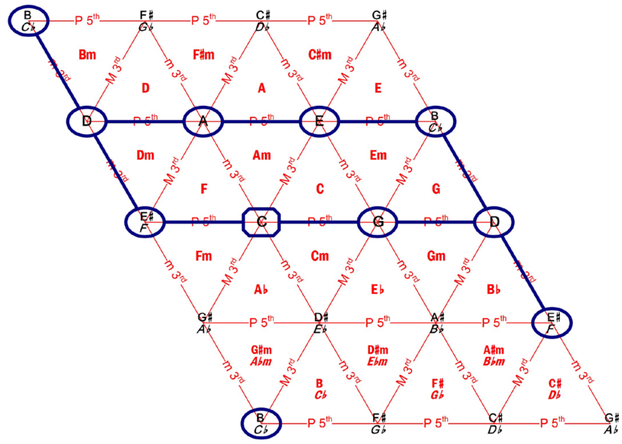


Figure 2. The C Major key on the Oettingen/Riemann Tonnetz.

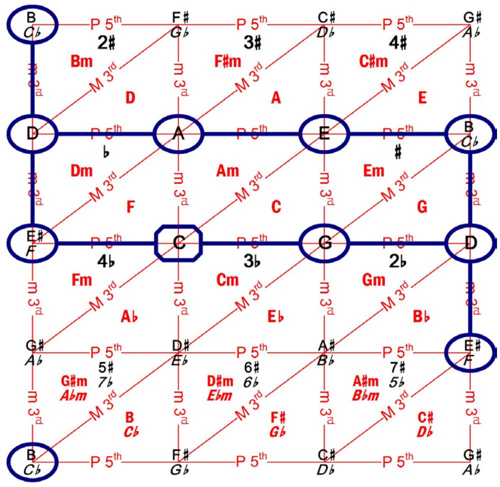


Figure 3. The C Major key on a rectangular grid.

In Figure 3, the notes at the bottom are repeated at the top, which means that this diagram is cyclic in the vertical direction. In contrast, obtaining a cycle in the horizontal direction requires to add more notes and triads until completing a cycle of fifths.

If we do so and then curve the diagram to make it circular, the result is the Harmonic Wheel, shown in Figure 4. This way, the rectangular grid is transformed into a polar one, so that the horizontal lines turn into circumferences, the vertical lines into radii and the diagonal lines into spirals. Consequently, the 3 types of cycles are now as follows: a closed cycle on each circumference, containing 12 perfect fifths (or fourths); an open cycle along each radius, containing 4 minor thirds; and an open cycle on each spiral, containing 3 major thirds. The first two cycles are clearly seen on the graph, while the last one is not so evident, and this is the reason why it was chosen with the least number of intervals (3).

In practice, the Harmonic Wheel is a physical tool consisting of two rotating discs: one cardboard, with the full polar grid printed on it (including the notes, interval lines, triads and key signatures, in black and red colours), and the other a transparent plastic, with the lines defining a major key printed on it (in blue). The two discs are joined together at their centres with a rivet, which allows selecting any major or natural minor key, together with its corresponding major and minor triads. To compare this tool with a 3D Tonnetz, Figure 5 shows a handmade toroidal Tonnetz, where the blue lines correspond to a major key (in this case, E♭ major). The difficulties for using it in practice are apparent.

Returning to the Harmonic Wheel, the scale degrees and the seventh-chord symbols are also printed on the plastic disc in blue (Figure 6). And, because the major scale region only takes up one fourth of that disc, in the final design two other fourths are utilized to print the corresponding lines, scale degrees and seventh-chord symbols for the harmonic and melodic minor scales, while the other fourth remains free.

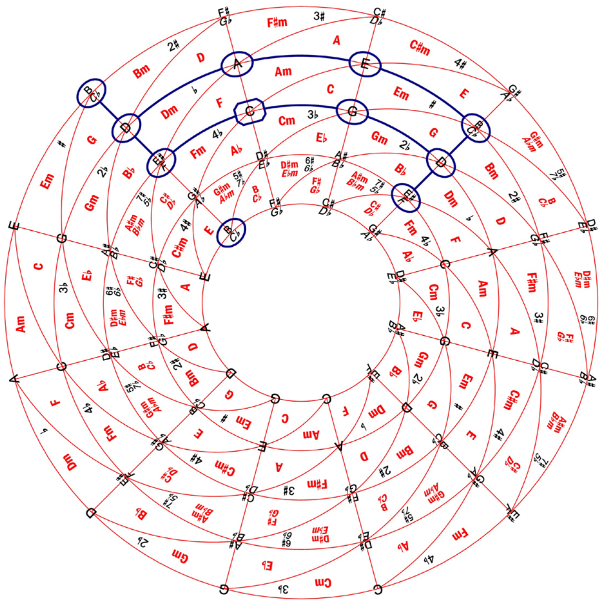


Figure 4. Harmonic Wheel and the C Major key (polar grid).

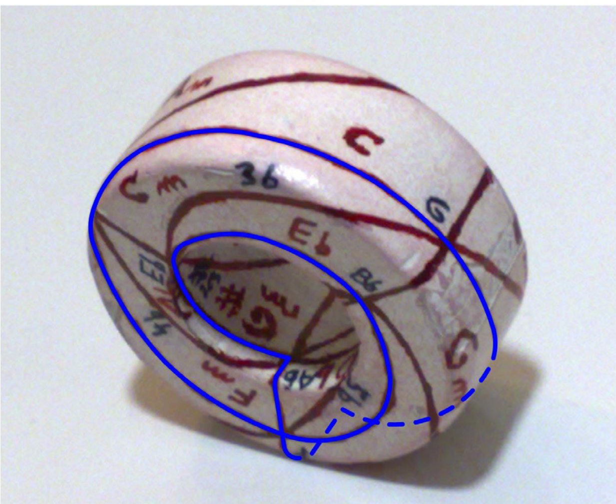


Figure 5. Handmade toroidal Tonnetz and the E♭ Major key.

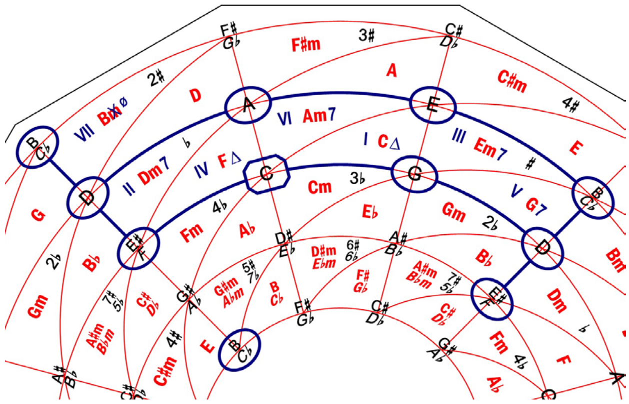


Figure 6. Scale degrees and seventh chords of C Major on the Harmonic Wheel.

Logically, the triads associated to each degree of those scale types are obtained by simply omitting the sevenths in the seventh chords. This includes not only major and minor triads, but also augmented and diminished.

The addition of the plastic disc to the polar grid, which allows selecting any major or minor scale (natural, harmonic or melodic) with its corresponding scale degrees, triads and seventh chords, makes this tool a powerful and versatile resource for analyzing and composing a variety of musical styles. To show it, some examples are given in section 4, where, for simplicity, only the bare diagram and the major scale will be used.

3. INTERVALS AND CHORDS

The red lines in both the rectangular and polar grids represent the consonant intervals, that is, the perfect fifths, the major and the minor thirds (as well as their inversions). The rest of intervals, which are dissonant, are the semitone, the tone and the tritone. Every interval belongs to an "interval class" characterized by its minimum number of semitones (considering the given interval and its inversion within an octave), so there are 6 interval classes, named 2-1 to 2-6 after Forte [10]. Here, we will represent them by icn, n being the number of semitones, which ranges from 1 to 6. Interval classes $ic3$, $ic4$ and $ic5$ are directly represented on the grids, whereas $ic1$, $ic2$ and $ic6$ can be expressed as a combination of two of the previous ones by

$$ic1 = ic4 - ic3 = ic5 - ic4 \tag{1}$$

$$ic2 = ic5 - ic3 = -2 \cdot ic5 \tag{2}$$

$$ic6 = 2 \cdot ic3 \tag{3}$$

Additionally, the following equation holds:

$$ic3 + ic4 + ic5 = 0 \tag{4}$$

which means that the sum of the 3 consonant intervals gives a closed line, which in turn defines a triangular area corresponding to a consonant triad, major or minor. All these equations have been represented graphically in Figure 7, where, for simplicity, a rectangular grid was considered. Of course, the corresponding representations on a polar grid are easy to visualize mentally.

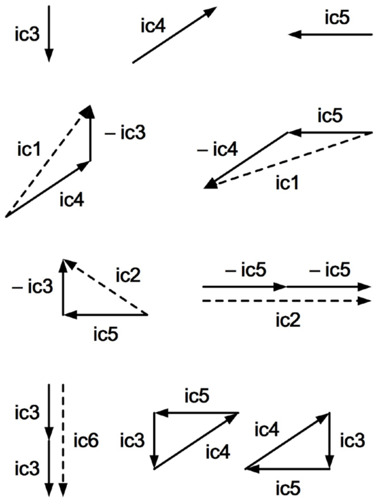


Figure 7. Interval classes on a rectangular grid.

In pitch-class set theory, every "set class" (a generalization of chord type) is assigned a Forte name consisting of two numbers separated by a hyphen, the first one corresponding to the "cardinality" (the number of "pitch-classes" or notes in the set class) and the second to an ordinal. For example, a diminished triad (3 pitch-classes) is named 3-10, and a minor seventh chord (4 pitch-classes) 4-26.

A practical way to describe the structure of a "pitch-class set" (a particular chord) is by means of the "intervallic form" [11], which is the sequence of intervals (in semitones) between every two adjacent pitch classes, including the interval between the last and the first ones. For example, the intervallic form of a major chord, such as C major, is $\{435\}$, because the intervals between its adjacent notes (C E G) are 4, 3, and 5 semitones (the latter being the interval from G to C). The circular shifts of this intervallic form, which are $\{354\}$ and $\{543\}$, also correspond to the same chord (but starting from a different note). For a minor chord, the intervallic form is $\{345\}$ or any of its circular shifts. As it is equal to the intervallic form of a major chord, but in reverse order, a minor chord is said to be the "inversion" of a major chord, and they two form a set class, whose Forte name is 3-11. So, in order to distinguish between them, a letter "a" or "b" can be added to the Forte name (a general criterion for assigning them is provided in [11]). As a last example, the intervallic form of a diminished triad (set class 3-10) is $\{336\}$. Since it is "inversionally symmetrical", no letter will be assigned to it.

To relate chords to intervals, for set classes with 3 or more pitch-classes (as trichords and tetrachords), an "interval-class vector" is defined, which has 6 components that list the number of times each interval class ($ic1$ to $ic6$) is contained in the given set class. For example, the interval-class vector of a diminished triad is (002001), because it contains 2 interval-classes $ic3$ and one $ic6$, and the interval-class vector of an augmented triad is (000300), as it contains 3 interval-classes $ic4$.

Now, a question arises: which are the most common trichords and tetrachords? Regarding the common practice period (around 1650 to 1900), the harmonies are mainly built by superimposing thirds on the 7 degrees of the major, harmonic and melodic minor scales [12, 13]. This leads to the 4 basic triads and the 7 basic seventh chords, which correspond to set classes 3-10, 3-11a, 3-11b, 3-12, and 4-19a, 4-19b, 4-20, 4-26, 4-27a, 4-27b, 4-28, respectively. In addition, the augmented sixth chords give rise to the 3-8a (Italian) and 4-25 (French). Table 1 shows all trichords from 3-8 to 3-12 and Table 2 all tetrachords from 4-19 to 4-28. They include the symbols here used to represent them, the intervallic forms (starting from the root) and the interval-class vectors. For other musical styles, such as Pop, Latin or Jazz, all harmonies in these tables are quite common. A useful list of most common chords in these styles is given in [14].

Figures 8 and 9 give the graphic representations of those harmonies (on a rectangular grid), where the roots are marked with circles and the dashed lines represent tritones ($ic6$).

Trichord	Symbol	IF	ICV
3-8a	7*	462	010101
3-8b	Ø**	642	010101
3-9	sus4	525	010020
3-10	dim	336	002001
3-11a	m	345	001110
3-11b	M	435	001110
3-12	+	444	000300

Table 1. Characteristics of trichords 3-8 to 3-12. An asterisk (*) means "omit 5" and a double asterisk (**) "omit ♭3". A major chord (3-11b) is usually represented by the root without any symbol. IF: Intervallic Form, starting from the root. ICV: Interval-Class Vector.

Tetra-chord	Sym-bol	IF	ICV	Basic Trichords
4-19a	mΔ	3441	101310	m, +
4-19b	Δ♯5	4431	101310	M, +
4-20	Δ	4341	101220	m, M
4-21	9*	2262	030201	7*, Ø**
4-22a	(9)	2235	021120	sus4, M
4-22b	m4	3225	021120	sus4, m
4-23	7sus	5232	021030	2 x 4sus
4-24	7♯5	4422	020301	7*, Ø**, +
4-25	7♭5	4242	020202	2 x 7*, 2 x Ø**
4-26	m7	3432	012120	m, M
4-27a	Ø	3342	012111	Ø**, dim, m
4-27b	7	4332	012111	7*, dim, M
4-28	O	3333	004002	4 x dim

Table 2. Characteristics of tetrachords 4-19 to 4-28. An asterisk (*) means "omit 5" and a double asterisk (**) "omit ♭3". Symbol "(9)" means "add 9", whereas symbol "9" adds both the minor seventh and the ninth. IF: Intervallic Form, starting from the root. ICV: Interval-Class Vector.

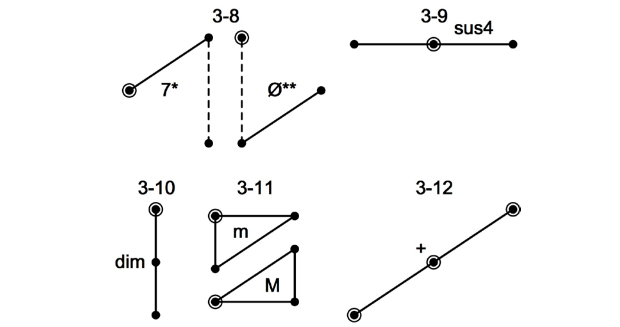


Figure 8. Trichords 3-8 to 3-12 on a rectangular grid.

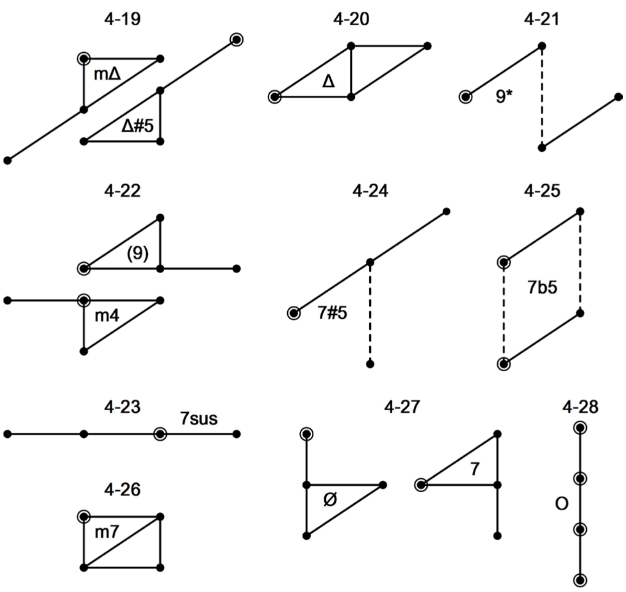


Figure 9. Tetrachords 4-19 to 4-28 on a rectangular grid.

The last column of Table 2 shows which and how many trichords from Table 1 are contained in each tetrachord, which can be easily visualized by comparing Figures 8 and 9. Note that both tetrachords 4-20 and 4-26 contain a major and a minor triad, but there is a great difference between them: 4-20 includes a semitone ($ic1 = 1$) while 4-26 does not ($ic1 = 0$).

Therefore, representing the trichords and tetrachords on the rectangular or polar grids gives us an insight into their inner structures and the relationship among them. Moreover, representing those chords on the Harmonic Wheel, which allows to visualize the relations among the chords belonging to a major or minor scale, will give us a further insight into the characteristics of tonal music, the relationship among the keys and a broader perspective of the theory of modulation.

4. REPRESENTATION OF HARMONIES

Apart from representing single chords, we will look for more general applications. Thus, a pair of examples on diatonic modulation are given, which deal with finding the pivot chords. As well, it is interesting to represent full harmonies from musical excerpts or whole pieces, since they can show the underlying design of the composition. Thus, five examples from different musical styles are represented on the Harmonic Wheel. The corresponding audios are available both on Spotify and iTunes. All representations happened to be simple and compact, and therefore easy to memorize, which makes this representation system an efficient mnemonic notation. Apart from Tables 1 and 2, Table 3 gives the symbols and notes of other extended and altered chords used in the examples, for the root C.

4.1. Diatonic Modulation: Pivot Chords

Diatonic modulation consists in changing from one key to another by means of a common or pivot chord, which is interpreted differently in each key [12, 13]. Finding all pivot chords between the two keys by comparing the chords associated to each of them is laborious. On the contrary, the Harmonic Wheel provides a simple and visual procedure for finding them. For simplicity, we will only consider consonant pivot chords, that is, major or minor.

We will start with the modulation from C Major to G Major. Figure 10 shows the original key with dark blue lines and, superimposed to it, the destination key with light blue lines. For clarity, only the curved rectangles defining the keys are represented. But, in practice, the scale degrees printed on the plastic disc will show the two degrees each chord represents in the two keys. From that figure, it is obvious that there are four common or pivot chords: C, Am, G and Em.

Secondly, let us examine the modulation from D♭ Major to B Major, which will involve enharmonic chords. The two keys are shown in Figure 11 with dark and light blue lines, respectively. The pivot chords are directly obtained from this figure, which are either G♭ and E♭m or F♯ and D♯m.

Chord	Notes	Chord	Notes
C6	C E G A	Cm(9)	C E♭ G D
C(9)	C E G D	Cm9	C E♭ G B♭ D
C9	C E G B♭ D	Cm7/6	C E♭ G A B♭
C7♯9	C E G B♭ D♯	Cm7/11	C E♭ G B♭ F
C9/13	C E G B♭ D A	Cm11	C E♭ G B♭ D F

Table 3. Symbols and notes of some extended and altered chords with root C.

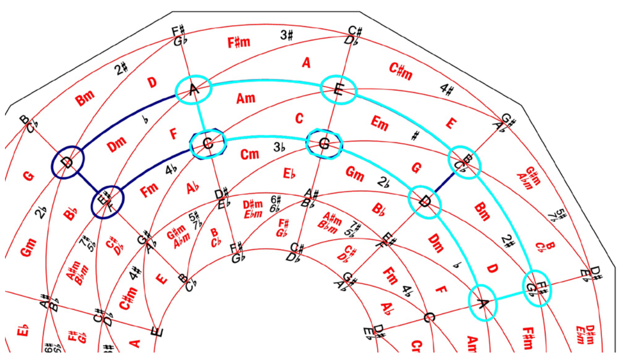


Figure 10. Pivot chords when modulating from C Major to G Major.

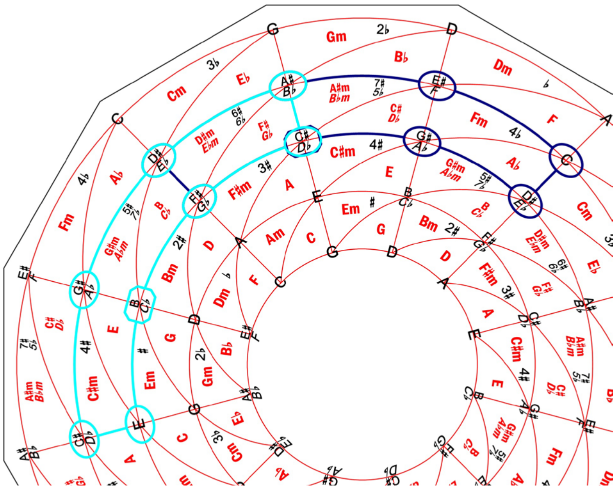


Figure 11. Pivot chords when modulating from D♭ Major to B Major.

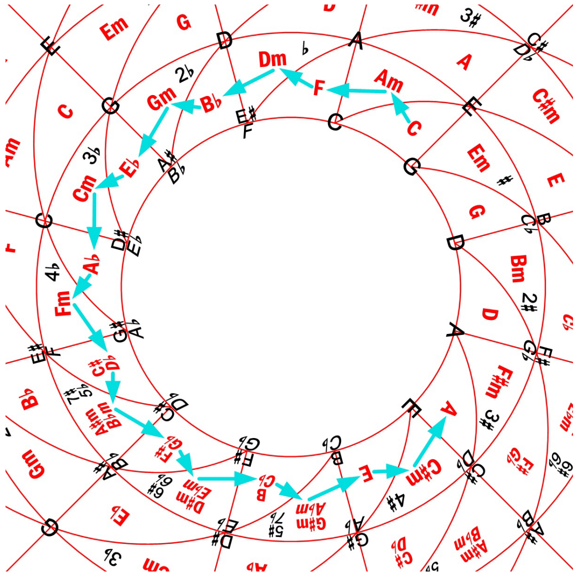


Figure 12. Harmonic structure of Beethoven's Ninth Symphony, second movement, mm. 143-176.

4.2. RL operations: Beethoven's Ninth Symphony

The next example is an excerpt from the second movement (Scherzo) of Beethoven's Ninth Symphony. In mm. 143-176, there is a series of consonant triads related by *R* and *L* operations as follows:

C Am F Dm B♭ Gm E♭ Cm A♭ Fm

D♭ B♭m G♭ E♭m C♭ A♭m E C♯m A

Figure 12 shows these triads on the Harmonic Wheel, where they follow a circular pattern on the cycle of fifths (or fourths), which is incomplete but includes both the major and minor triads. In each *R* or *L* operation, two notes remain fixed, which are shown on the interval line being crossed. The same example is analyzed in [7] and represented on a 2D Tonnetz, but in this case the representation takes up a full page due to the length of the chord progression, even though it does not complete an entire cycle.

4.3. Tonal Composition: Autumn Leaves

Autumn Leaves by Kosma [14] is one of the most well-known Jazz Standards and one that Jazz students first learn. Its harmony repeats the following chord sequence:

Cm7 F7 B♭Δ E♭Δ A^{ø} {D7} Gm (G7)

All these chords belong to the same key, B♭ major or G natural minor, with the only exception of D7, which for that reason is written in braces. The last chord, G7 in parentheses, does not belong to that key either, but it is included only sometimes to resolve to Cm7. Except Gm, all chords are seventh chords, so they cannot be represented properly on the Tonnetz (since it just contains triads). On the contrary, they largely match the chord types indicated on the Harmonic Wheel when choosing the B♭ major scale (Figure 13). The chord Dm7 is substituted with D7 to resolve to Gm. As well, there is a last chord sequence, slightly different from the previous one:

{C9 Fm7 B♭7} E♭Δ A^{ø} {D7♯5} Gm (G7)

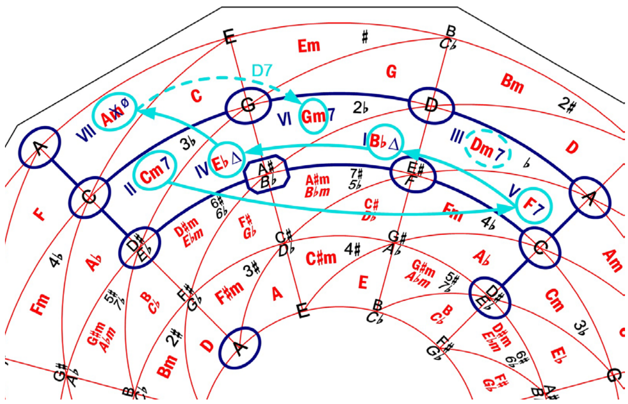


Figure 13. Harmonic structure of Autumn Leaves.

where the chords that are different are in braces.

This song is, in some cases, studied in the 12 keys [15], so using the Harmonic Wheel is of great help to visualize and memorize the harmonies.

4.4. Chords by Major Thirds: Giant Steps

In contrast to the last example, *Giant Steps* by Coltrane [16] is considered a challenging Jazz tune, since its harmony does not belong to any major or minor key. On the contrary, it follows a major third cycle, that is, a spiral line on the Harmonic Wheel. The full chord sequence is the following:

BΔ [D7] GΔ [B♭7] E♭Δ

[Am7 D7] GΔ [B♭7] E♭Δ [F♯7] BΔ

[Fm7 B♭7] E♭Δ [Am7 D7] GΔ [C♯m7 F♯7] BΔ

[Fm7 B♭7] E♭Δ [C♯m7 F♯7]

This harmony is based on 3 major seventh chords: BΔ, GΔ and E♭Δ, whose roots are a major third apart, thus dividing the octave into 3 equal parts. If we consider those chords as I degrees, the chords before them are either a V7 or a pair IIm7 V7, which are written in brackets to simplify the analysis. The cadence IIm7 V7 IΔ is the same as the first 3 chords in our last example (Cm7 F7 B♭Δ), which has a clear representation on the Harmonic Wheel. The other cadence, V7 IΔ, is simply a reduction of that one.

Figure 14 shows the harmonic structure of this piece, where the 3 main chords (BΔ, GΔ and E♭Δ) follow a spiral line and complete a major third cycle. The cadences V7 IΔ and IIm7 V7 IΔ are represented by solid and dashed lines, respectively. For clarity, the first part of the song is represented in blue and the second one in green. Of course, the 3 major chords are assumed to be major seventh chords.

As in the previous example, this song is also studied in the 12 keys [17], so the Harmonic Wheel is again a helpful tool, once the diagram in Figure 14 has been memorized.

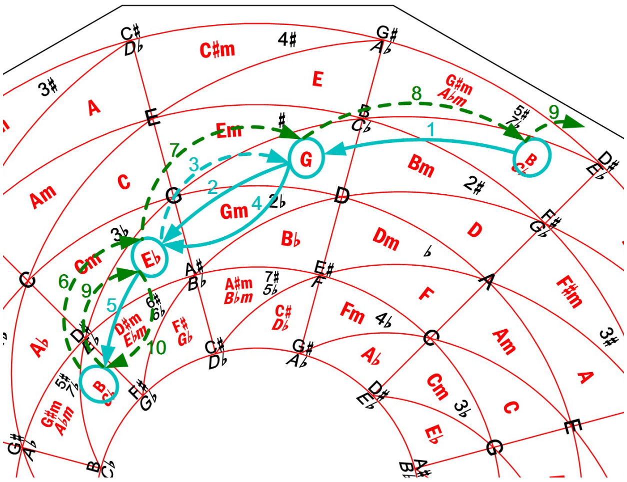


Figure 14. Harmonic structure of Giant Steps.

4.5. Chords in Béla Bartók's Axes: Indudable

Béla Bartók's axis system was first published by one of his disciples, Ernö Lendvai, after performing an exhaustive analysis of his work [18]. In summary, it states that relative and parallel chords have the same harmonic function (tonic, subdominant or dominant). This leads to groups of 8 chords with the same harmonic function, their roots being a minor third apart; that is, they follow a minor third cycle or a radius on the Harmonic Wheel. For example,

C Am A F♯m F♯ E♭m E♭ Cm

Indudable by Nuño [19] is a Bossa Nova whose second section has a harmony based on 2 such axes, one with major chords and the other with minor chords. The following basic chord sequence is played four times:

G♯m C♯ Fm B♭ Dm G Bm E

The major chords C♯, B♭, G, E belong to one of the axes and the minor chords G♯m, Fm, Dm, Bm to the other one. If the first chord, G♯m, is considered a Im degree, the next chord is the major IV degree (as in a melodic minor scale), C♯, which is enharmonic to D♭. Then, if this chord is now considered a new I degree, then the next chord is the IIIm degree (as in a major scale), Fm. And this process is repeated cyclically. The corresponding diagram is represented in Figure 15, where the relationship among the chords, as well as the minor third cycles in the radial direction followed by them, are clearly shown.

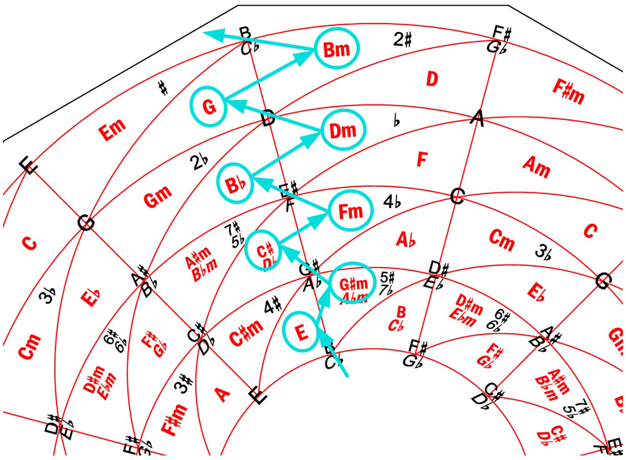


Figure 15. Harmonic structure of Indudable, second section.

The real chords, however, are more complex, since they contain 3 to 6 notes to enrich the harmony. The actual chord sequence is given below and is played twice. For clarity, similar chords have been grouped in brackets.

[G♯m(9) G♯m7/6] [C♯ C♯Δ] [Fm(9) Fm9] [B♭ B♭Δ]

[Dm7 ⁄] [G6 ⁄] [Bm7 ⁄] [E7sus ⁄]

[G♯m(9) G♯m7/6] [C♯^{O} ⁄] [Fm(9) Fm7/6] [B♭^{O} ⁄]

[Dm11 ⁄] [G6 ⁄] [Bm7 Bm7/11] [E7♭5 ⁄]

Each chord lasts one beat and the symbol "⁄" means to repeat the last beat (in these cases, the previous chord). As well, two diminished-seventh chords (C♯^{O} and B♭^{O}) were included to increase the variety of chord types.

4.6. A Piece with Modulations: All The Things You Are

All The Things You Are by Kern [14] is another well-known Jazz song. An Intro consisting of chords D♭7♯9 and C7♯9 is followed by this harmony:

Fm7 B♭m7 E♭7 A♭Δ D♭Δ [Dm7 G7] CΔ 𝄎

Cm7 Fm7 B♭7 E♭Δ A♭Δ [A$^{\emptyset}$ D7] GΔ 𝄎

Am7 D7 GΔ 𝄎 F♯$^{\emptyset}$ B7 EΔ {C7♯5}

Fm7 B♭m7 E♭7 A♭Δ D♭Δ {G♭9/13}

Cm7 {B^{O}} B♭m7 E♭7 A♭6 (G$^{\emptyset}$ C7)

where each chord lasts one measure, as well as each pair in brackets or parentheses, and the symbol "𝄎" means to repeat the last measure (in these cases, the previous chord).

The chords in the first phrase belong to the same key, A♭ major or F natural minor, except the last three, which form a cadence ending in CΔ, the dominant of F minor. Figure 16 shows these harmonies, where the first chord types are exactly as indicated on the plastic disc.

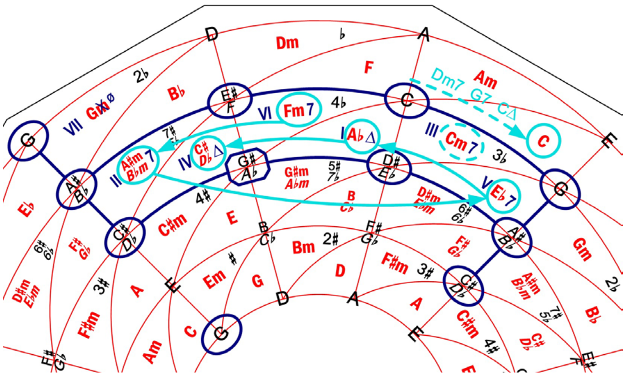


Figure 16. Harmonic structure of All The Things You Are, first phrase.

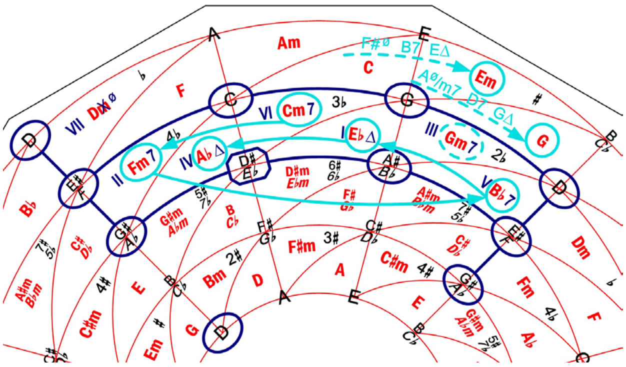


Figure 17. Harmonic structure of All The Things You Are, second phrase.

The chords in the second phrase are analogous to those in the first one, but in E♭ major or C natural minor, which represents a (transient) modulation to the dominant. The new chords are found by simply rotating the plastic disc one step clockwise, as seen in Figure 17. Again, the first chord types are exactly as indicated on the plastic disc. Unlike the first phrase, now the cadence starts with a half-diminished chord, $A^{\varnothing}$, but then it is repeated starting with Am7. Another cadence follows, ending in EΔ, and moving to C7♯5 to resolve to Fm7, the original key.

The rest of chords, with two exceptions written in braces, belong to A♭ major or F natural minor, so they are found by rotating the plastic disc one step counterclockwise. Now, the Cm7 chord is included and the phrase ends in A♭6, the tonic chord with the added major sixth. The last two chords in parentheses are used to return to the beginning.

As shown in Figures 16 and 17, most chords in this song are diatonic to a particular key, so the chord types are exactly as indicated on the Harmonic Wheel. Moreover, most of the time, their roots move by descending fourths, thus being easy to visualize and memorize.

5. CONCLUSIONS

The Harmonic Wheel is a physical tool consisting of two rotating discs: one cardboard, with a Tonnetz transformed into a polar grid, and the other plastic, with the lines defining the major, harmonic and melodic minor scales, together with the scale degrees and the symbols of the corresponding seventh chords. It has been used to represent single chords, excerpts and full harmonies from different musical styles, considering both triads and seventh chords, and including modulations. In some cases, the harmonies followed one of the three cycles defined by the consonant intervals and, in others, they were diatonic to a particular key. In all cases, the Harmonic Wheel has proved to be a powerful and versatile tool for representing the harmonies and, therefore, to show the underlying structures of the musical compositions here considered. Furthermore, it provides an efficient mnemonic notation by means of simple and compact diagrams.

6. REFERENCES

[1] A. S. Crans, T. M. Fiore and R. Satyendra, "Musical Actions of Dihedral Groups," *American Mathematical Monthly*, vol. 116, no. 6, pp. 479-495, 2009.

[2] J. Douthett, P. Steinbach, "Parsimonious Graphs: A Study in Parsimony, Contextual Transformations, and Modes of Limited Transposition," *Journal of Music Theory*, vol. 42, no. 2, pp. 241-263, 1998.

[3] E. Gollin, "Some Aspects of Three-Dimensional 'Tonnetze'," *Journal of Music Theory*, vol. 42, no. 2, pp. 195-206, 1998.

[4] D. Tymoczko, "The Generalized Tonnetz," *Journal of Music Theory*, vol. 56, no. 1, pp. 1-52, 2012.

[5] L. A. Piovan, "A Tonnetz Model for Pentachords," *Journal of Mathematics and Music*, vol. 7, no. 1, pp. 29-53, 2013.

[6] V. Mohanty, "A 5-Dimensional Tonnetz for Symmetric Hexachords," *Journal of Mathematics and Music*, 2020.

[7] R. Cohn, "Neo-Riemannian operations, trichords, and their 'Tonnetz' representations," *Journal of Music Theory*, vol. 41, no. 1, pp. 1-66, 1997.

[8] S. B. P. Briginshaw, "A Neo-Riemannian Jazz Analysis," *Nota Bene: Canadian Undergraduate Journal of Musicology*, vol. 5, no. 1, pp. 57-87, 2012.

[9] L. Nuño, *Harmonic Wheel*. Valencia, Spain: Luis Nuño, 2008.

[10] A. Forte, *The Structure of Atonal Music*. New Haven, CT: Yale University Press, 1973.

[11] L. Nuño, "A Detailed List and a Periodic Table of Set Classes," *Journal of Mathematics and Music*, 2020.

[12] A. Schönberg, *Theory of Harmony*. 3rd ed. Berkeley, CA: University of California Press, 1983.

[13] W. Piston, *Harmony*. 5th ed. New York: W. W. Norton & Co, 1988.

[14] C. Sher, *The New Real Book, Vol. 1*. Petaluma, CA: Sher Music Co., 1988.

[15] J. Aebersold, *Vol. 67: Tune Up - Standards in All 12 Keys*. New Albany, IN: Jamey Aebersold Jazz, Inc., 1995.

[16] C. Sher, *The New Real Book, Vol. II*. Petaluma, CA: Sher Music Co., 1991.

[17] J. Aebersold, *Vol. 68: Giant Steps - Standards in All 12 Keys*. New Albany, IN: Jamey Aebersold Jazz, Inc., 1995.

[18] E. Lendvai, *Béla Bartók. Un Análisis de su Música (Béla Bartók. An Analysis of his Music)*. Barcelona, Spain: IDEA BOOKS, S. A., 2003.

[19] L. Nuño, *Puesta de Sol, Vol. 1*. Madrid, Spain: Acordes Concert S. L., 2012.

RESURFACING MECHANICS AND ACTION IN MUSIC NOTATION: THE *MUSICWRITER*

Mauricio Rodriguez

San Francisco Conservatory of Music
mrodriguez@sfcm.edu

ABSTRACT

This text discusses the compositional use of the *Musicwriter*, a music notation device used after old engraving practices such as lithographic notation and metal printing blocks for music printing, but discontinued before modern digital notation. After a short description of this singular device, the author presents his ongoing composition/engraving project: *'Meaning The Score'*, a series of *tyscores* (or typed scores) performed using the Musicwriter; the notation of this series is conceived as an emergent property of the performative interaction with the Musicwriter, notation that is later reinterpreted by musicians that react to both, the live-typeset notation and the composer's performance that created the score in the first place.

1. MUSICWRITER

As early as 1885, Charles Spiro patented his *Columbia Music Typewriter*, a rotating disc with metal music-types to manually press or stamp on paper to notate music. Later 'music stamping' machines, such as the French *Dogilbert* (1905), the German *Nocoblick* (1910), the British *Walton Music Typewriter* (1923), and the German *Melotyp/Nototyp* (1931), were all original inventions that paved the road to one of the most flexible and interesting music typewriters: the *Keaton Typewriter*.

In 1936, Robert Keaton from San Francisco, California patented his music typewriter, a round portable metal reel of music-types mounted on a drafting-like table. Keaton's unique design allowed to treat the engraving surface as an open canvas, basically making possible any imaginable music layout (Figure 1).

Keaton's first machine was ten years later superseded by composer Cecil Effinger, inventor of his own music typewriter. Effinger's *Musicwriter* is basically a modified typewriter; it uses a reduced set of music notation symbol instead of the usual alphanumeric characters found in conventional typewriters. Anyone familiar with alphanumeric typesetting would find the Musicwriter's keyboard configuration as an intuitive and ergonomic interactive device (Figure 2).

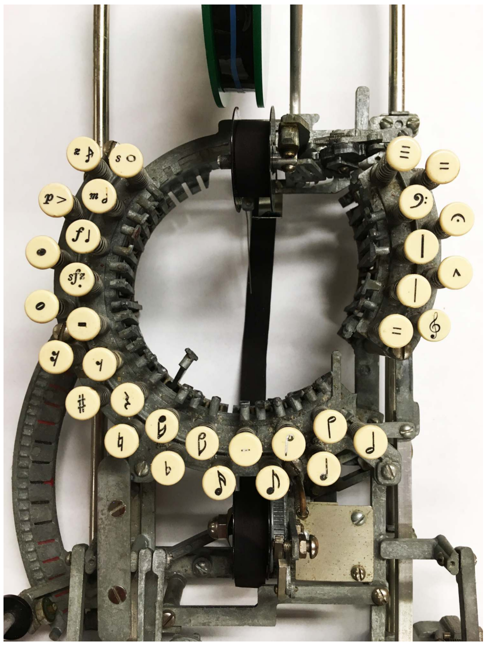


Figure 1. Keaton Music Typewriter. (Archive of Rec-orded Sound, Stanford University).

Figure 2. *Musicwriter* (Olympia GS3). Keyboard layout.

Effinger's company, *Music Print Corporation*, produced the prototype model for later music typewriters built in partnership with *Smith-Corona* (in 1950), *Allen* (in 1955), *Olympia* (in 1965), and even an electronic model produced by *IBM* in the late 1980s.

Despite its thoughtful design, the Musicwriter never really had a widespread use in major publishing houses or professional typesetting environments; its relatively high cost, laborious process to typeset conventional music, and its delicate calibration (extremely prone to errors), were all factors together that favor its eventual decay. However, all these inconveniences would eventually have been overcome if the Musicwriter had not been overshadowed once and for all due to the emergence of digital notation in the late 1960s.

2.TYPED SCORES OR TYSCORES & *MEANING THE SCORE*

Meaning The Score is a series of typeset scores or *tyscores* using the Musicwriter in its Olympia version (a modified GS3 machine). This piece is an evolving work in which each page of it is live-typeset as part of its performance. This is a work whose 'in situ' notation [1] is conceived as an emergent property of the performative interaction [2] with the Musicwriter, notation that is later reinterpreted by musicians that react to both, the live-typeset score and the composer's performance that created the score in the first place (Figure 3).

Figure 3. Live-typesetting of *Meaning The Score*. Matt Ingalls (Clarinet) & Mauricio Rodriguez (*Musicwriter*).

Due to the incremental nature and visual scope of the full score, the pages of *Meaning The Score* are usually displayed as standalone visual works in gallery settings. Over its various performances in collaboration with artists such as Wilfrido Terrazas, Matt Ingalls, and Guillermo Galindo, among others, this project has become an open platform to explore the dynamic relationship between music and its multiple forms of representation [4, 5, 6]. The endless notational plasticity of the Musicwriter to create different forms of music representation has definitely met the aesthetic and creative expressions (sonic and visually) of this artistic project (Figures 4 & 5).

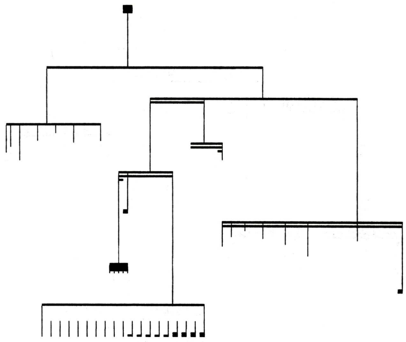


Figure 4. *Tyscore No. 002*. Ribbon ink on paper.

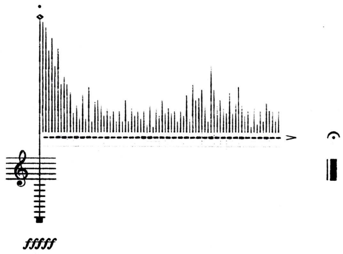


Figure 5. *Tyscore No. 016*. Ribbon ink on paper.

The Musicwriter was conceived to typeset conventional or common Western music notation [3]; despite the reduced set of the conventional music symbols embedded in the Musicwriter, the flexibility of its design allows to notate music documents of the most varied representations; as an example of that, it is shown here the author's transcription of one page of 'Threnody to the Victims of Hiroshima' by Krzysztof Penderecki [7] (Figure 6), and a transcribed fragment of 'Tertium Datur' by Boguslaw Schaeffer [8, 9] (Figure 7):

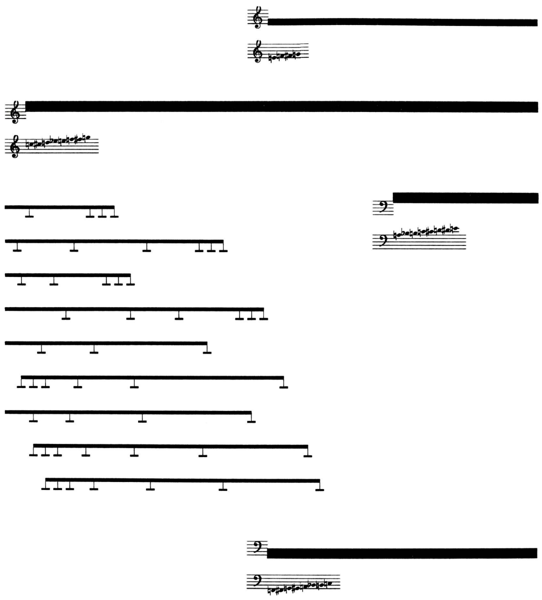


Figure 6. Tyscore Transcription of *Threnody to the Vic-tims of Hiroshima* by Penderecki.

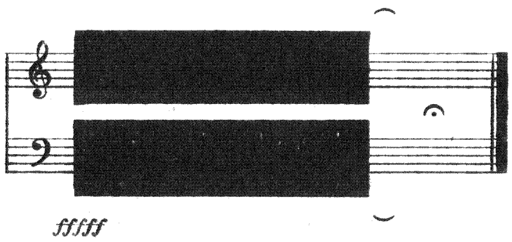


Figure 7. Tyscore Transcription of *Tertium Datur* by Schaeffer.

Thanks to the veteran-owned New York based company *FJA*, who have provided the author with original hand-made colored ribbon spools, the visual expression of the represented music adds some interesting subtlety with some coloring enhancement (Figure 8 and 9).

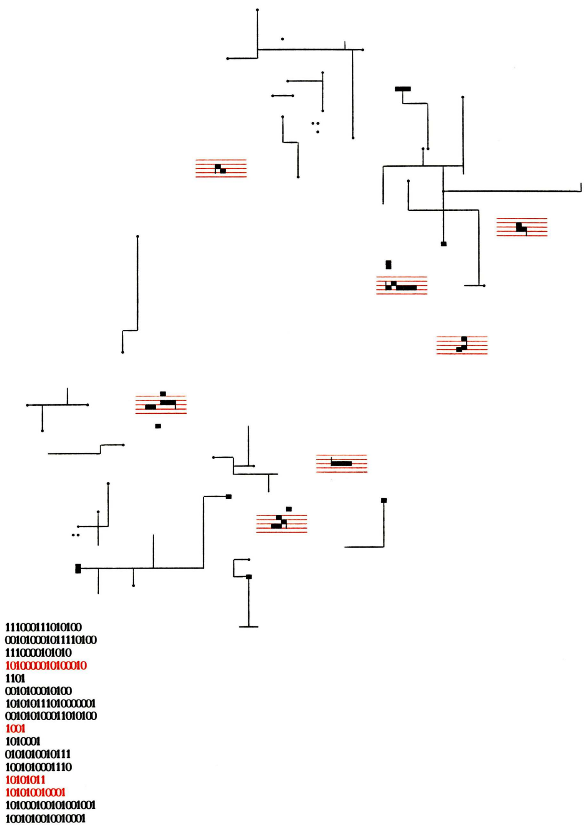


Figure 8. *Tyscore No. 068*. Colored ribbon ink on paper.

3.INTER-TEXT: PIANO ROLL MUSIC & *MUSICWRITER*

To show some additional interactions with the Musicwriter, an ongoing project incarnation of *Meaning The Score* uses piano rolls as the typeset surface. In this case, the tyscores result as the musical re-interpretation of the codified (punched) music on the rolls, so the typeset scores are 'visual/musical comments' over the visual/musical perforations on the rolls. The created pieces are then re-interpreted by musicians who react to the gestural live-typesetting performance, the typeset score, the piano roll perforations, and to the physical disposition of these very large-format scores (over 15 feet) on the displaying space (Figures 10, 11, 12).

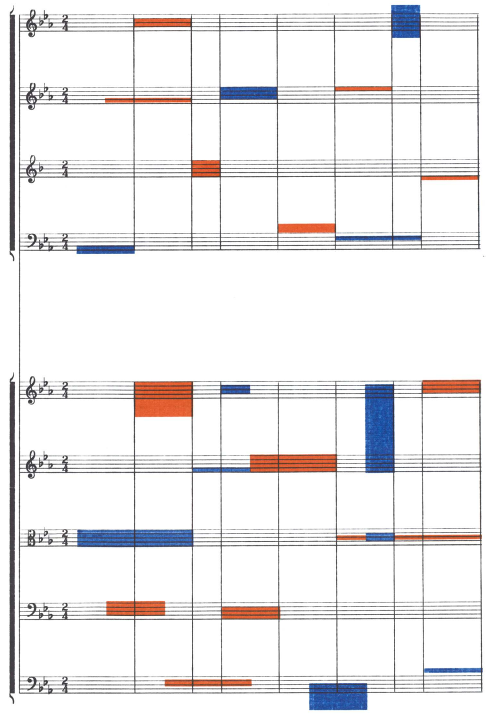


Figure 9. *Tyscore No. 080*. Colored ribbon ink on paper.

To appreciate the creative process in *Meaning The Score*, the following video-link presents a performance in collaboration with composer/improviser Matt Ingalls:

www.mauricio-rodriguez.com/MTS.mp4

4. CONCLUSIONS

The *Musicwriter* is an old heavy-duty typesetting device that never really had a widespread use in professional music typesetting. Its very laborious usage (a one single-page of conventional music notation averages 5000 key strokes), its extremely prone-to-error complicated calibration, its heavy weight (around 42 pounds), and most importantly, its disadvantageous position before digital notation technology, were all factors that contributed to the permanent discontinuation of this original music notation device.

Nevertheless, using the Musicwriter as a musical instrument for live-typesetting performances has opened a fruitful space to explore the multiple and dynamic relations of music and visual design. *Meaning The Score* is a work that hopes to stimulate a unique appreciation of sound and music through varied forms of visual representation, but overall, this work series aims to revive an engraving practice that is almost unknown in the times of a generalized usage of music notation software.

www.mauricio-rodriguez.com/tyscore.html

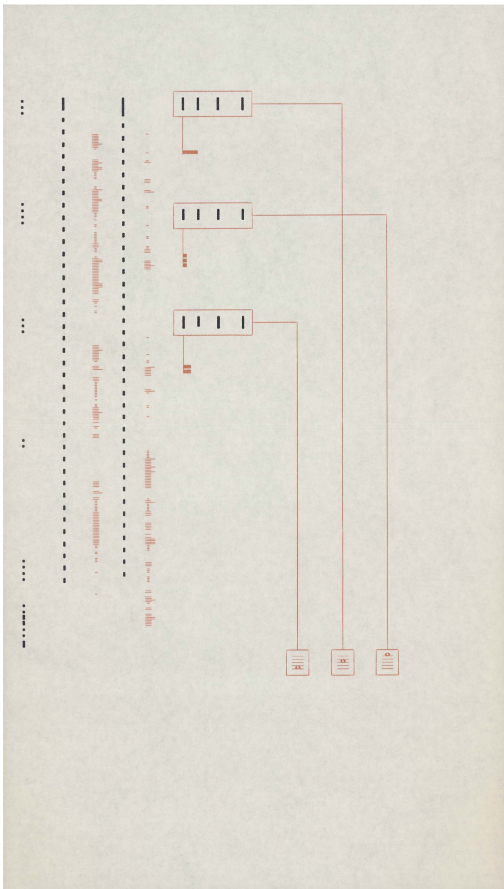


Figure 10. *Tyscore No. 120*. Colored ribbon ink on piano roll.

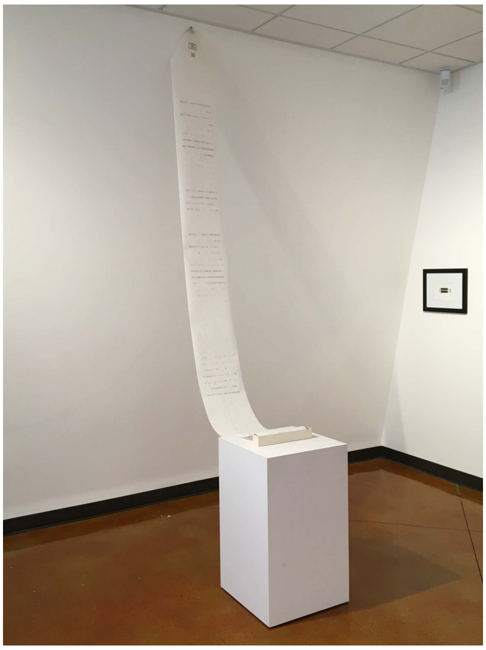


Figure 11. *Tyscore No. 138*. Ribbon ink on piano roll.

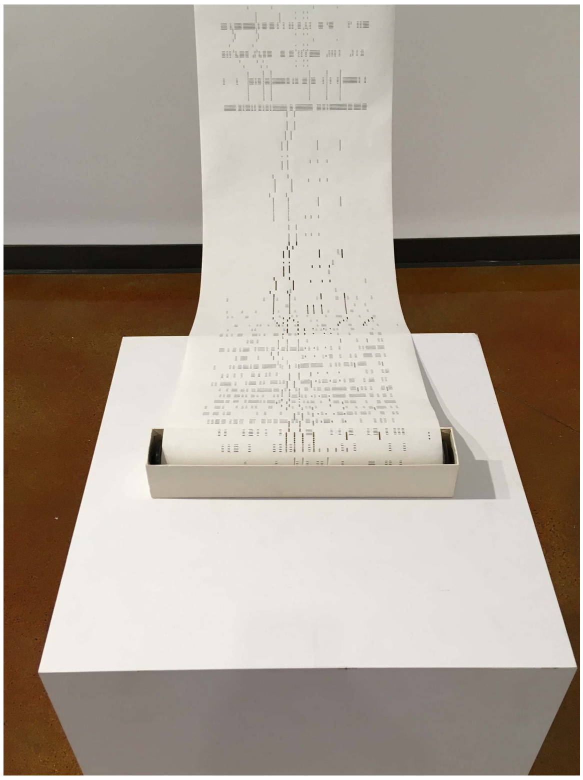


Figure 11. *Tyscore No. 138*. Ribbon ink on piano roll.

5. REFERENCES

[1] H. Ruprecht, Taktlos Musiklesebuch. Urs Engeler, 2006.

[2] L. Austin, D. Kahn, & N. Gurusinghe (Eds.), Source: Music of the Avant-garde, 1966–1973. University of California Press, 2011.

[3] E. Selfridge-Field, *Beyond Midi*. Center for Computer Assisted Research in the Humanities – MIT Press, 1997.

[4] J. Cage, *Notations*. Something Else Press, 1969.

[5] T. Sauer, *Notations 21*. Mark Batty Pub, 2009.

[6] M. Rodriguez, “Xa-lan: Algorithmic Generation of Expressive Music Scores Based on Signal Analysis and Graphical Transformations” in *Proceedings of the International Workshop on Musical Metacreation - 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Stanford University, 2012, pp. 83-85.

[7] E. Karkoschka. *Schriftbild Der Neuen Musik*. Moeck, 1966.

[8] A. M. Locatelli de Pergamo, *La notación de la música contemporánea*. Editorial Ricordi Americana, 1973.

[9] B. Schaeffer, *Introduction to composition.* Polskie Wydawnictwo Muzyczne, 1976.

NOTATION OF GESTURE AND MODELING: THE PROCESS OF COMPOSITION OF *MOTS DE JEU*

Jean-François Trubert
University of Côte d'Azur
France
Jean-Francois.Trubert@unice.fr

Alireza Farhang
University of Côte d'Azur, France
(IDEX UCA-JEDI, ANR15-IDEX-01)
University of Antwerp, Belgium
alireza_farhang@yahoo.fr

ABSTRACT

This paper aims to present the process of composition of a piece in which the musical material integrates syntactic and semantic dimensions of a poetic language. Modeling the semantics of speech gestures through graphical notation and using formant synthesis to generate the electronic sounds are also the subjects that will be explained in order to give an outline about the way poetry and prosody contribute to the microstructure and macrostructure of a vocal piece.

1. INTRODUCTION

Mots de jeu[1] is born of the challenge to compose a musical work built on the sensitive and emotional content of a text, and thereby to create what could be called a speech gesture. Beyond the abstract meaning of words (the signified), the composition seeks to capture the vocal gestural imprint of poetry and use it as both a morphological and structural model for the entire work, acting as a counterpoint to the signifier.

From a technical perspective, the piece involves reproducing phonemes and generating sounds using formant synthesis [1] in the OpenMusic visual programming environment [2, 3, 4] (OM-Chant library [5, 6, 7]), and producing a sound that gives the illusion of an augmented human voice. The ambiguous quality produced by formant synthesis echoes the challenges specific to the language of several poetic texts drawn from the collection *L'Espace du dedans* by Henri Michaux. The language of these poems is constantly revealing new discoveries, and their original and emerging poetic content can never be fully grasped,[2] opening the door to multiple points of view in a process of composition that is constantly moving back and forth between the microscopic and local dimension of the language and its models (phonemes, syllables and words) and the macroscopic dimension (syntactic and semantic). The result is a superposition of formal structures organized according to the vocal gestures selected for their paradigmatic function as form generators. As we will explain below, this has an impact not only on the sound structures but also on the spectral morphology of the whole piece, as the work on vocal formants necessarily has an incidence on the harmonic and textural dimension of the synthesis process and its combination with the voices.

The composition process of this piece will therefore reveal what happens upstream of the sound matter where, through graphic schemes, the prosody of Henri Michaux's text gives birth to an acoustic matter. How is the poetic environment perceived and analyzed? What vocal gestures have been combined and how? How can a gesture or vocal gestures lead to more macroscopic musical structures and what notation signs could be used to convey them? How do we bring human vocal matter in contact with synthesized vocal matter?

2. MODELING PROSODY

To meet the challenge of creating a formal structure for this piece based on the voice and of mobilizing all the vocal resources of the poem, it was essential to work from a text that could be segmented into small speech units, while retaining the poetic expression of the whole. The poetic material also needed to be malleable to a certain extent to allow us to move the words, the syllables and the phonemes, for instance. This full potential is immediately apparent on reading the poem collection *L'Espace du dedans* by Henri Michaux, where these very aspects reveal a tremendous musical power.[3]

[1] Mots de jeu for 5 female voices and electronics is the first of a series of pieces entitled *L'Espace du dedans*, based on the text by the Franco-Belgian poet, Henri Michaux (1899-1984). Commissioned by the French Center of Musical Creation (CIRM), it was created on December 9, 2018 by the vocal ensemble Mora Vocis at the Marc Chagall museum in Nice, as part of the Manca festival, with the assistance of Camille Giuglaris.

[2] "[…] while ordinary language tends to vanish, as soon as it is understood, to make room for the ideas, impressions, acts, etc. that it evokes, poetry tends, in its very form, to persist in our mind; the poem is something that lasts, it is *par excellence* memorable." [8, p.152]

[3] Consider the example below:
« Comme une cloche sonnant un malheur, une note, une note n'écoutant qu'elle-même, une note à travers tout, une note basse comme un coup de pied dans le ventre, une note âgée, une note comme une minute qui aurait à percer un siècle, une note tenue à travers le discorde des voix, une note comme un avertissement de mort, une note, cette heure durant m'avertit. »
This passage is used in the final section of the work (measure 98 until the end of the piece). The text revolves around the words "une note", repeated 9 times. With this repetition, the poet uses a circular form to

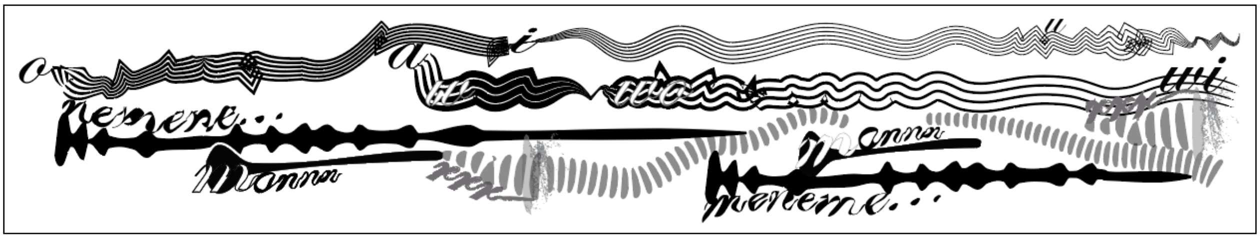


Figure 1. Graphic representation of a speech gesture.

Regarding the composition of *Mots de jeu*, the goal was not to simply reflect the text in the music,[4] but rather to follow the principle of a geometric translation: to model a form of contamination between verbal matter and musical matter, where the texture of the words – both their meaning and the bodily form of an intention or of a *logos* – would then serve as reference [9, pp. 12-20]. To achieve this goal, a graphic representation of vocal gestures was first imagined as an abstraction of speech and as a graphic representation of its acoustic temporal form (see Figure 1).

The preliminary process therefore involved modeling speech gestures and defining graphic schemes. The value of these schemes is strictly subjective. The graphic sign is placed in a space (on the page) that is neither ordered nor homogeneous, where it combines several functions conveying the strength, the energy, and the internal movement of the gestures that arise from the words, the letters, and the interjections, etc. In some cases relationships are established between the size of the letters and the dynamic and in others, correlations are made between durations, etc. (see Figure 2).

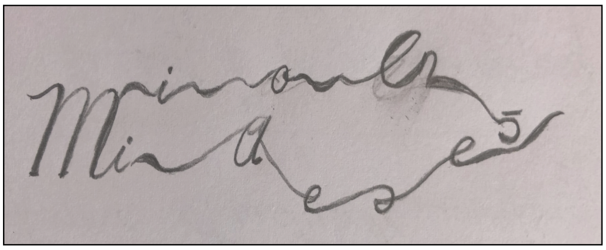


Figure 2. The strength, energy and internal movement of the phonemic gestures.

2.1 Graphic design of vocal gestures

The A graphic scheme is a visual bridge which seeks to lead the observer from a concrete unit to an abstract one. The unit becomes an element of material in the Schaefferian sense of the term: from each unit, a character (essentially of a structural and morphological nature) emerges and crystallizes as it adopts different variables and becomes "a structure of value variation" to use Michel Chion's own words [10, p. 67]. Each graphic sign can be used multiple times, in different situations and with different parameter settings. We therefore consider that there are several levels of modeling of a source vocal gesture. Each is determined by the degree of distance (i.e. the degree of fidelity) with respect to the source gesture, from level 1 where there are many similarities between the phonology of the text and its spectral and transcribed translation, to levels 3 and 4, where elements from several source gestures are combined in a complex way such that the origin of the gestures can no longer be identified. At this level, the process of abstraction grants the freedom to translate the prosody into the sound matter illustrated in the graphic notation and materialized in music. This approach is applied to both the sound synthesis part and the vocal part.

The first level represents the morphology of words derived directly from their phonetic analysis. In this case, the graphic schemes represent the succession of phonemes and take into account the transitions or morphing (passage from one state to another without a noticeable transition) between them. The first task was to group phonemes into classes according to their spectral content (see Table 1).

express the multiple facets of "une note". As indicated in the score, this forms a counterpoint of words or from a certain point of view, a counterpoint of vocal gestures produced by juxtaposing the sung and breathed gestures of two groups of singers (voices I and II against voices IV and V). The counterpoint is possible because of the circular and repetitive quality of the text that does not necessarily unfold in a linear fashion. Voice III then operates as a contact surface between the two groups of voices and gives a dramatic aspect to the musical discourse. The text also speaks of the independent quality of the music, of the fact that the only meaning of a note is the note itself without referring to an external meaning. In several respects, the act of reading the text is already music in its own right because it speaks about music in the same way that music speaks for itself.

[4] … and not necessarily by amplifying the meaning that music could add. Michaux was very reluctant to have his poetry set to music, and even categorically opposed to it, to the extent that he wrote to Robert Bréchon, the French poet and essayist, to tell him: "I am looking for a secretary who knows forty or fifty different ways of writing 'no' for me."

In 1966, weary of the requests he had received for adaptations, he wrote René Bertelé, the great specialist of 20th century literature, to tell him: "Would you be so kind as to answer no to this lady on my behalf. I am not taking any more risks and songs do not seem to me to be a good preparation for a composer who wants to capture 'exorcisms'."

In 1957, Pierre Boulez wrote to the poet: "I would like to set your *Poésie pour pouvoir* to music, and I would like you to read the text."

To which Henri Michaux responded: "Pierre Boulez, who is not just anyone, believed, even though I had warned him, that a musical composition with a powerful orchestration would add something or at least translate more directly (!) what this poem is about. After two auditions, the work was removed from the composer's catalogue."

H. Michaux, Lettre à Michel Mathieu, dated January 24, 1979, cited by J.-P. Martin.

	Vowels		Semi-vowels		Consonants: Plosives	Consonants: Fricatives	
Voiced	[ã] [ɛ̃] [œ̃] [ɔ̃]	Nasales	—	Nasales	[m] [n]	—	Voiced
	[a] [ɑ] [e] [ɛ] [ə] [i] [œ] [ø] [o] [ɔ] [u] [y]	Orales	[w] [ɥ] [j]	Orales	[b] [d] [g]	[v] [z] [ʒ]	
Voiceless	—		—		[p] [t] [k]	[f] [s] [ʃ]	Voiceless

Table 1. Classification of phonemes according to their spectral content.

	Plosive onset	Plosive offset	Soft onset	Soft offset	Stable	Unstable
Voiced	[b] [d] [g] [m] [n]	When a word ends with the plosives [b] [d] [g] [m] [n]	[w] [ɥ] [l]		[a] [ɑ] [e] [ɛ] [ə] [i] [œ] [ø] [o] [ɔ] [u] [y] [ã] [ɛ̃] [œ̃] [ɔ̃]	
Breathed	[p] [t] [k]	Quand un mot se termine par les plosives [p] [t] [k]			[f] [s] [ʃ]	[ʁ]
Voiced and breathed					[v] [z] [ʒ]	[ʀ] [r]

Table 2. The outline of the phoneme shapes.

In order to represent phonemes graphically, we grouped them in a slightly different, more subjective way. Categories were defined according to the rate of periodicity and the articulatory form of their spectral content. The black color represents the periodicity (voicing) and the gray color the breath noise (voiceless or devoiced phonemes). Some phonemes, such as vowels, have a simple and stable structure while others, such as liquids, have a shape that undergoes micro-evolutions. The phoneme [R] presents a more complex spectral structure. It is a vibrating phoneme (trill) whose shape varies depending on the context.7 Table 2 shows the outline of the phoneme shapes.

These profiles are used as a graphic alphabet, a type of abstraction designed to show the acoustic behavior of each phoneme. Obviously, the progressive transition from one phoneme to another must also be represented in the graphic scheme. Figure 3 shows the progression of the acoustic structure of the word "étrange".

Figure 3. Graphic scheme representing the profile of the acoustic structure of the word "étrange".

The word begins with the vowel [é], which has a hard attack and ends with the plosive consonant [t] which has an abrupt onset and offset, hence the short silence which precedes the phoneme [R] which has a granular character, and so on. The scheme also expresses the intonation of the French language. The upward movement of the word emphasizes the second syllable while the tone descends on the last two letters.

The second level is considered as an intermediate step in the transformation of the word gesture into a musical gesture. Thus, the graphic schemes that represent "phonemes" and "words", free themselves from their linguistic content (signified) to produce a musical gesture (a kind of virtual signifier). It can be quite short, limited to a simple phoneme or as long as a fraction of a word. The gesture is achieved by observing the spectral behavior of the profile and serves to form the composition material of the piece (see figure 4).

At the third level, the simple gestures are superimposed in order to build more complex gestures. At this level, the morphology moves away from its original state to the extent that source words can no longer be identified.

Voice processes can be included, such as vibratos, pitch changes, staccato or damped movements, which are not usual in spoken language. This level corresponds to purely vocal gestures, sometimes quite complex, but without identifiable words. At this stage, the macro-gestures are almost ready to be used as structural elements of the musical discourse.

The example below represents the gesture for the entry of voice V which is derived from the word "étrange". The vocal gesture is electronically augmented. The phoneme [R] which has a granular character, is first pronounced by the voice. Then the granular synthesized sound similar to the phoneme [R] transforms the beginning by stretching it out not only in time, but also in space. The spatialized synthesized sound which lasts around 4 seconds undergoes micro-variations in timbre. It is then enriched by another

Figure 4. The profile of the vocal gestures derived from the words “étrange”, “plus” and “chose”.

synthesized sound or gesture, which consists of the succession of the vowels [a] and [U] and a fairly fast vibrato. The result is represented graphically in Figure 5.

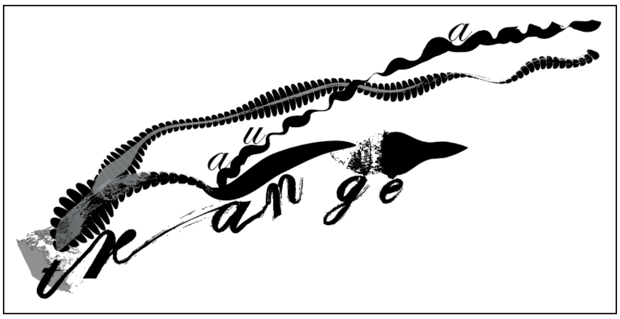


Figure 5. Profile of a complex gesture.

The fourth level appears in the final score, which can be considered as a hybrid medium where conventional notation meets graphic notation. We note that several complex gestures are superimposed to give a wider dimension to the vocal part. When this level is reached, the formal and textural progression of the work can be seen (see Figure 6).

The abstract aspect of these schemes generates a malleable material which, depending on the context, can be adapted to new musical situations in which the profiles adopt a new duration, a new spectrum, a new height, and a new temperament. Gestures can be segmented into shorter units or on the contrary combined with other micro-gestures to create a complex profile while sharing certain parameters such as temperament, etc.

Thus, the vocal gestures – sung but also electrically simulated – go through a series of derivations which produce variations in their parameters and particularly in their temporalities. The gestures can be very localized (limited to a phoneme or one syllable) or more extended (to a sentence or even a structural unit – see Figure 6).

2.2 Formal structure of the piece

The drawings also have a poetic and semantic function. They do not replace the traditional notation but complement it. They not only reflect the microstructures, but also a macrostructure which integrates the meaning of the text without using it in its original state. Consider for example the introduction and the first section of the piece composed from the following text:

Oh! Quelle étrange chose au début, ce courant qui se révèle, cet inattendu liquide, ce passage porteur, en soi, toujours et qui était. On ne reconnait plus d’entourage (le dur en est parti).
On a cessé de se heurter aux choses. On devient capitaine d’un FLEUVE…

In this excerpt, Michaux talks about music, time and the fact that music bends time and resists the flow of time. For the poet “to make music […] is to practice the art of drifting, which is not only to let oneself be carried along where the currents lead, but to modify the perceived movement of music.” [11] The introduction of the piece only represents synthesized sounds, accompanied by the singers’ body gestures, without voice. The synthesized sounds are modeled from the combination of fricative and plosive phonemes[5] such as [r] (a voiceless alveolar r), [ʃ], [k] and [p]. This opening represents both resistance and current (as in the image of the river). The modeled gestures therefore contain an accumulated energy which is suddenly released before coming up against other obstacles (see figure 7).

Measures 1 - 23	**24 - 39**	**40 - 86**	**87 - 97**	**98 - 114**	**115 - 129**	**130 - 144**
Entrance of the sung voice. This opening represents both resistance and current (as in the image of the river). This section has a floating aspect in which the words are not presented as such. The range is narrow. Measure 1 represents plosives and fricatives. Synthesized sounds only. Measures 2 to 23 mostly represent stretched vowels.	The words pronounced are more intelligible. The range is wider and the spoken voice meets the sung voice in counterpoint.	After a caesura, the discourse is now based on an increasingly dense flow of vocal gestures, sometimes unintelligible (simple or complex gestures), sometimes intelligible (words and sentences). The final chord first appears in measure 75.	A filtered breath (continuous or damped) based on whispered gestures begins this section of the piece. The spoken sentences are longer. The text is not necessarily intelligible however. The chord that briefly appeared in measure 75 reappears gradually through repeated gestures.	The discourse is led by the juxtaposed spoken and sung voices, where voice III seems to be opposed to the other voices.	The chord becomes more important. The textural progression appears in the relationships based on harmonic intervals.	This is the climax of the piece. The chord appears in its complete form such that the discourse resembles a chorale.

[5] A plosive occurs when a blocked airflow in the mouth, pharynx or glottis is suddenly released.

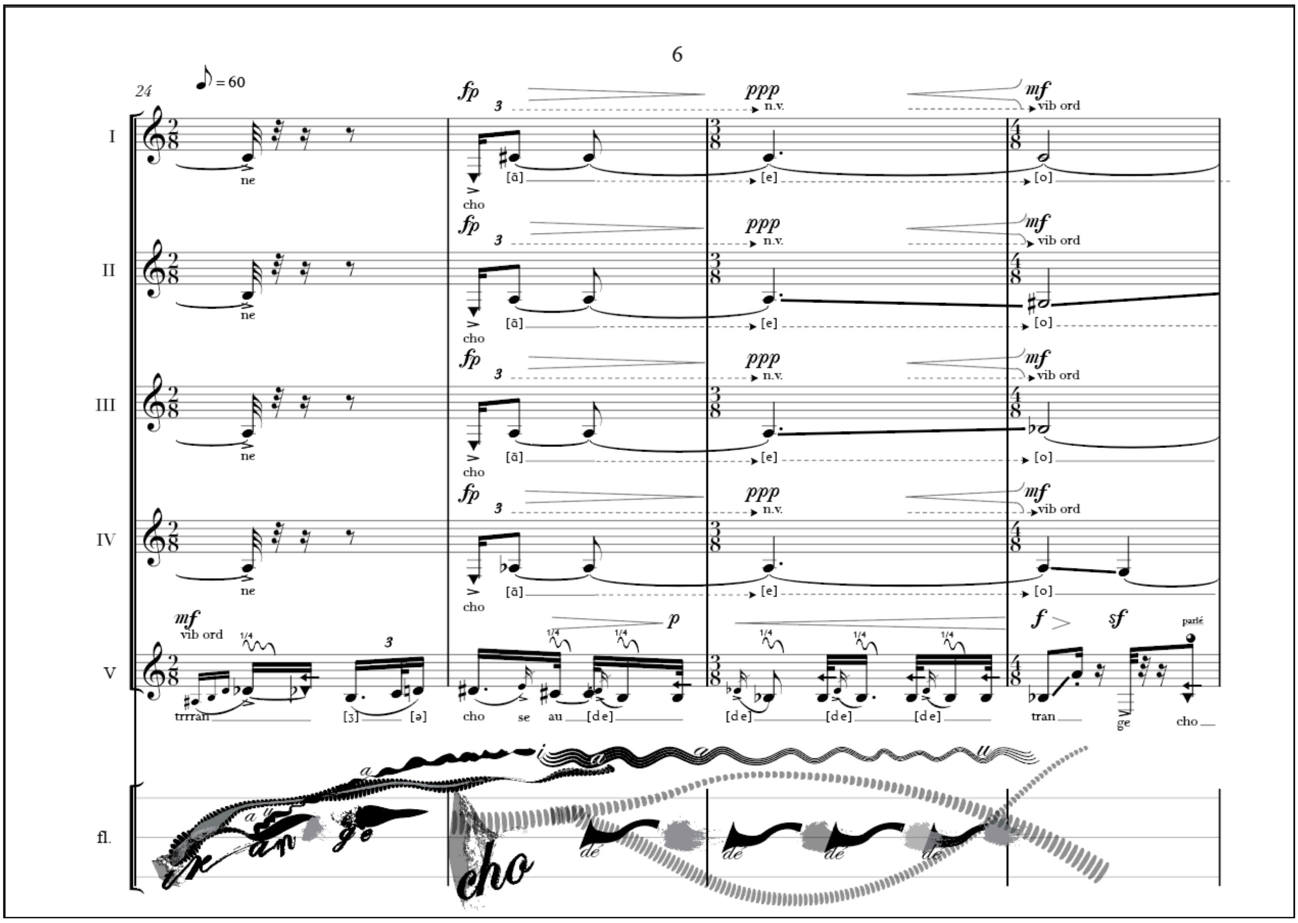


Figure 6. Graphic representation of gestures in the score (measures 24 to 26).

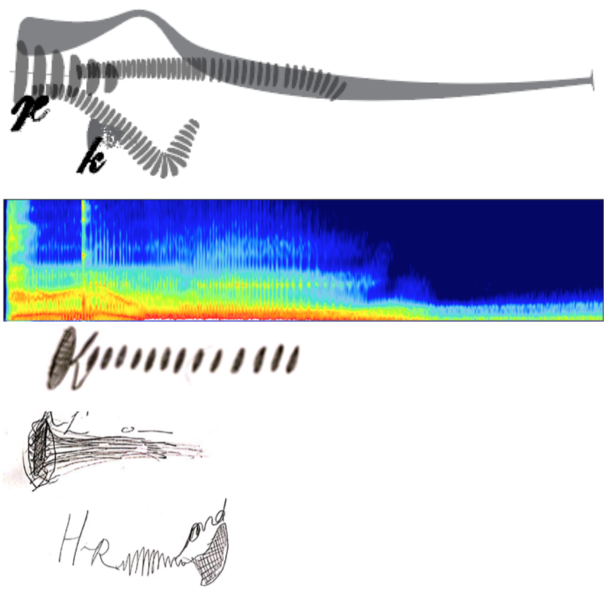


Figure 7. Sonogram of the opening of the piece (before the first entry of the singers) and the scheme distribution. This extract lasts a total of 12 seconds. The gestures result from the superimposed graphic schemes. In the temporal evolution of the spectrum we observe the sudden change in the formants produced by the occlusive phoneme “k”. The short, damped breaths are also visible.

The same text generates the entire next section (first section) where the song first appears: Figure 8 represents a rough drawing of a large gesture of affirmative speech which is the inspiration for the composition of the first section of the work (measures 2 to 43). This introduction, which lasts about 3 minutes, has a latent character, interrupted by the rhythm of short caesuras such as “OH”, a vocal gesture of exclamation (measures 2, 9, 12 and 19). The short cadence that begins from measure 24 concludes the 4 caesuras that preceded it.

Different types of gestures (voiced, voiceless, breathed, etc.) can be superimposed to create even more complex profiles (see Figure 9). In measures 43 and 44 for example, the fricative or plosive phonemes are combined with the constantly evolving vowels and nasals. Here again, a flow of vocal gestures is generated by the singers’ voices combined with the synthesized sounds. The texture of the piece is quite dense, unlike the beginning.

Sometimes, when two formant synthesis voices are superimposed while going through constant pitch and resonator modifications, a third virtual voice can be heard. Its material is a succession of harmonics from a fundamental tone, resembling East Asian overtone singing (measures 44 and 45). This further increases the density of the texture of this section.

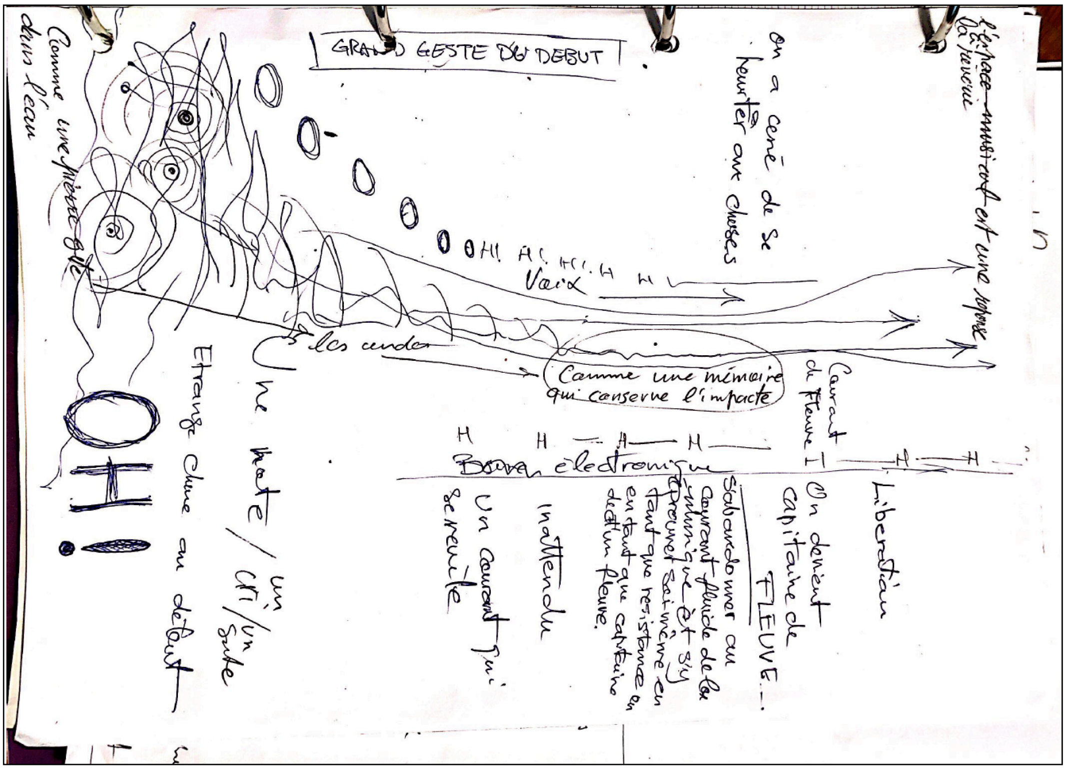


Figure 8. Graphic representation of a long gesture (the gesture that opens the work).

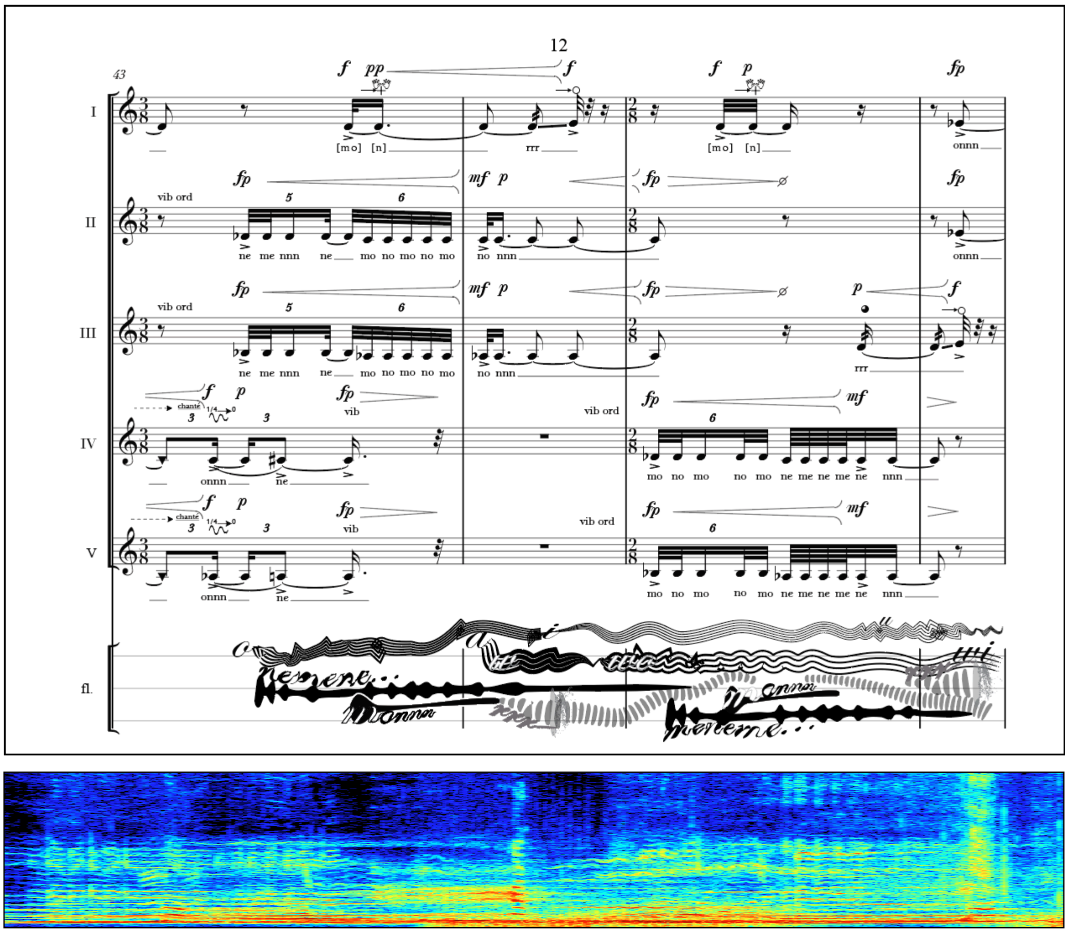


Figure 9. The combination of voiced and voiceless gestures generates more complex gestures.

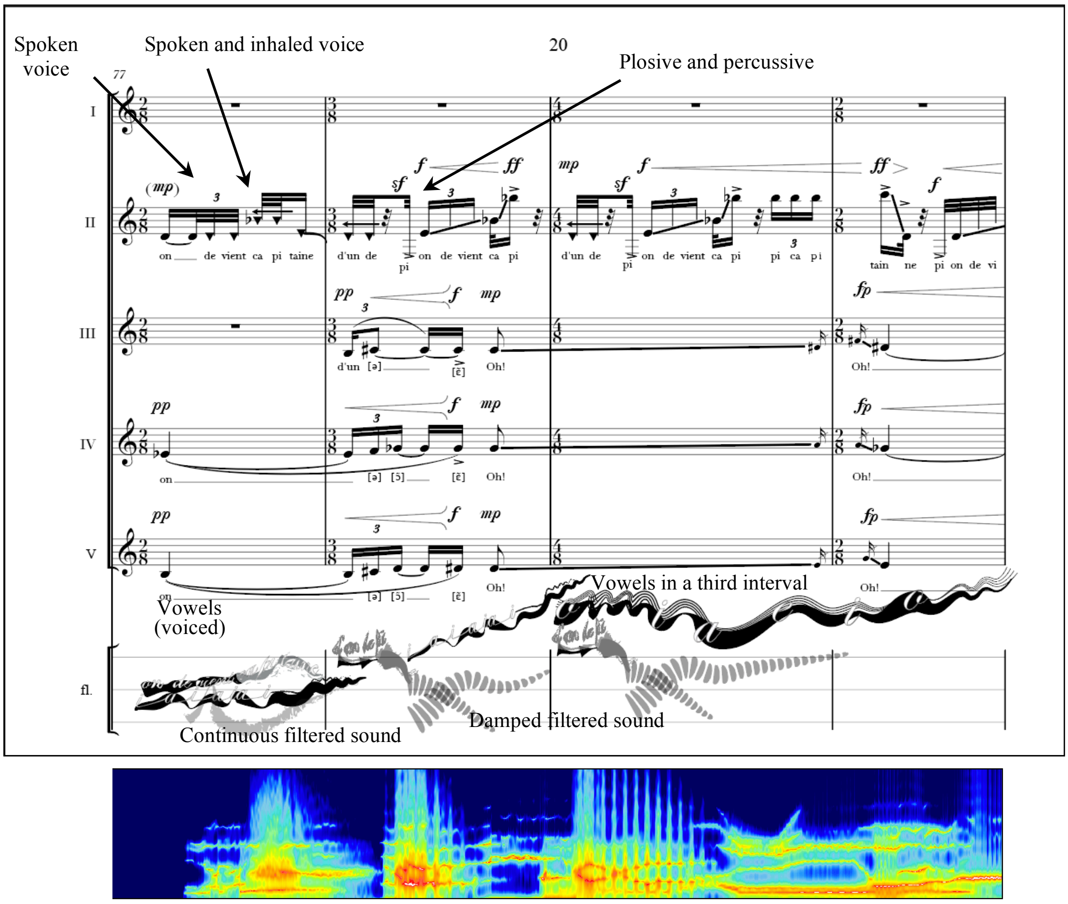


Figure 10. Measures 77 to 80, the sung voice combined with the synthesized sounds.

2.3 The textural progression and the harmonic outline of the work

Although the progression of the piece is based on changes in texture and timbre, rather than on harmonic relationships in the classical sense of the term, in some sections of the work the textural progression can be described through chords and relationships between intervals. The table below describes the formal progression of the piece.

The contrapuntal structure of the piece, quite discreet at first, progresses around a central pitch (B flat) and becomes increasingly dense. Starting from measure 25, the vocal part covers an interval that extends to an augmented fifth (see figure 9). The process continues and the gestures become shorter and shorter leading to reduced phrases. The texture's density increases and the gestural or phonetic units become shorter and increase in number.

Figure 10 is an example of complex matter in which the percussive phonemes of voice II are combined with the damped and filtered breath-like noise (pulse-train) of the synthesized voice, while the voiced gestures are augmented by a succession of synthesized vowels, simulated using resonators and morphing techniques.

In measure 82, voice II is in the high register, followed by the first brief appearance of a large chord in measure 75 (see Figure 11), which is the dominant color in the finale of the piece. In measures 81 and 82 (see figure 12), the range of the vocal part reaches a climax in the fortissimo dynamic. In measure 83, voice III (spoken) makes the transition from the loaded and tumultuous texture of measure 82 to a calm and more static texture. In measure 84, as in the previous measures, voice III pronounces "taine" with an inhaled breath prolonged by the sustained breath of the electronic sound. The word "d'un" in measure 85 puts an end to the voiced gestures of the synthesized sound and begins a new discourse whose material is the whispered voice and the damped breaths produced by electronic means.

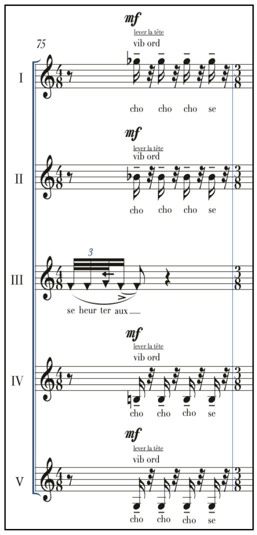


Figure 11. First appearance of the chord, a combination of voiced and voiceless gestures generating more complex gestures.

Figure 12. Measures 81 to 86, the sung voice combined with the synthesized sounds.

Figure 13. With the emergence of chords, the vertical temporality replaces the horizontal temporality.

At the end of the piece, with the return of the chord (see Figure 13) resembling a chorale, a vertical temporality finally replaces the horizontal temporality. The linguistic elements become more concrete and the text begins to surface in an intelligible way.

3. TECHNICAL ASPECTS

Voice description systems are usually based on a source-filter acoustic production model. The source produces a sound and gives it pitch or noise and power. The vocal tract acts as an acoustic filter and defines the timbre of the sound originating from the source. Each of these contributions play an important role in the spectral content and therefore in the phonetic information of each sound.

For the composition of *Mots de jeu*, a software called Chant was used to control the voice synthesis via the OM-Chant library in OpenMusic. OM-Chant provides for continuous control of the parameter settings for the different formants. Synthesis events can be fundamental frequency values of the pulse train (f0), FOF parameter matrices, noise generators or formant filters.

The role of electronics is to extend the human voice's timbre and technical performances. However, from a composition perspective, synthetic sounds must remain linked with the characteristics of the human voice, hence the use of formant synthesis. The modeled gestures therefore always remain linked with the vocal gestures of the singers. Different types of sounds and processing were used to produce the synthesized sounds: filtered or pulsed white noise to produce a colored breath (continuous, damped or granular) and periodic formants to produce the vowels. Further processing was added such as vibrato and morphing.

3.1 Filtered breath

Figure 14 shows an OpenMusic patch that generates a continuous noise. The noise parameters are controlled by an amplitude envelope and by inputs which can be used to adjust the total duration of the noise and its onset. Three filters can be applied to the source signal. Each filter contains two formants with two fundamental frequencies. The parameter settings for the number of formants, the onset, the duration and the amplitude of each frequency can be adjusted separately.

At the patch output, a synthesis engine generates the sound based on the input values. The source can produce a damped noise or a kind of staccato breath whose damping frequency is controlled by an envelope. The three filters mentioned above can be applied to the pulsed noise.

When the frequency is low, the sound generated resembles a pulsation. When the frequency value is increased, the sound rendered is similar to a voiceless alveolar [r] or a *jeté* or to flutter-tonguing on the flute filtered by independent trajectories.

This effect was often used to prolong the gestures resulting from the pronunciation of [r] by the singers (measure 50) or as a pulsed breath passing through the whispered voice in measures 87 to 90. Figure 15 shows the patch and the spectral content for this gesture. On the sonogram, the independence of the formants' trajectory is explicit. We can observe the change in frequency of the pulsation that follows the frequency envelope of the patch.

3.2 Vowels

As a general rule, the different vowels are simulated by adjusting the frequency, amplitude, and bandwidth values of three groups of simple sine waves (three formants). The

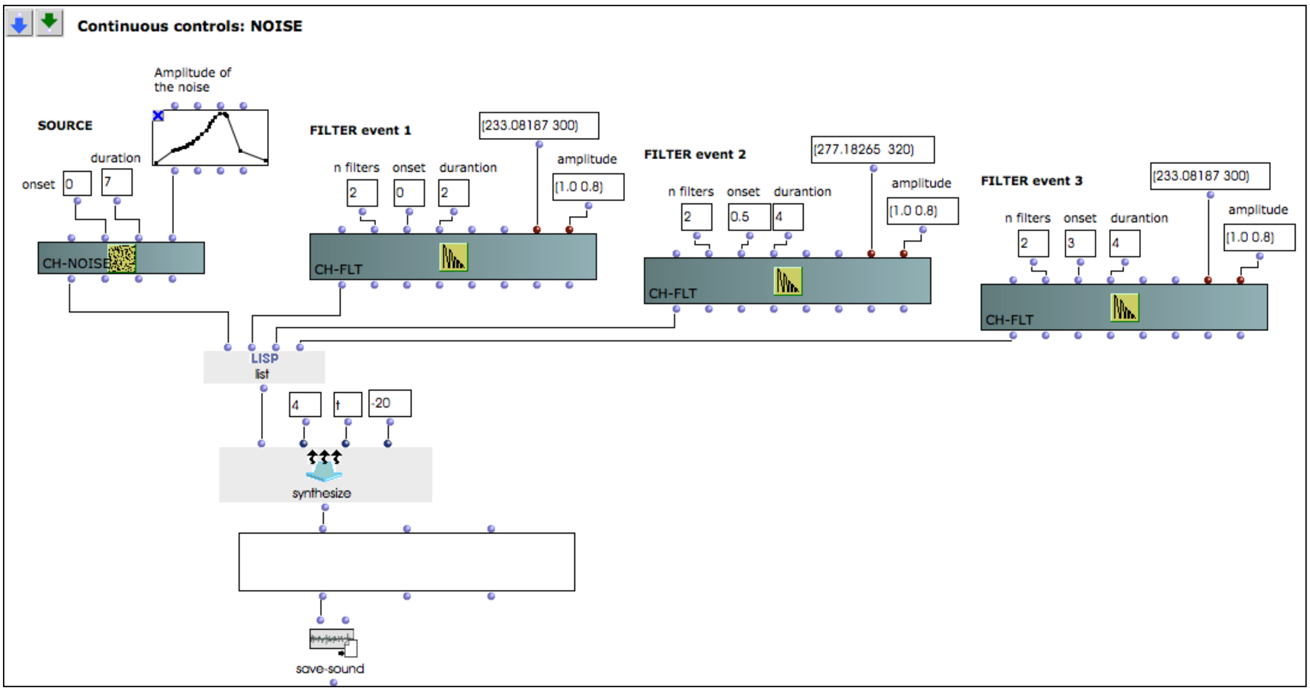


Figure 14. OpenMusic patch generating a continuous breath.

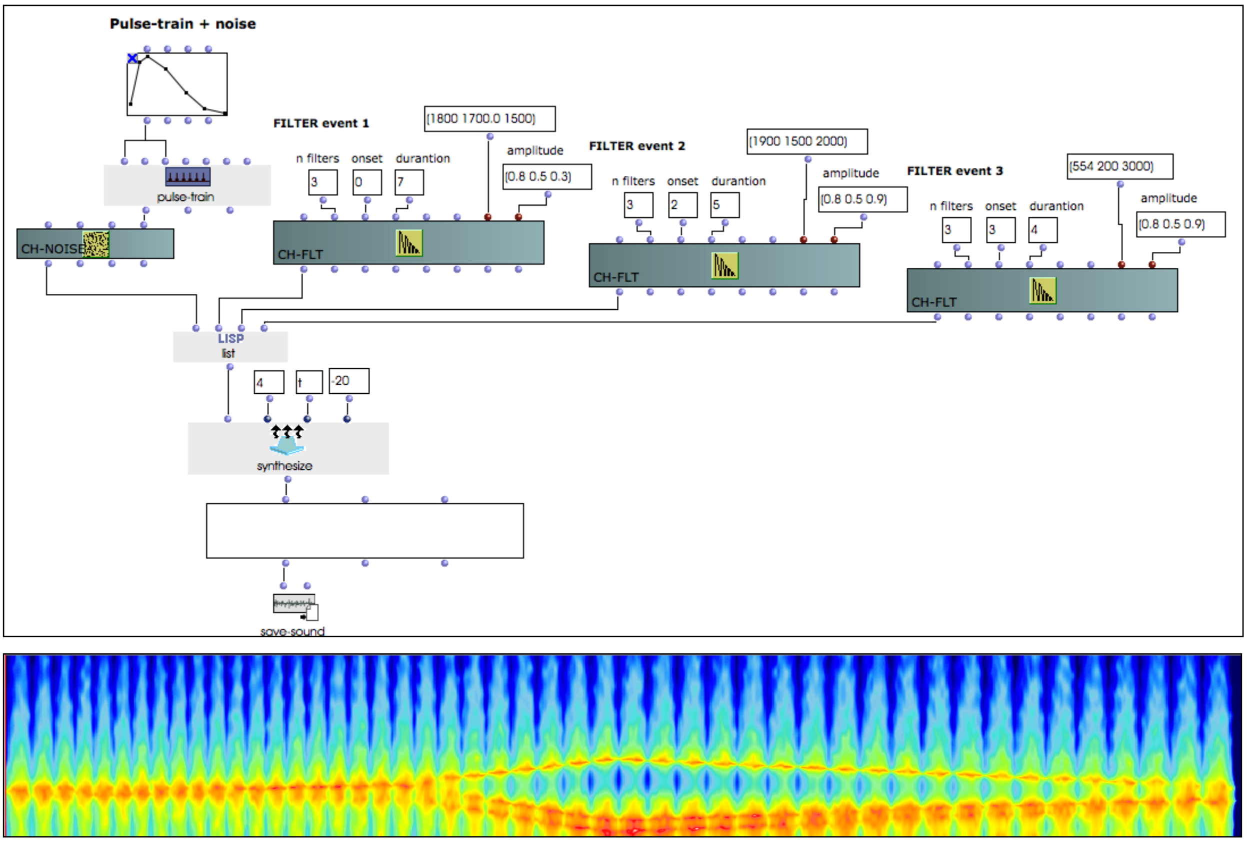


Figure 15. Patch generating a continuous breath with the settings for each formant. The sonagram below shows how the formants of the produced sound evolve.

trajectory of each of these values determines how the formant of the sound produced evolves, and therefore how the formant of the vocal gesture evolves.

3.3 Transition between phonemes

Compared with other signals, the speech signal requires more elaborate transitions. These sections show significant formant evolutions. Consonants or articulations (staccato, legato, etc.), for example, are expressive states which cause spectrum micro-evolutions that are essential for their phonetic and semantic profile. The word "ama", for example, represents the transition between two stable states (two vowels) via the nasal "m". Using the graphic representation of the formant paths (sonogram), we can observe the temporal behavior of the frequencies, amplitudes, and bandwidths, etc. between the phonemes. Figure 16 shows the evolution of the voice formants pronouncing "ana" (left) and "ama" (right). As can be seen in the sonogram, the profile of "M" is straighter. This can be explained by the fact that "M" has a more abrupt onset and offset than the phoneme "N", as can be heard by the ear.

The transition and morphing between voiced vowels and phonemes require an elaborate technical operation. The CH-TRANSITION function is used for the transition between phonemes. It adjusts the behavior of intervals when they overlap. Figure 17 represents a patch where the vowels (here O, EOE, I) make the transition between phonemes such as B or L. In the patch, the frequency band filters, the transition envelope, the fundamental frequency and the amplitude were used to adjust the parameter settings which determine the spectral content of the phonemes. By manipulating the envelopes, phonemes are produced which cannot be pronounced by the human speech production system.

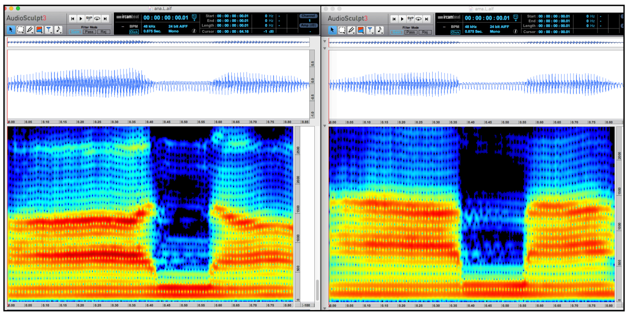


Figure 16. The sonogram resulting from the analysis of the phonemes "N" (left) and "M" (right) surrounded by the vowel "A".

3.4 Vibrato

Compared A vocal vibrato also requires a complex formant trajectory. The OM-Chant library allows us to adjust different parameter settings that generate vibratos that cannot be achieved by the human voice, while maintaining the voice's characteristics. Two of these parameters are speed (frequency) and amplitude.

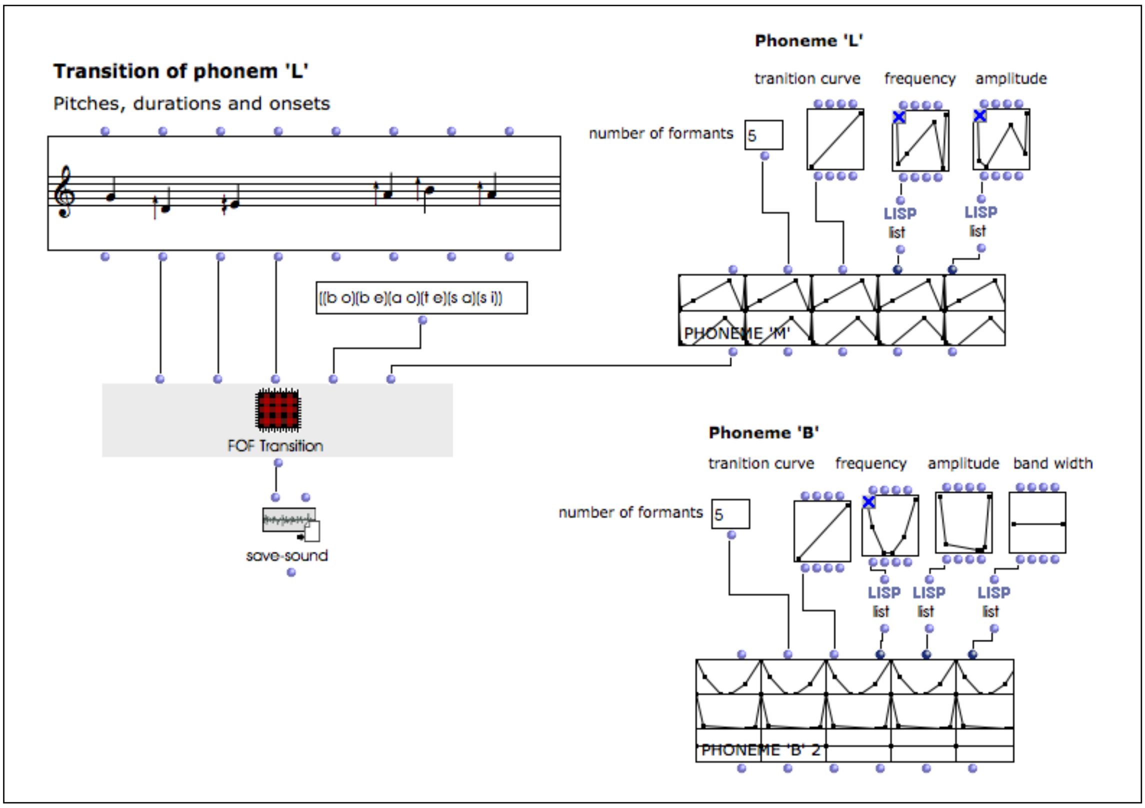


Figure 17. Phoneme succession patch with the transition controlled by frequency band filters.

3.5 Sound triggering

The electronic component of *Mots de jeu* was created entirely in a computer-assisted composition environment. None of the processing is carried out in real-time. Close to 500 micro-gestures were generated in OM-Chant then edited and spatialized (in stereo) in Logic Pro, reducing the number of edited sounds to 48. To trigger the sounds, a simple Max patch was designed. It can be activated either by the sound engineer in the sound room or by the computer music designer using a pedal or directly on a computer keyboard, in which case the electronic sounds follow relatively closely the temporality of the musicians.

4. CONCLUSION

Mots de jeu is born of the challenge Thanks to the human capacity to perceive the slightest changes in the spectral content of the voice and thanks to formant synthesis and visual control technologies, composers have a wide range of possibilities to explore material produced by the voice. *L'Espace du dedans*[6] is a series that draws from formant synthesis techniques while focusing on the man-machine relationship, on poetry and on new technologies. The composition *Mots de jeu* is an attempt to illustrate the poetry of Henri Michaux by conveying what language paradoxically is unable to express. Henri Michaux does not hesitate to express his frustration that words are trapped in what we are taught and what others would like to impose on us. For him, language imposes limits on beings and things. It forces the world into a grid and freezes meanings and identities.[7] Or, to use Gaston Bachelard's words, *Mots de jeu* somehow attempts to illustrate that "poetry is a metaphysics of the present moment […]. It is the principle of essential simultaneity where the more dispersed and disunited being achieves unity" [12, p.224].

While Michaux did not hesitate to invent new words which acoustically remained familiar to the ear of a French speaker, he would probably have been tempted to invent new phonemes and would have included them in his poetry, if he had had access to the technologies we have today. In this sense, *Mots de jeu* shares Michaux's approach.

5. REFERENCES

[1] X. Rodet, "Time-domain Formant-wave Function Synthesis", *Computer Music Journal*, 8 (3), 1984.

[2] J. Bresson, C. Agon, G. Assayag, "OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research." *ACM MultiMedia (OpenSource Software Competition)*, Scottsdale, AZ, USA, 2011.

[6] *L'espace du dedans* includes *Mots de jeu* (2018) for 5 female voices and electronics, *Chuchotements burlesques* (2019) for ensemble, an actor and electronics, and a piece for vibraphone and electronic chorus (2019).

[7] http://www.fabula.org/colloques/document1083.php >, date of consultation 21/09/2018.

[3] G. Assayag, C. Agon, J. Fineberg, P. Hanappe, “An Object Oriented Visual Environment for Musical Composition.” *Proceedings of the International Computer Music Conference (ICMC’97)*, Thessaloniki, Greece, 1997.

[4] C. Agon, *OpenMusic: Un langage visuel pour la composition musicale assistée par ordinateur*. PhD thesis, Université Pierre and Marie Curie (Paris 6), France, 1998.

[5] J. Bresson, R. Michon, “Implémentations et contrôle du synthétiseur CHANT dans OpenMusic.” *Journées d’Informatique Musicale*, Saint-Etienne, France, 2011.

[6] J. Bresson, M. Stroppa, M. “The Control of the Chant Synthesizer in OpenMusic: Modeling Continuous Aspects in Sound Synthesis.” *Proceedings of the International Computer Music Conference (ICMC’11)*, Huddersfield, UK, 2011.

[7] R. Foulon, J. Bresson, “Un modèle de contrôle pour la synthèse par fonctions d’ondes formantiques avec OM-Chant.” *Journées d'Informatique Musicale (JIM’13)*, Saint-Denis, France, 2013.

[8] N. Ruwet, *Language, music, poetry.* Paris: Éditions du Seuil, 1972.

[9] M. Beaufils, *Musique du son, musique du verbe*, Paris: Klincksieck, 2000 [New ed.]

[10] M. Chion, *Le Guide des objets sonores*, Paris: Buchet and Chastel, 1983.

[11] M. Villard. *Poétique du geste chez Henri Michaux : mouvement, regard, participation, danse*. Littératures. Université de Grenoble, 2012. <NNT: 2012GRENL009>.

[12] G. Bachelard, *Le Droit de rêver*, Paris: PUF, 1970.

A SCHEME FOR MUSIC INTERACTION DESIGN AND NOTATION BASED ON DYNAMIC SOCIOMETRY

Stefano Kalonaris
Music Information Intelligence Team
RIKEN AIP
stefano.kalonaris@riken.jp

ABSTRACT

A framework for musical interaction design and notation based on social network analysis is proposed. To this end, the affiliation network model, which comprises *actors* and *events*, is employed. Maintaining a sufficiently flexible definition of event can cater for both music improvisers and composers alike. If the (number of) events, their time occurrence and their action space (what happens in a given event) can be subjectively defined, then the concept of authorship and the continuum between improvisation and composition can be arbitrarily explored. The theoretical axioms of the affiliation network, along with methods for analysing its dynamics are presented. Furthermore, it is suggested that such analysis can provide a suitable strategy for notating emergent and/or composed musical interactions and, retrospectively, for designing some anew. Finally, a general scheme is illustrated, along with some speculative blends for its practical implementation.

1. INTRODUCTION

Networks can represent and study the interdependence, interaction and behavioural emergence of multi-agent systems, often a synonymous of complexity. Music ensembles, by virtue of the dynamical and complex interactions that define them, can also be considered as multi-agent systems, thus (socio-musical) networks. Improviser and researcher David Borgo, for example, states that "music, as an inherently social practice, thrives on network organization" [1, p. 11]. This characteristic has recently afforded ecological approaches and novel technological paradigms to music improvisation and composition, with contributions from practitioners, academics and researchers alike. In this paper a particular social network model, the affiliation network, is considered as a framework for both structuring and representing musical interactions.

1.1 Continuum

Musical composition is the strictest application of musical interaction, as it defines roles, times, content and expression for any given player in a music ensemble. Free musical improvisation sits at the opposite end of the spectrum, as it does not make a predefined commitment to any musical parameter, relying instead on the dynamic negotiation of the musical space and its organisation on behalf of the players, in real-time. Between these two extremes, there are countless hybrid tropes that blend improvisation and composition with varying degrees of integration, aimed at providing expressive opportunities and improvisational space to the players, while abiding by a top-down structural design. Amongst these, one can list *aleatoric* music [2] and *comprovisation* [3], although the contemporary and modern music composition practice abounds with examples of such blends. These, inevitably, require bespoke notation systems and methods.

1.2 Horizontal Time

There are innumerable approaches to notational systems developed to grant varying degrees of freedom to the performers. Amongst them, one could list proportional spatial notation (*e.g.*, Luciano Berio's *Sequenza I*), time-based pictographic scores (*e.g.*, Cage's *Waterwalk*), approximate pitch systems with (*e.g.*, tape-notation in Krzysztof Penderecki's *Threnody for the Victims of Hiroshima*) or without staves (*e.g.*, Schillinger graph-style notation [4]), altered (*e.g.*, Baude Cordler's *Belle, Bonne, Sage*) or specific notation systems (*e.g.*, Xenakis' *Psappha*), time-based abstract representation (*e.g.*, Hans-Cristoph Steiner's *Solitude*) or notation (*e.g.*, Rudolph Komorous' *Chanson*), free abstract representation (*e.g.*, Earle Brown's *December 1952*) or notation (*e.g.*, Mark Applebaum's *The Metaphysics of Notation*).

However, in the majority of the above, one cannot but notice the hegemony of linear time, which includes circular or periodic structures (*e.g*, George Crumb's *Songs, Drones, and Refrains of Death*). Temporal dependencies in (but not limited to) this particular musical domain are more often than not viewed as a serial procedure: event *A* happens before event *B*, and so forth.

Things do not seem too different in contemporary practices involving computer aided notation and/or composition for interactive musical performances. These include *screen scores* [5] and other scoring methods in the context of *networked performance* [6, 7] and laptop orchestras, sometimes referred to as responsive scores. Recent couplings of MaxScore [8] with node.js [1] have enabled be-

[1] URL: https://nodejs.org

spoke generation of scores for each client in networked performances comprising large numbers of players [9]. Nevertheless, systems as the latter, or such as *John* [10], *ZScore* [11], *Decibel ScorePlayer* [12] and so forth, are still primarily anchored to a horizontal viewpoint of time.

Motivated by the desire to break with a linear representation, notation and design of musical interactions, a method drawing from social network theory is proposed.

2. NETWORK THEORY

A network can be thought of as a systemic architecture within which elements connect and interact with each other. The bare minimum needed to describe a network is a definition of its topology and of its logical and operational affordances. In other words, it is necessary to know who/what is connected to whom/what, how these connections are formed or abandoned, and what are the eventual logical (or non) rules upon which these connections are contingent. A network can be represented as a system of connections (*edges*) between nodes (*vertices*). A basic understanding of network theory's terminology is assumed, however, the reader can refer to the Appendix, to this end.

2.1 Considerations

Many of the network model architectures developed in the context of graph and social network theory are difficult to port to a musical domain. The reasons are case-specific and embedded in their core axioms. For example, in the Erdös-Rényi random graph model [13], given a sufficiently large network, nearly all nodes will have the same *degree* (see Appendix). Furthermore, such model does not account for the instantiation of edges beyond randomness. Similarly, the Watts and Strogatz's *small-world* model [14], while it exhibits small average shortest *path* length (see Appendix), a large *clustering coefficient* (see Appendix) and addresses the absence of *hubs* (a node with atypically high degree), is still eminently stochastic. The *scale-free* model [15], on the other hand, abandons randomness as a potential way to explain large, complex social networks, thanks to the notion of *preferential attachment* [16]. However, a scale-free model is no less problematic when applied to music ensemble interaction or composition. In an ensemble of music performers and/or improvisers, the number of musicians is relatively small and each element (the musician) cannot be considered simpler than the system (the ensemble). Each musician is a complex decision-maker, evaluating strategies and responding not only to the local neighbourhood but also to the system as a whole. Furthermore, while it needn't be so, the prevailing norm in music performance and composition is to have a fixed number of players, while a scale-free network is dynamically growing and comprises a very large number of nodes (far exceeding the typical largest music orchestra). A candidate model for this paper's speculative framework is, instead, the affiliation network.

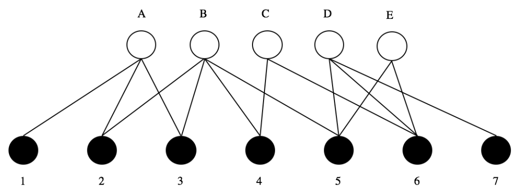


Figure 1. A bipartite graph representation on a two-mode network.

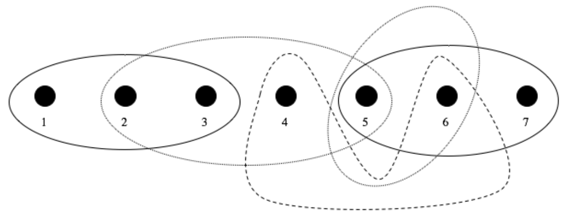


Figure 2. A hypergraph representation on a two-mode network.

3. AFFILIATION NETWORKS

An affiliation network, also called *two-mode* network, is described as "a network in which actors are joined together by common membership of groups or clubs of some kind" [17, p. 2570]. This model manages to introduce some non-trivial behaviours and a deviation from the low clustering coefficient of the random graph. A two-mode network can be represented as a *bipartite* graph or a *hypergraph*, as shown in Figure 1 and 2, respectively. Alternative ways exist to express the network concisely, without using a graphical representation, as it can be seen in the *incidence matrix* in Table 1. Rows represent the events and columns the actors, and their affiliation to any given event is expressed as a binary value. Based on the interaction strength, the edges in a bipartite network can be weighted, using a ratio scale (4 is twice 2, etc.). Figure 3 is an example of the same network with added weights.

Affiliation networks are relational and can show how actors and events are related, how events create ties amongst actors and how actors create ties amongst events. In general, affiliation networks can exhibit non-overlapping, nested or overlapping relations, as shown in Figure 4. Two-mode networks can include the synchronous existence of several events, which actors are free to choose from. This *modus operandi* can be useful for a musical interaction design that wants to expand and develop in time vertically, as well as horizontally (see Section 1.2). To design and notate musical interactions based on this model, it is worth describing its inner working a bit more in detail.

3.1 Measures and Metrics

The one-mode projection of the events (A, B, C, D, E) of the network in Figure 1 is obtained by constructing the 5-vertex network such that every event is connected to another if there is at least a member that participated to both.

1	1	1	0	0	0	0
0	1	1	1	1	0	0
0	0	0	1	0	1	0
0	0	0	0	1	1	1
0	0	0	0	1	1	0

Table 1. The same affiliation network in Figure 1, expressed as an incidence matrix.

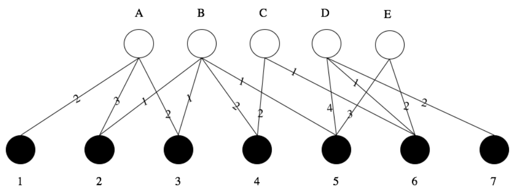


Figure 3. A weighted bipartite graph.

Conversely, the one-mode projection of the nodes (1, 2, 3, 4, 5, 6, 7) is obtained by connecting actors who have been part of the same event. The two one-mode projections are shown in Figure 5.

One-mode projections lose some of the information of the bipartite network, for example in the actors' projection, the number of events that any given two connected members have in common is not deducible. Things improve when using weighted edges, as shown above. However, neither hypergraphs nor bipartite graphs offer a comprehensive visualisation of the three layered structure consisting of actor-event, actor-actor and event-event. For this, Galois lattices have been proposed [18], but they are beyond the scope of this paper.

Despite the partial information loss, one-mode projections can be useful if one wishes, for example, to calculate shortest paths in the two-mode network, by simply projecting onto either actors or events and calculate path lengths as one would do in one-mode networks. A simple notion in a two-mode network is that of *co-affiliation*. An unnormalised measure of co-affiliation can be constructed by simply using a pairwise actor contingency table, as shown in Table 2, which refers to nodes 1 and 2 in the weighted bipartite graph of Figure 3.

In Table 2, the quantity a is a measure of the number of times that nodes 1 and 2 co-attended an event. To normalise this quantity, it is sufficient to divide a by n, which can be useful to compare other pairs of nodes. Alternatively, one can divide n by the $\min((a+b),(a+c))$, thus accounting for the maximum possible overlap given the number of events attended by both. Yet another normalisation would be to divide a by $(a+b+c)$, which expresses a in relation to the events that are possible to attend. Another important property in a network is that of *centrality*, which has been defined in a number of ways, for example based on *degree* (how active an actor is in the network), *eigenvectors* (if a central actor has ties with other central actors),

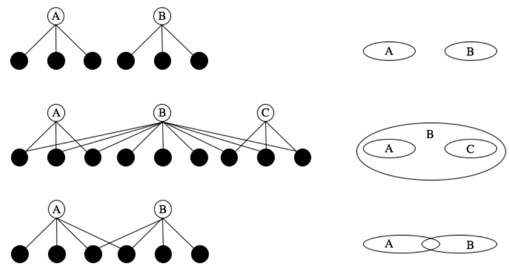


Figure 4. Non-overlapping, nested and overlapping relations (top to bottom).

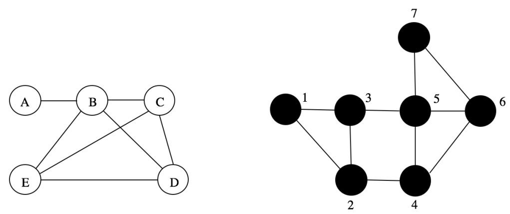


Figure 5. One-mode projections.

closeness (how short are the potential paths to other actors), and *betweenness* (potential to mediate between other actors).

4. SOCIOMETRIC ANALYSIS

The measures discussed so far can be useful for the sociometric analysis of the network. For example, one could deduce who are the most central actors or events, how these relate to one another, the eventual overlaps of memberships and in general capture inner structures and behaviours. When thinking in musical terms, the actors being players, these insights can be used retrospectively for the design of structured musical interactions or for notating interaction dynamics which can be conveyed back to the performers.

In [19] it is posited that there are three main approaches to representation used in responsive scores. These are based on low-level audio, mid-level performance data, and high-level score data, respectively. In the same paper, a fourth approach is proposed, leveraging on machine learning to classify latent "musical agents" based on information retrieval methods and *telemetric* data. It is suggested that "technology can help us navigate this unknown territory through the transmission and generation of vital information and create new performance perspectives." In this paper, in agreement with the last statement, yet another approach is put forward, based on sociometric analysis, instead.

4.1 Dynamics

The principal motivation for using an affiliation network is to afford concurrent events and group memberships. This

		Node 2		
		1	**0**	
Node1	**1**	2 (a)	0 (b)	2 (a+b)
	0	2 (c)	0 (d)	2 (c+d)
		4 (a+c)	0 (b+d)	4 (n)

Table 2. Co-affiliation of nodes 1 and 2 in Figure 3's network.

property opens up design strategies for a multiplicity of musical interactions, which remain available to the players. However, this does not mean that one can entirely disregard sequential dependencies. Rather, the two viewpoints can coexist in the same conceptual space. An actor might be part of several events at different times in the performance, whether sequentially or periodically, although several events might be available at the same time. In a musical context it is reasonable to assume that a player can only be part of one event at the time, unless placing severe constraints on attention and focus which would reflect negatively on the quality of the music played. Therefore, to achieve dynamical re-configurations of the bipartite graph, a notion of sampling could be introduced. The rate at which players make changes in their outgoing ties could be also used as a further measure, *e.g.*, a 'rate function'. These repeated network snapshots could be regarded as discrete observations of a process developing in continuous time, where actors make changes in their evaluation of the state of the network, and constitute each other's changing environment. At each sampling step each player controls his/her membership and within each sampling period he/she (potentially) controls his/her behaviour in relation to the *clique* (in the current analogy: the musical cluster). Such discrete observations could be used to compute the measures described in Section 2.3, as well as for generating a time-explicit graph for visualisation purposes. Figure 6 illustrates two time steps in a hypothetical affiliation network (with only two events). Measures such as degree have to be redefined in relation to the network model. If considering a "one-mode" representation, then one will have measures for actor degree and for event degree. For an actor i and an event j these will be, respectively, the number of different actors who participate to the same event as actor i, and the number of different events which share an actor in common with the event j. As an example, Figure 7 shows the evolution of four player's degree over time [2]

5. SCHEME

The proposed framework is medium agnostic, and can be realised in whichever format is most congenial to the designer. For example, graphic scores could well be employed, as would any arbitrary blend of traditional notation techniques, abstract scores and technology-based representations. However, since we would like to benefit from the sociometric analysis described earlier, the specific

[2] This data is taken from one of the author's system's during a real performance.

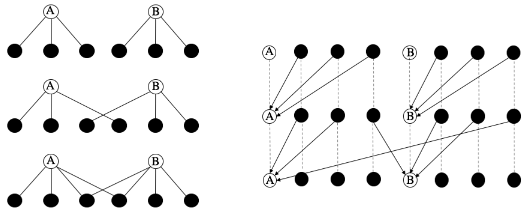


Figure 6. Cumulative graph after two periods (left) and corresponding time-explicit graph (right).

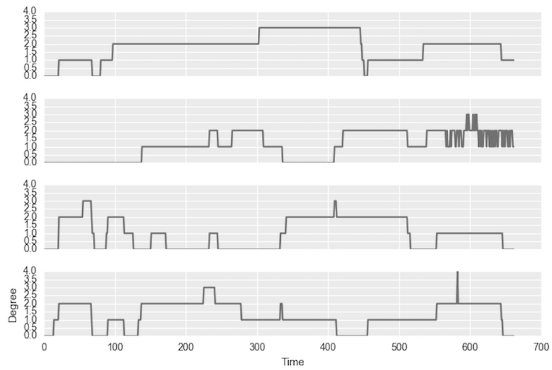


Figure 7. Example of four player's degree over time, from a "one-mode" perspective.

choice would depend on having a way for actors to route membership decisions back to the network. To this end, computer-aided systems, and in particular the networked performance paradigm, whereby the design and/or compositional parameters (*e.g.*, the timeline of scheduled event, the content specifics, etc.) are shared over a co-located or remote computer network, are an obvious choice. Practically, the system could be configured as a many-to-one topology, where each player receives and visualises the available information about the affiliation network state on his/her terminal. A central server would host the 'score' or the sequence of available events that are presented to the actors, along with arbitrarily complex descriptions of what any given event musically entails. Affiliation decisions' routing could be done in many different ways, for example, a foot-pedal array, a numerical keypad, a graphical user interface, and so forth. The server could thus perform some of the analysis discussed in Section 4 and this information could feedback into the system, at some level. For example, if it was the case that a particular event was consistently poorly attended, the system could choose to replace it [3] with a suitable alternative.

In exploring this model speculatively, two extremes can be considered to illustrate the potential flexibility of the system. In the first case, the framework would be used by ensembles of improvising musicians, thus the specifications on both actors and events would be assumed to be minimal and/or consensually agreed amongst the constituent players. In the second case, the framework would be used by composers who can have complete agency over

[3] stochastically or according to specific rules.

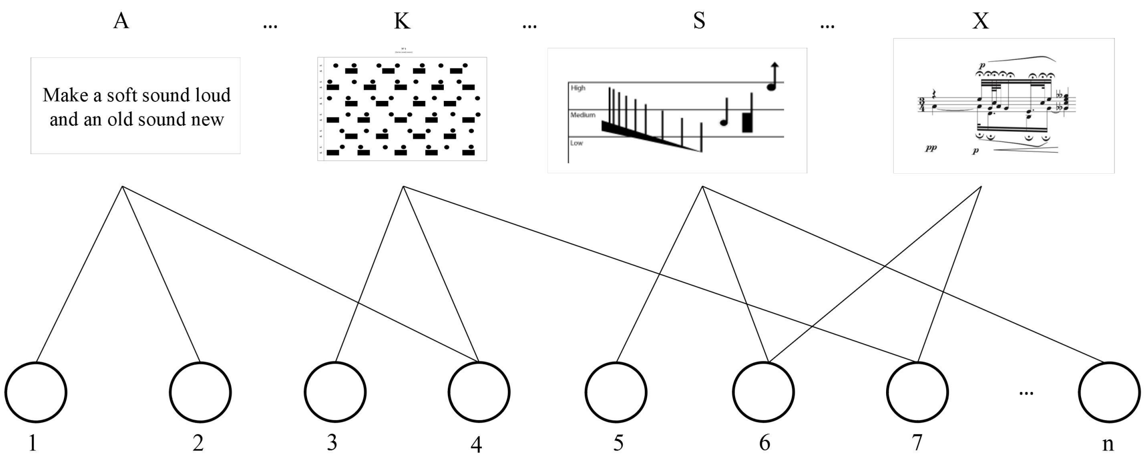


Figure 8. A speculative score, for illustration purposes only, with events ranging from abstract/aleatoric to conventional notation. Event *A* is borrowed from Oliveiros' *Klickitat Ride, 108 possibilities 54 opposites*, in [20]. Event *K* is Kirkpatrick's The Book of Musical Patterns No1 [21]. Event *S* is an example of relative pitch notation [22], and event *X* is a detail from Verheul's Nocturne No.14 [23].

both the actors (*e.g.*, orchestration) and the events (*e.g.*, event 1 = play specified musical material/score/ideas, event 2 = . . .). However, this very case would collapse the affiliation network into a sequence of pre-established memberships to groups on behalf of given actors. Notwithstanding this caveat, all continuous options in between free improvisation and highly constrained network parameters could be considered. Whatever the case, it is assumed that actors are free to join any given event available. This affordance will be dependent on how the time sampling is implemented, whereby, at given time occurrences, the players are presented with the opportunity to change their group membership. Of course, it is possible to introduce arbitrary flexibility regarding the sampling period. For example, this could be stipulated a priori by the composer (*e.g.*, regular or periodic time frames, time-line score, etc.) or negotiated in real-time by the improvisers, for example by means of a voting system or by cueing. Similarly, the events available at any given sampling step could be invariant or changing (both in action space and in number). These options could be chosen/fixed by the composer, stochastically determined (*e.g.*, using a probabilistic automaton), or even negotiated as above. In general, the network dynamics could depend on endogenous (*e.g.*, intrinsic in the design or resulting from reciprocity, transitivity, etc.), exogenous (*e.g.*, actor variables) and dependent (*e.g.*, ego-by-proxy, network position of strongest personalities) factors. The attendance to a given event in a musical framework of this type will then likely be a function of the willingness to cooperate with a given set of players (for a wide range of motivations) and of the appeal that a given event has for a player. Regarding the events' content, it is ultimately up to the end users or music designers to make specific choices in this regard. Figure 8 illustrates some of these options along the continuum, from abstract to notated. The overall scheme derived from this speculative application of affiliation networks to musical interaction is thus very simple, and can be summarised as a parametric space spanning a continuous domain ranging from designed to emergent, as seen in Table 3.

SAMPLING	EVENTS			ACTORS		
	Number	Content	Membership	Individual	Sectional	Mixed
Fixed						
⋮						
Mutable						

Table 3. A simple scheme.

6. CONCLUSIONS & FUTURE WORK

Graph models can be useful for exploring modalities of musical communication, interaction and creation, whether more composition-oriented or more improvised. However, to harness the potential of the network's notion in the context of finite music ensembles (which normally do not grow or shrink arbitrarily during the course of a given performance), considerations regarding structural and dynamic characteristics of the chosen graph model are paramount. Affiliation networks can offer an interesting viewpoint in that they allow concurrent options for the players to choose from. Thus, they challenge the well-established paradigm of sequential time normally used for interpreting, experiencing, and designing or composing musical interaction. Providing that non-intrusive and seemingly integrated ways to route membership decisions are implemented in this context, sociometric data analysis can be fed back into the network, thus injecting the musical interaction design process with real-time opportunities to morph and adapt, if one so wishes. However, since this model is purely speculative at this point, it remains to be seen whether or not it constitutes a valid scheme that can be flexible enough to accommodate a wide range of musical organisation level needs. Thus, the author endeavours to implement a working prototype in the near future and to test it in a real-performance environment.

7. REFERENCES

[1] D. Borgo, "Sync or Swarm: Musical Improvisation and the Complex Dynamics of Group Creativity," in *Algebra, Meaning, and Computation: Essays dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, K. Futatsugi, J.-P. Jouannaud, and J. Meseguer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–24.

[2] J. Cage, *Silence : lectures and writings / by John Cage*. Calder and Boyars London, 1968.

[3] S. Bhagwati, *Notational Perspective and Comprovisation*. Leuven, Belgium: Leuven University Press, 2013, pp. 165–177.

[4] J. Schillinger, *The Schillinger System of Musical Composition*, A. Shaw and L. Dowling, Eds. New York: Da Capo, 1977.

[5] C. Hope and L. Vickery, "Screen Scores: New Media Music Manuscripts," in *Proceedings of the International Computer Music Conference (ICMC)*, Huddersfield, UK, 2011.

[6] A. Carôt, P. Rebelo, and A. Renaud, "Networked Music Performance: State of the Art," in *Audio Engineering Society Conference: 30th International Conference: Intelligent Audio Environments*, Saariselkä, Finland, 2007.

[7] R. Mills, *Telematics, Art and the Evolution of Networked Music Performance*. Cham: Springer International Publishing, 2019, pp. 21–57.

[8] N. Didkovsky and G. Hajdu, "Maxscore: Music Notation in Max/MSP," in *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, Northern Ireland, 2008.

[9] G. Hajdu and R. Gottfried, "Networked Music Performance in the Old Elbe Tunnel," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*. Melbourne, Australia: Monash University, 2019, pp. 55–60.

[10] V. Goudard, "John, the Semi-Conductor: A Tool for Comprovisation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*. Montreal, Canada: Concordia University, 2018, pp. 43–49.

[11] S. Zagorac and P. Alessandrini, "ZScore: A Distributed System For Integrated Mixed Music Composition and Performance," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*. Montreal, Canada: Concordia University, 2018, pp. 62–70.

[12] A. Wyatt, L. Vickery, and S. James, "Unlocking the Decibel ScorePlayer," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*. Melbourne, Australia: Monash University, 2019, pp. 61–68.

[13] P. Erdös and A. Rényi, "On Random Graphs I," *Publicationes Mathematicae Debrecen*, vol. 6, p. 290, 1959.

[14] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, no. 393, pp. 440–442, 1998.

[15] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[16] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, "Structure of Growing Networks with Preferential Linking," *Physical Review Letters*, vol. 85, pp. 4633–4636, Nov 2000.

[17] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 1, pp. 2566–2572, 2002.

[18] L. C. Freeman and D. R. White, "Using Galois Lattices to Represent Network Data," *Sociological Methodology*, vol. 23, pp. 127–146, 1993.

[19] R. Wilson, "Towards Responsive Scoring Techniques for Networked Music Performances," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*. Melbourne, Australia: Monash University, 2019, pp. 46–54.

[20] P. Oliveros, *Anthology of Text Scores*. Deep Listening Publications, 2013.

[21] R. Kirkpatrick. (2006) The Book of Musical Patterns. Accessed: 18 March 2020. [Online]. Available: http://www.spiralcage.com/hollowearthrecordings/bomp/BoMPpdfs/TheBookofMusicalPatterns.pdf

[22] VoxNovus. Relative pitch staff. Accessed: 18 March 2020. [Online]. Available: https://commons.wikimedia.org/wiki/File:Relative_pitch_staff.png

[23] M. C. Verheul. (2013) Nocturne no.14. Accessed: 18 March 2020. [Online]. Available: https://imslp.org/index.php?title=Nocturne_No.14_(Verheul,_Maurice)&oldid=2200777

Appendix

The *degree* of a vertex is the number of edges ending in that very vertex. *Directed* and *undirected* graphs are defined based on whether the edges can be traversed in both directions or not, respectively. *Walks* are ways to get from one vertex to another, for example, a walk $(ahkjs \dots z)$ is a walk between *a* and *z*. A *path* is defined as a walk where all edges and nodes are different and a *cycle* is a closed path. The shortest path from one vertex to another is called a *geodesic path*, whereas the *average distance* is the average of the minimal path length between all pairs of vertices. The *coordination number* is the average degree in a graph, and the *network diameter* represents the maximum degree of separation between all pairs of vertices. In other words, is the longest geodesic path between any two vertices. A *clique* is a fully connected sub-graph. More formally, a clique is a set of nodes where every node is connected to every other in the set and where no node outside of it is connected to all the nodes that are members of it. Further essential definitions include the *clustering coefficient* of a vertex, defined as the average ratio between the vertex's degree and the number of neighbours that are also connected to each other, and the degree distribution p_k which, for a graph with N nodes and X_k being the number of nodes having degree k, is equal to $\frac{X_k}{N}$. A *complete graph* is such that each pair of vertices are adjacent, which means there is an edge joining them and the vertices are incident with such an edge. Finally, two graphs are *isomorphic* if there is an injective mapping (one-to-one) between the vertices on one graph and the vertices of the other, such that the number of edges linking any two vertices in one graph is equal to the number of edges linking the corresponding vertices in the other graph.

JAM TABS: A COLOR BASED NOTATION SYSTEM FOR NOVICE IMPROVISATION

Torin Hopkins
ATLAS Institute, University of Colorado, Boulder
Torin.Hopkins@colorado.edu

Ellen Yi-Luen Do
ATLAS Institute, University of Colorado, Boulder
Ellen.Do@colorado.edu

ABSTRACT

Music improvisation, which is defined as creatively and playfully exchanging musical concepts, is often associated with expert musicianship and musical skill. While there are numerous methods for live scoring improvisation, there are not many resources aimed at assisting novice performance during improvisational activities. Jam Tabs is a notation system that helps coordinate idiomatic musical expression for novice musicians using color-coded instruments and a colored visual display.

Jam Tabs utilizes a seven-color notation system, LED cubes and a colored piano keyboard to assist novices in coordinating chord progressions—a common musical element used among improvising musicians in popular western music contexts. We have observed that novice piano players could follow chord progressions in tonal music note for note using color coordination while improvising. We found that seven colors in any key provide enough information for players to experience a satisfactorily expressive and creative jam session.

1. INTRODUCTION

1.1 Improvisation in Music Notation

1.1.1 Music Notation

Music notation is defined as a visual notation containing images, symbols or codes used to represent musical sound or instruction for performance [1]. Though music notation has been used to depict many aspects of music throughout history, symbols that place emphasis on the relationship between pitch and time are common in popular western music contexts.

Music notation can be differentiated from music visualization, which is often expressed as a representation of music in real-time—produced with emphasis on aesthetic value, or to depict an aspect of the music. Music notation in the context explored, however, conveys information that allow players to refer to a common resource, synchronizing musical activity between musical agents.

Figure 1. Example of a lead sheet for Jazz style improvisation.

1.1.2 History of Improvisation in Music Notation

Throughout world history music notation for improvisation has been utilized to convey broad or generalized musical information. One of the first examples of music notation, discovered in ancient Sumer around 1250 BC, contained: a poetic text of hymnic character, some lines of musical notation, a symbol specifying the genre of composition, the instrument tuning, and the composer [2]. It can be argued that these ancient scores were intended for improvisation rather than note for note replication of a piece—common in written scores of today. In the 13th century, and likely before then, India's Raga music, a traditionally improvised classical music, was written and performed in the eastern world [3]. Also of note, western music composers like Mozart and others in the eighteenth century, utilized figured bass to indicate the underlying musical content for improvising over a cadenza [4]. In more recent years, musicians of many genres use lead sheets, chord charts, and other abstract notations to show only the essential components of the underlying musical material.

1.2 Current Use of Notation in Improvisation

Traditionally, jamming is a term used to describe a musical improvisation activity, implying real-time musical creation, often with an underlying theme or familiar tune.

Though jamming is more popular in genres such as jazz and blues music than in many other western music traditions, it is nevertheless a skill available to musicians in all genres of music [5]. A common form for representing music, both for performance and for jam sessions, is the lead sheet. A lead sheet usually consists of the chord root and quality contained within a bar of a given number of beats (see Figures 1 and 2). A lead sheet may also contain written music, but it is not an essential component of the notation. Additionally, instead of the letter name for a chord, a roman numeral is given instead to represent the chord in reference to the key or tonal center of the song (see Figure 3).

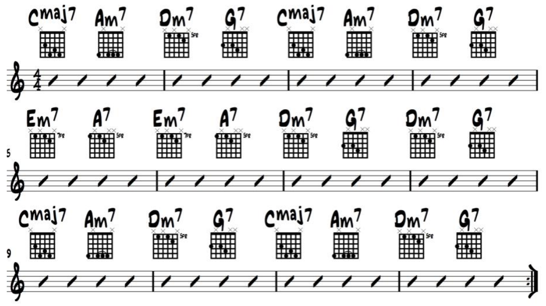


Figure 2. Example of how a lead sheet may appear. Contains chord charts or diagrams

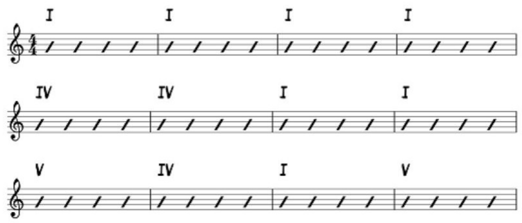


Figure 3. Example of a lead sheet for Jazz or Blues style improvisation may appear. In this example roman numerals represent harmonic content

1.3 Music Notation for Improvisation

The traditional western music score has formed the basis for many other, less formal, western music notations. Many of these notations are used among musicians in recording studios, jam sessions, gigs and during practice for the purpose of creating flexibility in their musical output. Music improvisation, which entails some degree of freedom during a performance, requires less information in a notation than a musical piece that is more prescriptive and explicit in its intended expression. Therefore, music notation for improvisation is uniquely concise in its use of symbols—often displaying only a few musical descriptors.

1.4 Act of Improvisation

When improvising, the musicians in a group will often play notes that are present within the chord being played on the lead sheet. More advanced musicians may be able to incorporate other elements of music present in their improvisations, but nevertheless having access to this information while participating in a jam session or while practicing improvisation over a backing track is helpful for many improvisers. In order to fully utilize a lead sheet a musician must know:

1. What the symbols on the lead sheet refer to.
2. Where the notes are that are being referred to on their instrument.
3. Some musical vocabulary to use while referencing the music being played.

Depending on the instrument being played this task can be very difficult to achieve. Musicians improvising in real time can have timing constraints at approximately 30 milliseconds before sounding asynchronous [6, 7]. In the case of the piano keyboard, the task requires a minimum of 12 notes to be recognized from a pattern of black and white keys. If a note, or series of notes, is to be played at 60 beats per minute, in common time (four beats per bar), with a chord change occurring every bar, an improviser would have to locate a new and unique note on the piano every four seconds in order to reference this basic musical concept in the chord progression. An improviser who does not have knowledge of the keyboard would not be able to play notes in the chord progression with just a lead sheet and a piano keyboard. Therefore, without instruction or prior knowledge, an improviser would not have enough information to begin an improvisational activity.

1.5 Improvisation Pedagogy

Improvisation in music education has been researched in numerous western music genres. Some Jazz and Blues music educators have created exercises to bolster improvisation skills by breaking down solos, trading rhythmic clapping, scat singing, and utilizing multiple software tools to support the improvisation activity [8, 9]. There is also a rich history of improvisation pedagogy relating to cadenzas in classical music. For this type of improvisation education, musical passages are selected from famous classical improvisers, such as Mozart, and practiced in various ways—often in an effort to replicate the composers style without playing the music verbatim [4].

Stages of improvisation education have been outlined by Kratus for optimizing the growth of the improviser [10, 11]. These stages of learning can be best categorized as being a multi-leveled system with tasks that increase in behavioral complexity [12]. Pedagogical techniques in this system separate aspects of the music for isolated training, such as rhythm, style and technique. According to Huovinen, improvisation pedagogy can also be broken down into two methods: a music theoretic approach, and the other being a 'dramaturgical' approach. In this view the music theoretic approach focuses education on the chords and scales that an improviser should be playing during the improvisation, and the dramaturgical focuses on balance, variation and tension [13]. When analyzed for differences in improvement for variation in play, both methods have shown to increase complexity of musical concepts while improvising. Those who are taught using a music theoretic approach demonstrated higher variability in dissonance, while those taught using the dramaturgical method demonstrated an increase in rhythmic variability [13].

Extra-musical factors can also play a significant role in the effectiveness of improvisation pedagogy. Studies in children have shown that while learning to improvise, social context plays a critical role when ascribing meaning to improvisation. Learning how children ascribe meaning to improvisation can influence how they perceive what they are doing and how they can work to improve upon it [14].

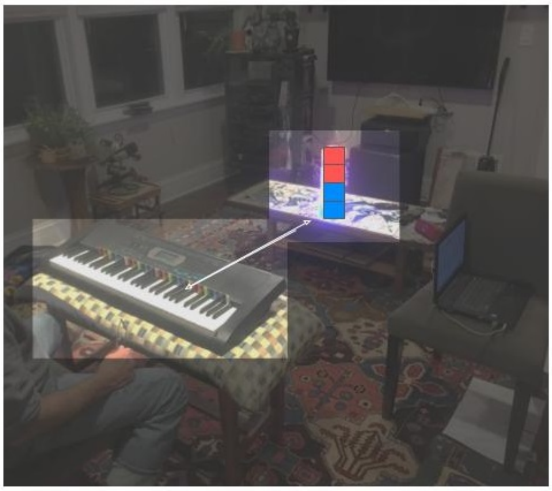


Figure 4. Jam Tabs system containing notation and colored piano keyboard.

1.6 Color in Music Notation and Instruments

The use of color in music has been used many times over the course of history. Colored notation has been utilized by composers such as, John Cage, Olivier Messiaen, Alexander Scriabin and Gyorgy Ligeti to enhance the musical content of their scores in a variety of ways [15]. The use of color has also been demonstrated to make music notation easier to read for young children [16] and has been used in piano instructional books from many popular publishers. These instructional books are often accompanied by a set of stickers that are to be applied to the instrument. A variety of stickers can be found in stores, including: black and white letters of the musical alphabet, pictures of musical notes as they appear on a score, and colors of various shapes.

More recently, color and light have been used to supplement music education by lighting notes that are to be played on the lit instrument. Piano companies such as Casio® and Yamaha® have lit keyboards, and guitar companies such as Fretlight® have LED embedded guitar necks to guide learning guitar players to the correct notes.

2. MOTIVATION

Skilled musicians often describe being in a pleasurable state, achieving flow [17] and bolstering creativity [18] when improvising. While there are numerous methods for live scoring improvisation [19, 20], there are not many resources aimed at assisting novice performance during improvisational activities using notation. Many resources are available for players that have an understanding of scales, chords, time signature, and their instrument. These visualizations often display various chords or scales while a musical piece is played and are aimed at an audience that has already spent a significant amount of time with their instrument.

Several resources are available for novices by augmenting instruments so that they are easier to play [21, 22]. Many more resources are available for novices to experience music in novel ways through collaborative musical instruments and interactive musical systems [23, 24, 25, 26]. These methods, however educationally useful, can neglect desirable aspects of participating in a traditional jam session, such as playing a traditional instrument or maintaining control of ones musical contribution.

The proposed system was designed to coordinate players of both traditional and non-traditional instruments as well as accommodate music improvisation at more difficult levels of play. The system was designed to assist in making collaborative musical experiences easier for novice musicians of any instrument by providing an optimal amount of visual information.

The following design problems were identified as the most pressing issues to address in novice music improvisational settings:

1. Providing information regarding the root note of a given chord on a visible notation.
2. Providing a series of root notes that clearly represent an order or progression.
3. Demonstrating a time signature, or how many beats are present per chord.
4. Including features that allow musicians to recognize where in the progression they are at any given time.
5. Providing a reference to the instrument itself in order to aid in the interpretation of the notation element.
6. Maintaining visibility of the notation from any angle in the jam session allows musicians to focus attention on a single common notation when in a group.

Information about the chord progression ties all members of the jam session together and is often the main content of a lead sheet. This information is available to musicians either visually or auditorily. Aural information, however, does not persist in time. An intermediate level of aural skill (auditory musical element recognition) is generally required in order to follow along in a jam session. This observation coupled with the fact that current systems of visually displaying music are beyond the interpretation of a novice player, suggests that a new method of visually displaying relevant chord progression information is needed in order to increase the number of novice musicians who can participate in improvisational activities.

This paper presents a system that addresses these design constraints and also provides additional benefit to the novice improviser. The design solution was developed through numerous observations of jam sessions using various relevant technologies.

3. JAM TABS DESIGN PROCESS

In order to make improvised musical activities less stressful and creatively productive for novice musicians, Jam Tabs utilized a seven-color notation system, addressable LED lights inside of an acrylic cube (LED Cube, see Figure 10) and a color-coded piano keyboard to coordinate the essential content of a lead sheet—the chord progression, a

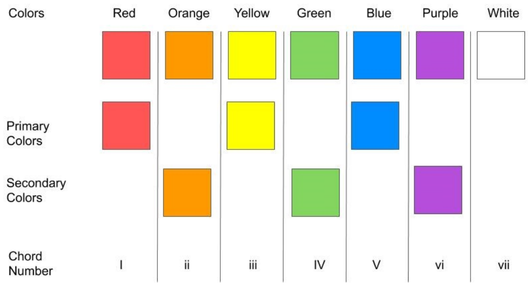


Figure 5. Color selection and reference to relevant harmonic content

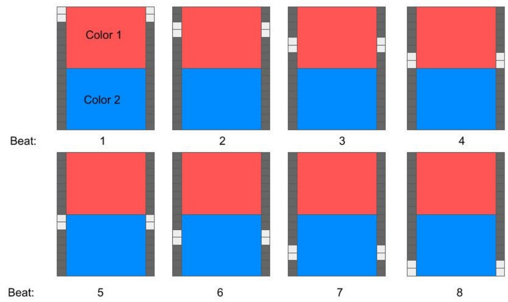


Figure 6. Beats per minute demonstrated by a moving LED on either side of the colored face of the cube. Number of lights per color/section indicate the number of beats per bar of music. This example indicates a 2-chord progression (I-V), each played for a duration of four beats (common time) at a given tempo.

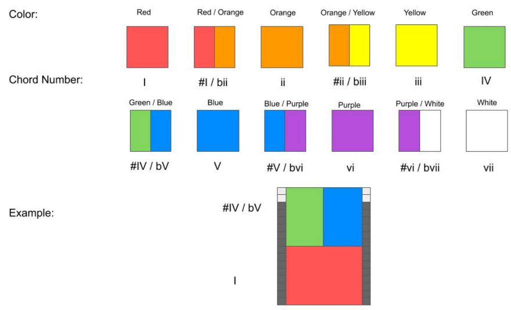


Figure 7. Demonstrates how accidentals are portrayed in color. Accidentals (sharps and flats) are shown by splitting the neighboring colors in half and merging them into a single section.

common musical element used among improvising musicians. A color-coordinated notation in addition to color-coded instruments enables novice musicians to quickly identify the salient notes in a chord progression on their instrument (see Figure 4).

3.1 Color Selection

Colors for the LED cubes were selected in order to provide seven distinguishable colors to display. Primary and secondary colors were used to distinguish the first six colors. The seventh color chosen was white also because it is easily distinguishable from the other six (see Figure 5). Additionally, by using the sequence and colors of the rainbow, an inherent order is given. Doing so enabled the LED cubes to be easily distinguished from each other while simultaneously providing an order to the colors. Pitch information is notated through color because it does not have spatial restrictions. This allows color to be used on the cubes and any visual instrument. Color also has the capability, if not confined to a shape, to be omnidirectional.

3.2 Time Signature

The LED cubes, similar to lead sheets or chord charts (see Figure 2), indicate a full measure of music. The cubes indicate a time signature (number of beats present per cube in a bar of music), through a series of neopixels located on the edges of the cube (see Figure 6). For example, if the chord progression is in common time (four beats/bar) the neopixel strip will show a white light that moves from one neopixel to the next lower neopixel. This occurs four times in one cube and then the following series of lights in the cube initiates.

3.3 Accidentals (Sharps and Flats)

Notes that are outside of a given key are displayed as a combination of the adjacent colors. Each cube is separated into two parts in order to retain the capability of showing the seven colors used to represent the major scale as well as the five notes that are in between the seven notes (see Figure 7).

Splitting the bar had two notable benefits. First, increasing the number of colors to twelve spread the spectrum of color and took away from the ability to distinguish them. Second, the colors that are found in between are perceived as being some combination of the surrounding notes. This is similar to the already familiar concept of sharps and flats in classic music notation. This, among other features, allows for players to recognize notes that fall outside of the key quickly and easily. For these reasons, only seven colors were included in the final design.

3.4 Backing Tracks

A jam or backing track, defined as a minimalistic instrumental track that provides the background for a small jam session, consisting of a drum beat and at least two other instruments, can be played to support the players. This type of support is not often used in group jam sessions but is often used by musicians to practice their soloing skills [27]. Musicians will often utilize resources such as YouTube®, Band-in-a-Box® software, iReal Pro® and other forms of generated or recorded jam tracks for this type of practice, regardless of genre. It has also been demonstrated that children as young as five years of age can experience positive effects during music improvisation sessions when using harmonically relevant backing tracks [28, 29].

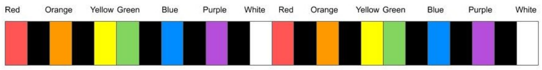


Figure 8. Piano bar color arrangement

Figure 9. Vinyl piano bar placed behind the keys of a digital piano keyboard instrument.

3.5 Piano Bars

Vinyl piano bars, colored in the same colors as the LED cubes, are used to demonstrate the relationship between the notation and the instrument. Piano bars easily fit behind the keys of most piano keyboard instruments making color associations nearly instantaneous. The colors can be moved from one key to another to demonstrate key changes and are also accurately spaced to retain the relative spacing differences between keys (see Figure 8 and 9).

4. HARDWARE PROTOTYPE

The Jam Tabs system consists of LED cubes that provide the user with an interactive light display. The root note of the chords present in the jam track is displayed as a color that can be found on the piano keyboard.

This general design structure lends itself to a myriad of design options. Any system that can omnidirectionally display seven colors and a vertical arrangement of squares and dots can accomplish the main design parameters of Jam Tabs.

Included in these possible designs are projections, physical colored non-backlit cubes, light projection elements such as flashlights and laser pointers, LED strips, LED cubes, holographic displays, and other similar technology.

4.1 Supported Technology

4.1.1 LED Cubes

Jam Tab lights reflect the color of the note present in the music at a given time. Jam Tabs allow for keeping track of what beat in the chord progression is currently occurring. LED Cubes reflect bars of music and accurately depict chord changes within key by color. LED Cubes can be seen from any angle (see Figure 10).

4.1.2 Colored Keyboard

Colored bars were used for color application. These match colors present on LED cubes (see Figure 8 and 9).

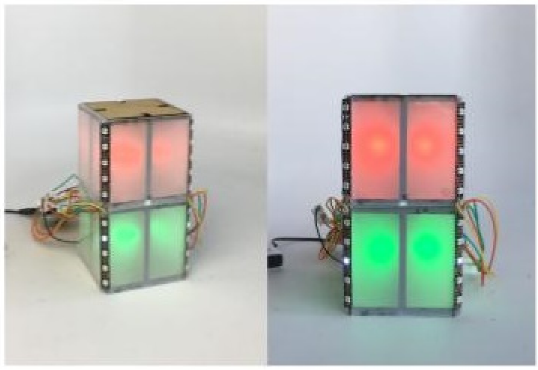


Figure 10. Physical representation of the LED cube hardware prototype.

4.1.3 Backing Tracks

Music in the form of "backing tracks" are provided to musicians at a recommended tempo of 75 beats per minute.

5. LIMITATIONS

5.1 Western Music

Jam Tabs can be used in improvisational settings for any genre of music. Currently however, the Jam Tabs system is designed to be played within tonal western music scales in order to improvise effectively. This is because the colors chosen are optimized for a twelve-note system.

5.2 Subdivision of the Bar

The current Jam Tabs system does not accommodate mid-bar chord changes. This is because each cube is treated as a single bar of music. It is possible to change the time signature to the number of beats present in the smallest subdivision of the bar and then apply the appropriate number of LED cubes to fill the rest of the notation, but this method could easily result in many LED cubes displaying what could be non-essential components of the progression. It is therefore preferred to assign subdivisions as full bars on a case by case basis.

6. DISCUSSION

Jam sessions using the LED cubes are possible in small or large group settings. The majority of users thus far have utilized the cubes in a private setting, in which they were practicing their improvisational skills. The usefulness and design implications of a commonly visible notation require further investigation, however; it is hypothesized that common visibility, like that of Jam Tabs, will enhance the interactive component of a jam session and will create a better sense of community than personal music notation.

Our initial observations lead us to believe that the system successfully accomplished the design parameters it set out to fulfill. It was observed that novice piano players can have fun, creative, and surprisingly successful improvisation sessions using the Jam Tabs system. The system has been utilized in groups of up to 25 people, a session which had its own challenges and successes. It was nevertheless interesting to observe which design parameters work best for which group size. The effect of group size on the successful use of the Jam Tabs system is a topic for future investigation and may even shed light on the social dynamics present in group and collaborative musical play.

Overall accuracy of the notes being played by the participants when following a chord progression seems to be an accomplishable task for most, if not all, participants. Few errors with regards to pitch and duration occur throughout the improvisational activities indicating that players were not getting lost or confused.

Multiple players have responded in regards to the usefulness of the lights as a reference for the order of colors, such as in a progression. It is possible, however, that some participants have more auditory recognition skills than others for musical elements. This may suggest that there is a slightly different set of design parameters that may be uniquely applicable to players who are beginning to play a new instrument but have significant experience in music generally, or with another instrument. These players may be able to track the progression and time signature better than complete novices, but still have very little knowledge and capability of finding the correct notes on the piano keyboard instrument provided. Further investigation is needed to address this type of musician and a separate system may be designed if necessary.

Several usability problems arose in creating and observing the use of the prototype. The first is an issue with displaying subdivision of a bar of music. A single color currently represents a chord and a bar of music. If one wanted to represent two chords in a bar of music they would have to reduce the time signature or increase the number of cubes in the system. This issue is likely best solved on a case by case basis. The second issue that arose was the number of LED's present on the neopixel strip representing time signature. To solve this issue it may be helpful to use only common time signatures. It is also possible to use irregular spacing to solve this problem, although it may create other usability issues.

These two issues cannot be addressed without redesigning certain components of the system. Further iterations of Jam Tabs will address these concerns and relate them to current structures in music. It is likely to be the case that not all music can be completely represented on the Jam Tabs system, but it may be generalized or used in such a way that novices can still be successful with most music to warrant the current design.

7. CONCLUSIONS

Overall, the Jam Tabs system proved to be very useful for providing the essential information of a jam session to the novice musician. The practice of following the root note of a chord progression seemed to be improved and it was particularly helpful for identifying and playing sharps and flats. It was also observed that near all novice musicians were able to creatively and satisfactorily express themselves while improvising regardless of key. Most improvisers played as though playing with four sharps, such as in the key of E major, was just as simple as playing in the key of C with all white notes.

In order to explore these concepts further, future studies aimed at evaluating the impact of the Jam Tabs system on improvisation learning and interaction will be conducted. In addition, future user studies that investigate the effectiveness of the several visual components as well as what degree of complexity can be achieved in the system will also be conducted. This would include a study investigating most effective choice of colors as well as a study addressing the effectiveness of the system on learning improvisation. Though improvements will be made in future iterations of this technology, the Jam Tabs system is apparently quite useful for novice musicians in group improvisational activities.

Acknowledgments

The first author would like to thank Iowa State University, Dr. James Oliver, Dr. Elcin Gunay, and Dr. Jonathan Kelley for facilitating, overseeing and reviewing the initial steps of the JamTabs project.

8. REFERENCES

[1] L. Bouchard and J. Browne, "LIVE STRUCTURES: OCULAR SCORES™," p. 7.

[2] M. L. West, "The Babylonian Musical Notation and the Hurrian Melodic Texts," *Music & Letters*, vol. 75, no. 2, pp. 161–179, 1994. [Online]. Available: www.jstor.org/stable/737674

[3] L. Rowell, *Music and Musical Thought in Early India*. University of Chicago Press, Dec. 2015, google-Books-ID: h5_UCgAAQBAJ.

[4] A. Berkowitz, *_USThe Improvising Mind: Cognition and Creativity in the Musical Moment*. Oxford University Press, Jun. 2010. [Online]. Available: https://www.oxfordscholarship.com/view/10.1093/acprof:oso/9780199590957.001.0001/acprof-9780199590957

[5] R. Ashley, "Musical improvisation," *The Oxford handbook of music psychology*, pp. 413–420, 2009.

[6] V. Iyer, "Embodied mind, situated cognition, and expressive microtiming in African-American music," *Music Perception: An Interdisciplinary Journal*, vol. 19, no. 3, pp. 387–414, 2002.

[7] ——, "Improvisation, temporality and embodied experience," *Journal of Consciousness Studies*, vol. 11, no. 3-4, pp. 159–173, 2004.

[8] M. Fein, *Teaching Music Improvisation with Technology*. Oxford University Press, 2017.

[9] B. Tomassetti, "Beginning blues improvisation pedagogy for the non-jazz specialist music educator," *Music Educators Journal*, vol. 89, no. 3, pp. 17–21, 2003.

[10] J. Kratus, “Growing with improvisation,” *Music Educators Journal*, vol. 78, no. 4, pp. 36–40, 1991.

[11] ——, “A developmental approach to teaching music improvisation,” *International Journal of Music Education*, no. 1, pp. 27–38, 1995.

[12] M. Biasutti, “Pedagogical applications of cognitive research on musical improvisation,” *Frontiers in psychology*, vol. 6, p. 614, 2015.

[13] E. Huovinen, A. Tenkanen, and V.-P. Kuusinen, “Dramaturgical and music-theoretical approaches to improvisation pedagogy,” *International Journal of Music Education*, vol. 29, no. 1, pp. 82–100, 2011.

[14] P. Burnard, “How children ascribe meaning to improvisation and composition: rethinking pedagogy in music education,” *Music education research*, vol. 2, no. 1, pp. 7–23, 2000.

[15] M. Poast, “Color Music: Visual Color Notation for Musical Expression,” *Leonardo*, vol. 33, no. 3, pp. 215–221, Jun. 2000. [Online]. Available: https://www.mitpressjournals.org/doi/10.1162/002409400552531

[16] G. L. Rogers, “Effect of Color-Coded Notation on Music Achievement of Elementary Instrumental Students,” *Journal of Research in Music Education*, vol. 39, no. 1, pp. 64–73, Apr. 1991. [Online]. Available: https://journals.sagepub.com/doi/abs/10.2307/3344609

[17] C. Hooker and M. Csikszentmihalyi, “Flow, Creativity, and Shared Leadership: Rethinking the Motivation and Structuring of Knowledge Work,” in *Shared Leadership: Reframing the Hows and Whys of Leadership*. Thousand Oaks: SAGE Publications, Inc., 2003, pp. 217–234. [Online]. Available: http://sk.sagepub.com/books/shared-leadership/n10.xml

[18] M. Benedek, B. Borovnjak, A. C. Neubauer, and S. Kruse-Weber, “Creativity and personality in classical, jazz and folk musicians,” *Personality and Individual Differences*, vol. 63, pp. 117–121, Jun. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0191886914000919

[19] S. W. Lee, J. Freeman, and A. Colella, “Real-Time Music Notation, Collaborative Improvisation, and Laptop Ensembles.” in *NIME*, vol. 12, 2012, pp. 21–23.

[20] S. W. Lee and J. Freeman, “Real-time music notation in mixed laptop–acoustic ensembles,” *Computer Music Journal*, vol. 37, no. 4, pp. 24–36, 2013.

[21] G. Griffin and R. Jacob, “Priming creativity through improvisation on an adaptive musical instrument,” in *Proceedings of the 9th ACM Conference on Creativity & Cognition*, 2013, pp. 146–155.

[22] J. Bowers and S. O. Hellström, “Simple interfaces to complex sound in improvised music,” in *CHI'00 extended abstracts on Human factors in computing systems*, 2000, pp. 125–126.

[23] S. Jordà, “Sonigraphical instruments: From fmol to the reactable*.” in *NIME*, vol. 3. Citeseer, 2003, pp. 70–76.

[24] B. Bengler and N. Bryan-Kinns, “Designing collaborative musical experiences for broad audiences,” in *Proceedings of the 9th ACM Conference on Creativity & Cognition*, 2013, pp. 234–242.

[25] I. Poupyrev, R. Berry, J. Kurumisawa, K. Nakao, M. Billinghurst, C. Airola, H. Kato, T. Yonezawa, and L. Baldwin, “Augmented groove: Collaborative jamming in augmented reality,” in *ACM SIGGRAPH 2000 Conference Abstracts and Applications*, vol. 17, no. 7, 2000, p. 77.

[26] Y. Wu and N. Bryan-Kinns, “Supporting non-musicians? creative engagement with musical interfaces,” in *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, 2017, pp. 275–286.

[27] F. Seddon and M. Biasutti, “Non-music specialist trainee primary school teachers' confidence in teaching music in the classroom,” *Music Education Research*, vol. 10, no. 3, pp. 403–421, Sep. 2008. [Online]. Available: https://doi.org/10.1080/14613800802280159

[28] D. M. Guilbault, “The effect of harmonic accompaniment on the tonal achievement and tonal improvisations of children in kindergarten and first grade,” *Journal of Research in Music Education*, vol. 52, no. 1, pp. 64–76, 2004.

[29] ——, “The effects of harmonic accompaniment on the tonal improvisations of students in first through sixth grade,” *Journal of Research in Music Education*, vol. 57, no. 2, pp. 81–91, 2009.

STRATEGIES AND SUGGESTIONS FOR SINGING IN FOREIGN LANGUAGES BASED ON PHONETIC MUSICAL NOTATION

Korbinian Slavik
LMU München
slavikk@phonetik.uni-muenchen.de

Markus Jochim
LMU München
markus.jochim@lmu.de

Verena Klasen
TU München
verena.klasen@tum.de

ABSTRACT

There are different approaches to a good pronunciation when singing in foreign languages. One promising example is the use of the International Phonetic Alphabet (IPA). Transcribing lyrics with this alphabet has a tradition of sev-eral decades, but it can be very time-consuming when done manually, especially if you want to store IPA information directly within a score. We investigated, for what musi-cians the use of IPA is useful and what they normally do when singing in foreign languages. In a questionnaire with more than 450 participants from 19 different countries we asked singers and conductors about their strategies when singing in a foreign language and whether they thought it was useful to have IPA inside a score. We identified a vari-ety of strategies which are used for singing in foreign lan-guages like listening to recordings or to experts of the tar-get language. Additionally, 60-70 % of all participants and 90 % of opera singers think that a phonetic alphabet could be helpful in a score. Test subjects were also asked to name the languages they wanted as transcriptions in the notes, where Russian was second to none. As a consequence of these results, we are working on an automated approach for writing IPA information directly in MusicXML data, thus combining IPA transcription with the original score.

1. INTRODUCTION

Using the International Phonetic Alphabet (IPA) for transcribing lyrics has a tradition of several decades, but it can be very time-consuming when done manually – especially if you want to store IPA information directly inside a score. To investigate the acceptance of IPA notation in scores among the music community, we ran a questionnaire study with 457 participants from 19 different countries, aged between 17 and 83. We asked singers and conductors about their strategies when singing in a foreign language and whether they thought it useful to have IPA inside a score. As a solution for this linguistic problem of singing, we are working on an automated approach for writing IPA information directly in MusicXML data, thus combining IPA and the original score. Firstly, we will describe the history of using IPA for singing and the linguistic issues that arise when singing in different languages. Then, we introduce the structure of our online survey, and discuss our participants' answers. Finally, we will conclude with a short summary and an outlook on our technical approach which we developed for storing phonetic information into a score.

2. LINGUISTIC ISSUES AND A HISTORY OF USING IPA FOR SINGING

In linguistics, the fields of phonetics and phonology are concerned with the pronunciation of sounds in all languages and their physiological constraints. How and for which reasons speech sounds are formed are the central questions in this area. Furthermore, they investigate and describe sets of sounds and rules about them in individual languages. One of their most important tools is the International Phonetic Alphabet (IPA) [1]. For singing, we must consider some additional rules (that would not be appropriate for speaking) to account for the adjustment of phonetics and phonology, because spoken and sung sounds are often different:

> One of the most widespread errors is that spoken and sung sounds are the same. Nothing is further from the truth. [...] The adjustment of phonetics to the vocal phrase is the real problem for any accompanist and coach and the solution constitutes a very important part of this art. [2, p. 5]

The Austrian-American conductor Kurt Adler, who wrote these remarks, was among the first people to take an interest in IPA regarding musical lyrics. He wrote about using IPA for singing in different languages already in the 1960's. A further pioneer was Berton Coffin in the 1970's [3]. He explained physiological characteristics of singing for breathing, vowels and consonants. He wrote further books about singing in Italian, French, and German [4]. If we read such literature, we must be aware that there is a continuous development of the phonetic languages and that there are often different schools for singing in a particular language. Therefore, it is often difficult to identify a common rule set for singing. Since the 1990's, Nico Castel has translated and transcribed Italian, German, and French opera libretti, which have set the standard for vocal literature today. Further literature for singing with IPA is available for Latin, old French, old English and many other languages [5, 6, 7, 8].

In these 50 years of efforts of using IPA for transcribing musical lyrics, the lyrics were always separated from the musical score. In our view, the logical consequence of this development is to use IPA directly in musical sheets, which is not a standard procedure in classical music yet. Nevertheless, there are various problems with storing IPA information in a score. The transcription itself is difficult because there are different orthographic texts and simple one-to-one descriptions of the pronunciation are not possible for these texts. Furthermore, one must consider many languages with complex linguistic rules, like the syllable-final obstruent devoicing in German. Additionally, rules that apply specifically to singing must be accounted for. Diphthongs in German, for example, are to be transcribed differently for singing than for speaking and the distribution of syllables might differ from that for speaking.

3. QUESTIONNAIRE

The online questionnaire in German and English described in this chapter was created in November 2018 [9]. In a first step, the three-part structure of the questionnaire is explained. Subsequently, the questionnaire is systematically evaluated on the basis of this structure.

Firstly, socio-demographic data of the participants will be analysed. Then, strategies for singing in foreign languages and the evaluation of phonetic transcription in the scores will be discussed.

3.1 Design

At the end of the questionnaire, we asked the participants about their proficiency concerning singing in general. Here, we were also interested in socio-demographic data, like age, gender, country of origin, or profession of our participants. In the second part, we asked about strategies for singing in foreign languages. The most common approaches to such pieces of music are determined by means of Multiple Choice. Another Multiple Choice and a Single Choice question clarified what kind of annotations the participants use when singing in foreign languages and whether they are familiar with the principle of phonetic transcription. There was always the possibility of free text input. The third part was about evaluating the usefulness and value of using IPA in scores. In addition, it was determined for which languages, pieces and genres a transcription in the scores is desired. These questions were again based on multiple choice, a rating scale and free text information.

3.2 Sociodemographic data and self-assessment

457 participants from 19 different countries, 280 female, 171 male and six persons with undefined or diverse sex took part in this survey over a period of almost three months. 396 test persons came from Germany, 12 from Austria, ten from Switzerland, nine from Sweden, three from the USA, three from Slovakia, and further participants from Belgium, the Netherlands, Italy, the UK, Uruguay, Spain, Norway, Lithuania, Israel, Denmark, the Czech Republic, China, Bulgaria or from other countries not marked.

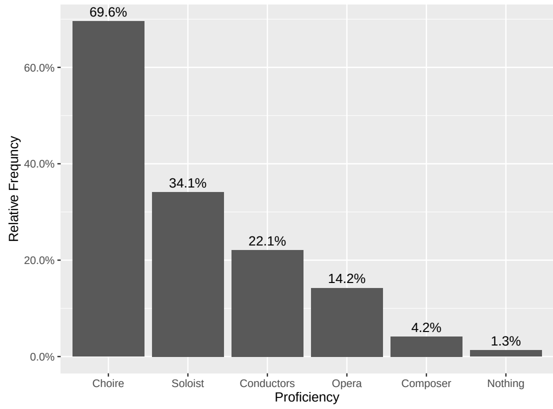


Figure 1. Participating choir singers, soloists, conductors, opera singers, composers, and participants who did not identify with these options - Question 9. This being a multiple choice question, the percentages need not add up to 100 %.

The relevance of the questionnaire is demonstrated not only by the participants from numerous countries, but also by the age of the test subjects, which varies from 17 to 83. There are many participants between 20 and 30, but there is also active participation in the survey between 30 and 60. There are even some participants older than 70 and 80. One can thus speak of a complex topic that attracts people of all ages.

The self-assessed singing ability of the singers is well suited for further examination for several reasons. Professionals are represented with 35 %, advanced singers with slightly more than 32 % and amateurs with almost 18 %. As we will show in the following two subchapters, the needs and strategies of these groups in dealing with foreign-language music are different under certain conditions. Multiple entries for the profession are rare at around 5 % and are therefore not taken into account in the analysis.

We also expect different preferences in the musical specialization of the participants. So, it is obvious to assume that soloists, who are represented with 34.1 %, have other needs than choir singers, who, as can be seen in Figure 1, are very strongly represented with 69.6 %. Conductors and opera singers are also clearly represented with 22.1 % and 14.2 %, respectively. We interpret more than 100 conductors within a questionnaire as an indicator that this problem of singing in foreign languages is particularly important for leaders of singing groups.

Some of the participants identified with more than one of the groups, e. g. they reported to be both a soloist and a choir singer. We treated these simply as a part of both of these groups and not of a special group "soloist-choir-singer". We assume that in a situation where a participant acts as a choir singer, it does not play an important role whether or not he also acts as a soloist at other times.

3.3 Strategies for Singing in Foreign Languages

When singers are confronted with pieces in foreign languages, they use a variety of possibilities to cope with this linguistic challenge. As shown in Figure 2, the strategy of

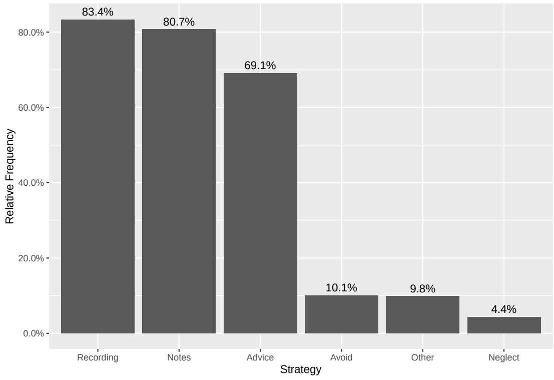


Figure 2. Relative frequency of the strategies used for singing in foreign languages. These strategies are Listening to recordings, taking notes in the sheets, or seeking advice from experts. The strategies to avoid singing in a foreign language, to neglect other strategies or the correct pronunciation are used much less frequently - Question 1.

listening to recordings is particularly popular (83.4 %) and not a significant problem in the digital age. Also, more than 80 % of all singing individuals take notes. According to the statements of many participants, writing serves as a memory aid and is therefore a central component of their work with music. In the case of borrowed musical sheets, however, it can be problematic or even forbidden to take notes. It is also not useful or necessary to take notes, when the test subjects are in an early singing stage, have an excellent auditory memory or elaborate language skills for the target languages in question.

Advice from a person who is familiar with the language to be sung is sought in about 69 % of cases. About 31 % of the test subjects do not seek additional opinions. On the one hand, this may be due to the lack of opportunities, especially in the field of laypersons, of whom only about 50 % seek advice. Or, on the other hand, to the unwillingness to deal with the target language more closely. Such a reluctance can point to a conscious neglect of the correct pronunciation. However, only 4.4 % of the test subjects state that correct pronunciation is sometimes not so important to them. This low relative frequency shows the intention of almost all musicians to achieve a pronunciation as "correct" as possible when singing. Moreover, avoiding to sing in a language one does not master is rare, with a prevalence of only about 10 % among the participants. This means that most singers are not afraid to be confronted with music in foreign languages.

Almost 10 % of the test subjects also use other strategies for singing in foreign languages. The additional remarks include the involvement of speakers of the mother tongue, the use of specialist literature and websites, translations and often handwritten transcription or transliteration.

The next question investigated, which strategies the test subjects use when taking notes in the sheets. This question specifically aims to show whether singers approach this issue systematically or not. As can be seen in Figure 3, only 22.5 % of the respondents stated that they follow a strategy that they had previously defined when taking notes in the sheets; 11.4 % of all laypersons, 24.2 % of all advanced singers and 30.6 % of all professionals follow a strategy. A logistic regression gives significant results for the positive responses of the advanced ($z = 2.061$; $p < 0.05*$) and for the positive responses of the professionals ($z = 2.710$; $p < 0.01**$) compared to the general sample of positive responses to this response option. For the laypersons, no significant result is achieved. This is probably due to the already small group size of laypersons and additionally to the few positive answers, with only eight occurrences in total.

Also, whether the singers take notes above or below the text to be sung seems to be unsystematic with about 50 % each. There is also no decisive difference between laypersons, advanced singers and professionals. Normally, the original text is retained and only erased by 5 % of all test subjects. It is astonishing that only about 20 % sometimes do not know how to write down unknown sounds. This value indicates a high level of linguistic knowledge and a conscious examination of the musicians' pronunciation strategies through singing. Since a large number of academics are also surveyed in this questionnaire, this value was certainly influenced by phonetic instruction at colleges and universities and by the intensive study of languages and speaking. However, it is likely that for certain sounds and words in some languages, awareness is only rudimentary, e. g. when one thinks of the complex realisation of tonemes in Norwegian or Mandarin in singing [10, pp. 294-295]. When interpreting this answer, it must also be taken into account that 20 % of the test subjects do not make any notes at all in the sheets, but are nevertheless taken into account in the false answers and thus also have an effect on the value of the true answers given in the graph. Thus, the error rate in this question is increased. However, this does not diminish the general tendency towards a lack of systematic notes in the notes. This impression is reinforced by further remarks of the respondents, who, for example, say that they write the improved pronunciation on the staff, that they adapt themselves specifically to the place on a sheet of music. They further stated that they use partly phonetic transcription (IPA), partly invented characters, or that they stick over the original, e. g. Japanese, text with having listened to a recording.

Approximately half of all test persons state that they have not yet been confronted with phonetic transcription. 100 people, or 22 % of participants, can remember a discussion about phonetic transcription, but not exactly in which context. 128 people, or 28 % of participants, can remember and often cite the encounter with foreign languages, the confrontation at school and university or pronunciation dictionaries as a source. The high number of those who apparently have not had any encounter with phonetic transcription or IPA can be interpreted in two ways. On the one hand, it is possible that some test persons have interpreted this question in the context of this questionnaire in such a way that they have never worked with phonetic transcription while singing or that they did not know how to use the term phonetic transcription or IPA, but know the theoretical concept. On the other hand, it is possible that people

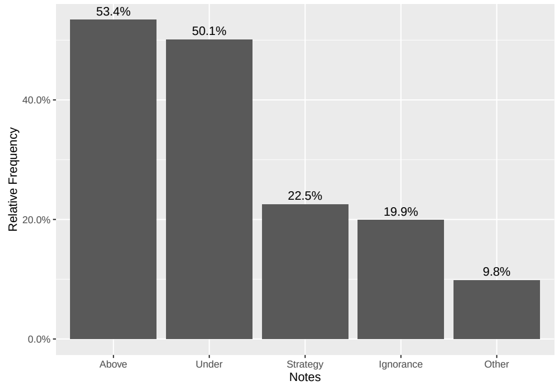


Figure 3. Answers to the question how the subjects proceed when taking notes. The answers are distributed more or less equally, whether they take notes above or below the notes. Followed by the frequency of those who make up a strategy before taking notes in the sheets and by the statements of those who sometimes do not know how to transmit a sound, other remarks and the statements of those who sometimes cross out the original text - Question 2.

with a lower educational level or older people could not actually gain any experience with this concept, as it is explicitly noted by some test persons.

3.4 Phonetic transcription in the sheets

In the third section of the questionnaire, the subjects were confronted with a concrete example of IPA transcription within musical scores (see Figure 4). The purpose of this figure was to assess whether they considered such a representation of a language useful. Exactly 60 % of the participants find a transcription in the sheets meaningful. Considering that, as described above, 50 % of the participants state that they have not worked with phonetic transcription yet, this value is high. However, a concrete example is given for this question, which might facilitate an answer. This assumption is confirmed by the fact that those who indicated in the previous question that they had not been confronted with phonetic transcription before are evenly distributed between the yes and no answers to the meaningfulness of IPA in the scores.

Slightly more than 31 % of the participants do not find such a transcription in the sheets useful, and only 3 % do not care. The rest are undecided or give their opinion on this question in the notes. Fears are, for example, that IPA is too difficult to interpret for the untrained or that the text becomes too confusing. In addition, the above-mentioned problem between phonological theory and phonetic reality in singing is addressed and the therefore the approach with IPA is criticized. Some musicians also note that a discussion of the target language is necessary in order not to neglect semantic and other linguistic problems; moreover, it would tempt laziness if one no longer had to deal with the target languages. Positive feedback is also provided, such as the emphasis on IPA's easy readability, or its use in languages that use a foreign alphabet and are thus difficult for respondents to read, e. g. Russian for singers whose

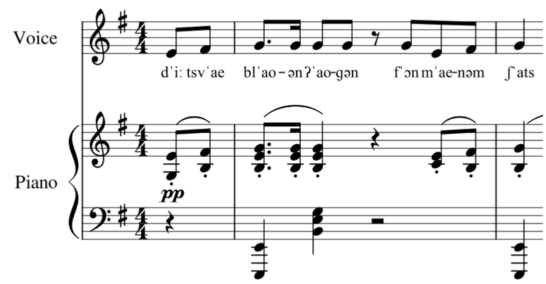


Figure 4. Example of a phonetic transcription in a musical sheet. This was presented in a similar fashion to the participants as a part of Question 4. The piece "Die zwei blauen Augen" by Gustav Mahler can be found in the song cycle "Lieder eines fahrenden Gesellen" (around 1884–85).

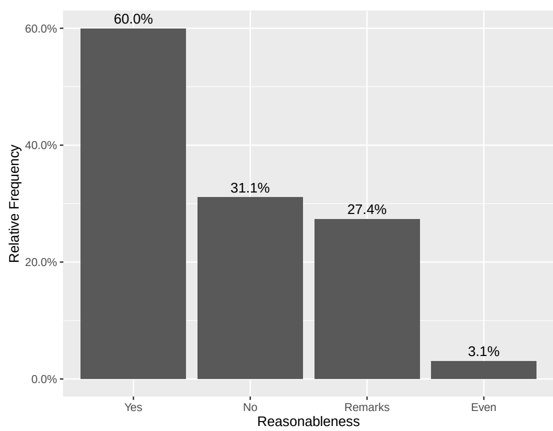


Figure 5. Answers to the question whether the experimental subjects think whether transcriptions in sheet music are useful or not - Question 4.

native language uses the Latin alphabet.

In the next question, the test subjects were asked to name the languages they wanted to see as transcriptions in the notes (see Figure 6). Russian is clearly in the first place with an absolute frequency of 113 wishes, which corresponds to over 35 % of all wishes. We suppose that in many cases, this might also mean Old Church Slavonic language variants or dialects, which often appear in choral literature and which the respondents may have identified as Russian; this is due to the Cyrillic alphabet, which is used for Old Church Slavonic and present-day Russian alike. A priming bias from the first question of the questionnaire is also likely, as it explicitly mentions a foreign language that uses a Cyrillic script. This example was chosen to make the relevance of singing in a difficult-to-read language clear to the singers from the outset. Note that this priming only potentially explains the high result for Russian but not for other often-named languages. Two other languages were named substantially more often than the rest and those are French (53 times) and Czech (37 times). There were also responses that specified language groups or rough geographical indications. "Chinese" was mentioned nine times and probably refers primarily to Man-

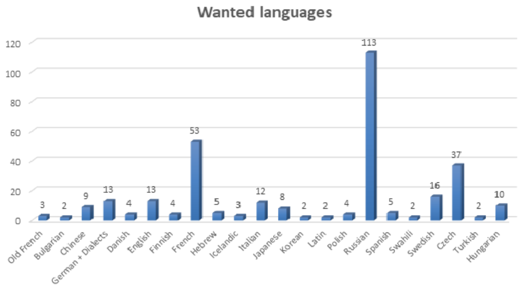


Figure 6. Languages the subjects stated they wanted as transcriptions in the scores. Question 6.

darin, the most widely spoken language in China, which is also the world's most common mother tongue with around 918 million speakers [11]. With eight mentions, Slavic languages lead the field of explicitly named groups. Note that responses clearly referring to a group of languages and responses that were only given once were not included in Figure 6. Of course, these answers mirrored the geographic and linguistic background of our participants. They are therefore not universally generalizable, but still represent an interesting sample from within the questionnaire's target audience.

4. CONCLUSION AND OUTLOOK

The aim of this paper was to show that the phonetic and phonological problem of singing affects many singers and associated persons. Furthermore, it was shown that this is a complex linguistic problem with many facets and possibilities for a solution.

As a promising solution, we have described the development of phonetic transcription for singing in the last 50 years and suggested as a logical consequence to use IPA directly in the musical sheets. In a questionnaire, we asked singers about their methods when dealing with the problem of singing in a language they do not know. These strategies, such as taking notes or consulting an expert of the target language, were evaluated and conclusions drawn for possible help. The test subjects were also specifically confronted with the suggestion of a transcription directly in the scores. The question as to whether they think such an approach makes sense was answered in the affirmative by 60 % and only a little over 30 % are against such a solution. It also turned out that the test subjects had clear preferences regarding the languages to be transcribed. Russian, French and Czech are particularly desired for transcription in the notes. Genres that should be used for a transcription in the sheets are choral music and operas, but also classical solo literature.

Now, we are working on an automated approach in the form of a computer program that facilitates transcription in the musical sheets. The goal of our circular transcription process is to become as automatic as possible. With the help of a decision tree and numerous improvement mechanisms, such as regular expressions, this program is specially programmed for sung text. The potential for improvement of the transcription process under programming aspects and possible further offers, such as voice recordings, were further results of this questionnaire.

A frequent criticism of our proposed solution is that the central theme of the meaning of the text and the learning of the language is not taken into account. In order to counter this criticism, additional work should be done with narrow and wide translations and assistance provided for learning the target language, such as alphabets and audio examples. In this way, it is ensured that the emotional content of the target texts is not lost, but transported across borders.

Acknowledgments

We would like to acknowledge some helpful comments from a conversation with Bard Suverkrop.

5. REFERENCES

[1] International Phonetic Association, Ed., *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge, U.K. and New York, NY: Cambridge University Press, 2014.

[2] K. Adler, *Italian, France, Spanish, German Phonetics and Diction in Singing*. Minneapolis: University of Minnesota Press, 1967.

[3] B. Coffin, *The Sounds of Singing: Vocal Techniques with Vowel-Pitch Charts*, 1st ed. Boulder, Colo.: Pruett Pub. Co., 1976.

[4] B. Coffin, R. Errolle, W. Singer, and P. Delattre, *Phonetic Readings of Songs and Arias*, 2nd ed. Lanham, Maryland: Scarecrow Press, 1982.

[5] T. J. McGee, *Singing Early Music: The Pronunciation of European Languages in the Late Middle Ages and Renaissance*. Bloomington: Indiana Univ. Press, 2004.

[6] L. F. Stelten, *Dictionary of Ecclesiastical Latin: With an Appendix of Latin Expressions Defined and Clarified*, 10th ed. Peabody, Mass.: Hendrickson Publishers, 2013.

[7] K. LaBouff, *Singing and Communicating in English: A Singer's Guide to English Diction*. New York: Oxford University Press, 2008.

[8] D. R. Karna, *The Use of the International Phonetic Alphabet in the Choral Rehearsal*. Lanham, Md.: Scarecrow Press, 2010.

[9] D. Leiner, "SoSci Survey," 2018.

[10] W. Ophaug, "The challenge of identifying vowel phonemes in singing," in *Text and Tune*, T. Proto, P. Canettieri, and G. Valenti, Eds. Peter Lang AG, 2016.

[11] D. Eberhard, G. Simons, and C. Fennig, *Ethnologue: Languages of the World: Online Version*, 21st ed., Dallas, Texas, 2019.

USING GESTURE DATA TO GENERATE REAL-TIME GRAPHIC NOTATION: A CASE STUDY

Gil Dori
Phonos Foundation
Barcelona, Spain
gil.dori@asu.edu

ABSTRACT

The paper describes the concepts and compositional process of my recent real-time graphic score work *Arcos*, for cello and augmented violin bow. The work's graphic notation is generated directly from gesture data of various bowing techniques. A Myo armband was used to record cello bowing data, and the augmented bow's own position tracking module was used to record its motion data. After processing the data, gestures were visualized on the screen, as a form of real-time graphic notation based on imitation. This notational approach—a low-level symbolic representation of gestures—allows for an immediate, intuitive interpretation on the spot, and provides an instantaneous connection between notation and action. My work also offers a distinct perspective on notation for electronic instruments in the context of real-time action-based scores.

1. INTRODUCTION

1.1. Real-Time Action-Based Notation

One approach to digital scores with notation that is "created or transformed during an actual musical performance" [1, p.1], i.e. real-time scores, focuses on representation of performance actions. This so-called action-based notation, albeit not restricted to real-time scores, helps to facilitate sight-reading and to provide the performer with easily available information for interpretation on the spot. These are crucial features for the practice of real-time scores. Gerhard Winkler, a real-time notation pioneer, writes:

> Different parts of the score have to be reduced to a number of elements, which can be learned and 'trained' in advance, and which can be seized with 'one glance' immediately during a performance. On the other hand the used signs have to be precisely [sic.] enough to avoid that the musicians shift into 'improvisation' [2, p. 2].

Furthermore, he emphasizes the importance of creating an environment in which performers and computers can communicate and influence one another in a "complex, non-linear way", [2, p. 1].

Winkler maintains that combining graphic notation and standard, common-practice notation is the most efficient way to create real-time scores. For example, his work *KOMA* (1995-96), for string quartet and interactive live electronics, features a score that changes in real-time according to decisions the leader of the quartet makes at the moment of performing. The score combines different types of notation, such as common-practice notation to indicate pitch; graphic notation to indicate glissandi; and text to indicate microtonal changes in cents [2, pp. 2-3].

Seth Shafer continues Winkler's approach with *Terraformation* (2016-17), for viola and computer. Its real-time score combines a five-line staff for common-practice notation; tablature for fingering positions; and color gradients for bow position and left-hand finger pressure [3, p. 4]. By applying physiological models of performance actions, Shafer designed a score that facilitates the "cognitive translation from graphic representation to bodily action" [3, p. 3]. The algorithm that drives the score takes into account these physical action models when selecting what musical material to present at a given moment. This action-based procedure guarantees to produce idiomatic material that is easy to sight-read, yet still allows for wide sonic variety and depth [3, p. 6]. However, while Shafer intends to balance the cognitive strain on the performer, the score interpretation process appears to be time-consuming and unintuitive. In his paper about *Terraformation,* Shafer relays violist Mike Capone's experience interpreting the piece. This process involves several stages, from gathering information from the different parts of the score to executing the performance actions [4].

On the other hand, graphic notation by itself could provide sufficient information to the performer, especially when used to indicate simple actions. Ryan Ross Smith, for example, developed an idiosyncratic graphic action-notation language which employs radial motion to specify points of attack (and sometimes sustain) in time. Albeit its apparent simplicity, this notational approach allows for complex rhythmical procedures, from hocket to polytemporality. Smith also applies ideas of imitation for action notation. For example, the scores of his works *Study no. 15b* (2013), for cello or upright bass, and *Study no. 15b.1* (2013), for violin or viola, show the fingerboard of the instrument, with circles that indicate where to press the strings [5, 6].

Particularly interesting to this paper is Smith's *Study no. 50* (2015), for a percussion ensemble, in which each performer plays on seven wooden planks. The real-time score shows attack cursors that travel along arcs between seven nodes. One node is assigned to each plank, and the attack cursor indicates when to play which plank. The score is projected downwards from above the ensemble, so that the nodes are positioned directly over the planks. Projecting the score in such fashion "creates a direct correspondence between the notation and the instrument" [7, p. 6]. The performers, then, can interpret the score instantaneously by imitating the motion they see. It is, however, a one-way communication between computer and performers. The algorithm that drives *Study no. 50* generates the notation in real-time automatically and randomly, without any external influences nor an overall compositional structure, as Smith strives for an infinite "consistently-inconsistent flow of events" [7, p. 6].

I set forth to design an intuitive graphic notation environment by recreating actions on the screen, directly from gesture data, that the performers can imitate. This approach allows to extend the amount of information the performers receive, while maintaining a simple, easy to sight-read notation. It provides an instantaneous connection between notation and action through a low-level symbolic representation of gestures. In addition, performer's actions influence the score in real-time, and every action may produce different outcomes, consequently forming a non-linear interaction between performer and computer. This interaction is further enhanced by composing for an electronic instrument. Doing so, however, necessitates special attention in designing this real-time action-based score.

1.2. Gesture and Morphology

Action-based scores for acoustic instruments rely on the inherited, physical interrelation between gesture and sound. The composer who notates actions presupposes that the physical properties of a gesture affect the sound directly. An example for this link between gesture and sound would be a guitarist who wants to produce a loud, bright sound, that encompasses a wide frequency range. To do that, the guitarist would forcibly strum all the strings in one motion, close to the bridge, with the fingernails or with a plectrum. On the contrary, to produce a soft, warm, low frequency sound, the guitarist would pluck the lowest string gently with the inner part of thumb, close to the fingerboard. Thus, as Garth Paine notes, the "physical gesture determines the amplitude, pitch, and timbre of each [sound] event" [8, p. 1]. Trevor Wishart, in his seminal book *On Sonic Art,* coins the term sound morphology to describe this link between action and sound objects, the "gestural structure" of sound, and how this interrelationship manifests in time [9].

Live electronic performance poses a unique problem in regards to morphology, as the "traditional links with physical sound-making are frequently ruptured" [10, p. 1]. Kim Cascone even characterizes live laptop performances as sound created in a non-existent place, with a fake sense of authenticity [11]. A solution to this issue lies within gestural sound control and embodied interaction with sound, using what is generally referred to as new interfaces for musical expression (NIME). However, to provide the most convincing sense of authenticity in live electronic performance, such interfaces must take into account vital gestural elements that affect the sound, such as pressure, speed, and position [12].

My composition is written for an acoustic instrument, cello, and a hybrid instrument, which combines a violin bow with a digital sound processing component. For lack of a better term, I refer to this instrument as an augmented violin bow. Composing a real-time action-based score for such an electronic instrument raises challenges not only regarding morphology in electronic music performance, but also regarding expressing electronic instrument's morphology in notation. The gesture data approach to graphic action-based notation presented in this paper offers one possible way to address these challenges.

2. COMPOSING ARCOS

2.1. Background

Arcos was composed during my residency at Phonos Foundation in 2019. The idea for a real-time score that is based on gesture data formed after premiering my piece *(bcn)621* (2018), for cello, percussion, and laptop, which is unified by bowing actions. All of the performers in this piece are using bows, playing similar and related gestures. For the laptop part, it meant creating a fitting interface for performing that will provide the same performance abilities and relationship between action and sound as the cello and the percussion do. Thus, I built an augmented violin bow, guided by notions of sound morphology in electronic music performance. Naturally, in this context of an action-based composition for an ensemble, these notions have an even greater significance. Section 2.3, about instrumentation, elaborates more about the augmented violin bow.

The composition of *(bcn)621* uses isorhythm techniques, applied to actions within blocks of time, in order to shape a perceivable structure through timed sections of actions. The parts consist of a fixed series of gestures, each within a defined time frame. The series repeats itself in the same order but in different durations. Changes in the overall duration of the sequence affect the durations of the individual time frames proportionally. For example, the cello sequence begins with playing tremolo, then resting, then scraping the bow along the strings. The total duration of the first instance of the sequence is 45 seconds, with playing tremolo for 4 seconds, resting for 3 seconds, and scraping for 6 seconds. When the sequence repeats for the third time, its overall duration is 90 seconds. Then, the same gestures take 8 seconds, 6 seconds, and 12 seconds respectively.

Although *(bcn)621* was successful in terms of achieving the compositional and conceptual goals it aimed for, there were practical issues that complicated rehearsing and performing the piece. I believe that the gesture-based real-time score of *Arcos* offers a solution for these issues.

The main problem was maintaining the accurate times of the sections. The temporal units are measured in seconds, indicated by a tempo mark of 60 beats per minute. The interface of the digital instrument shows a timer, cues count, and automatic on-screen instructions. The cello and percussion performers, however, had to constantly count beats in order to make sure they execute actions in the exact times and for the right durations. This need for endless counting was very stressful for the performers, and diverted their attention away from performing the piece as an ensemble in a musically sensible way. To relieve the performers, I made a visual click track, a Max patch, that shows beats and bar numbers[1]. While this patch helps immensely, and allows for a smooth performance, it brings additional technical requirements that could be cumbersome, such as mounting an auxiliary screen. In addition, monitoring the digital instrument is done visually, which presents too much visual information at once, and can be confusing during a live performance. A real-time score, then, would significantly reduce the strain on performers, as well as it would keep technical requirements to the essential minimum. It would alleviate practical difficulties, while still supporting and developing the concepts explored in *(bcn)621*.

2.2. Key Concepts

Same as with *(bcn)621,* the broad concepts of *Arcos* are unity by bowing gestures, and structure through blocks of time. The current work also presents the central concept of creating a more natural, intuitive performance environment through a low-level symbolic representation of gestures. This idea, which came as a response to the pitfalls of *(bcn)621*, is manifested in generating the real-time graphic notation directly from gesture data.

Basing the whole notation on gesture data brings the gestures even more into the forefront of the work, further emphasizing the idea of actions as a means of unifying the composition. As with any real-time score, sight-reading becomes the prominent way of performing. The gesture data driven notation presented here offers a more immediate, intuitive approach to sight-reading.

My basic assumption is that imitation is a rudimentary form of communication, which elicits instantaneous, instinctive response. Since the graphic notation is generated directly from gesture data, the performers can imitate what they see naturally, executing performance actions intuitively and idiomatically. They can perform in a way that is already ingrained in the performance practice of their instrument, without the intervention and interpretation of high-level symbolic notation. By contrast, the score of *(bcn)621* combines common-practice notation, action-based graphic symbols, and text instructions. While this notational approach is effective in explaining on paper which gesture to play and for how long, the process of learning the score—the symbols and their meaning—and interpreting it, is time consuming and requires intense conscious effort.

The real-time aspect of *Arcos* also contributes to this concept of intuitive performance by eliminating the need for counting temporal units, and even for thinking about time all together. It lets the performers focus exclusively on performing.

Generating the graphic notation directly from gesture data brings the work closer to what Juraj Kojs sees as the “pure action-based scores”, which provide immediate, easy access to music, and “utilize images that suggest clear instructions at first sight and [that] need no further explanation” [13, p. 67]. One reservation, however, regarding the configuration of my real-time score, is that the algorithm selects which gesture to display from a closed set of given actions. This type of real-time score is referred to as live permutated, or real-time permutative score [14, 15]. Since only the order is indeterminate, but the material itself is known, the performance does not entail pure sight-reading. In a sense, however, this resonates Winkler's notion of real-time notation that can still be learned in advance.

An isorhythmic structure is applied in *Arcos* as well, with large sections, divided into blocks of time, that repeat in augmentation or diminution. When the sections repeat, the change in overall duration alters the durations of the internal units proportionally. The ratios in this case derive from data analysis of gestures, but simplified. And, as I have mentioned above, the order of the actions is not fixed, unlike the sequences of *(bcn)621.*

Since actions are no longer organized in a set order, units of rest, or inaction, have a greater importance for the perception of the structure. As these units of inaction are the only fixed segments within each section, they become the key element for comprehending the structure, functioning as markers of form.

In addition, the visual information contributes to the sensation of time as an indicator of structure. Seeing how long a gesture appears on the screen, as opposed to only hearing a sound being played, may help with grasping the blocks of time of which the sections consist, and the temporal relationships between them. The combination of visual and auditory information possibly allows for perceiving more complex temporal relationships. The algorithm itself facilitates the use of unconventional ratios that are hard to implement on a paper score. On the other hand, to make these relationships perceivable, they cannot be too complex. Hence, I decided to simplify the ratios I derived from the gesture data.

2.3. Instrumentation

The piece is written for cello and augmented violin bow. Since the work focuses on bowing gestures, the cello part is made exclusively for the right hand, playing with a bow. The sole left hand instruction is to stop all the strings lightly—so they are neither fully pressed nor completely muffled—to produce sound which could be described as “hollow” or “airy”. The cellist, however, could do that by placing a rubber band over the top part of the neck of the cello, thus, not using the left hand at all. The cello may be substituted by a double bass. In any case, the acoustic instrument should occupy the lower range of frequencies, in order to complement the augmented bow, which produces higher frequencies.

The augmented violin bow consists of a tangible part (a violin bow) and a digital part (a Max patch). As mentioned above, the purpose of the augmented bow is to afford a physically engaging electronic performance, fo-

1 A standard click track was not a viable option, due to the need for indicating bar numbers.

cused on bowing gestures. I wanted to express the bowing idea not only in terms of movement, but also in terms of sound. Therefore, the performance is done with the bow, and its own sound is amplified and processed. The bow's motion and position data is mapped to the sound processing units (figure 1).

The construction of the bow is fairly simple: I mounted a contact microphone on the bow frog, and an LED diode on the bow stick. The contact microphone amplifies the bow's vibrations, i.e. the sound of the bow itself, and the LED is used for tracking the position of the bow, based on the Jitter jit.findbound method[2]. The Jitter algorithm follows the LED light as a point in space, and outputs two-dimensional coordinate positions of this point. Since the input is from a webcam, it is crucial to find the right position in the performance space, and to make sure no other light interferes with the LED light input. The amplified sound of the bow is processed in real-time, according to movement detection, brightness level, and position tracking data of the LED light. The audio signal processing includes a flanger effect and a spectral pitch shift effect. The augmented bow, then, is both the sound generator and the gestural sound controller, making it a standalone instrument with clear morphology.

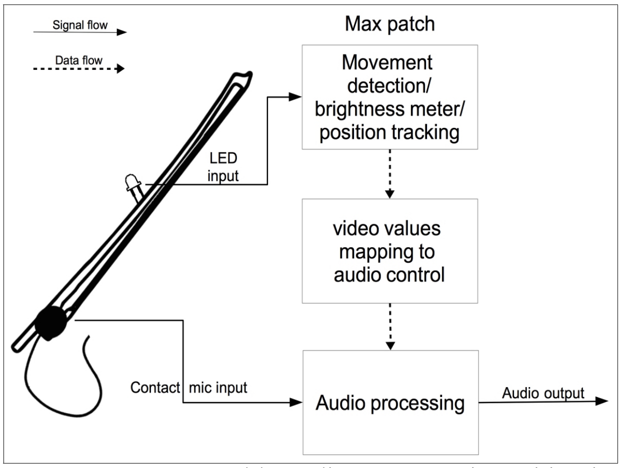


Figure 1. Augmented bow diagram. X axis position is mapped to pitch shift and flanger delay gain, Y axis position is mapped to flanger modulation rate, and distance from the center is mapped to flanger delay time. Movement activates the audio output, and brightness controls the volume.

2.4. Data Acquisition, Processing, and Mapping

Clearly, a vital step in creating a score based on gesture data is to obtain the data and make it usable. Collecting motion data from the two instruments was done by registering Myo armband[3] data of cello bowing gestures, and by capturing videos of the augmented bow position tracking, as it appears on the Max patch Jitter window.

Cellist Leo Morello lent a hand, quite literally, with the cello gesture data acquisition. In a long rehearsal session, Morello performed various bowing actions, while wearing a Myo armband on his right hand. The actions included normal bowing, scraping the bow along the strings, circular bowing, ricochet, arpeggiation, applying different amounts of bow pressure, and bowing on different parts of the instrument. The Myo output of each gesture was saved onto a separate text file. Each file, then, contains accelerometer, gyroscope, quaternion, and EMG data of specific gestures[4]. It is important to note that the Myo device was used only for recording data, and it is not meant to be used during the actual live performance.

In order to make the cello data available for visualization—recreating the gestures on the screen with a virtual bow—the next step was to calculate the tilt angle, angular velocity, and angle of rotation around the imaginary center from the accelerometer, gyroscope, and quaternion values[5]. A complementary filter was applied in order to integrate accelerometer and gyroscope values. With the final angles calculated, it was possible to map three-axis values to the position of the virtual bow, with X and Y mapped to the two-dimensional position of the bow on the screen, and Z to the size of the bow (giving the impression of moving towards and away from the viewer). To indicate pressure on the bow, EMG values were extracted from separate text files containing data of different degrees of pressure (heavy, medium, light, and gradual change). Figure 2 demonstrates the clear difference in EMG values between light pressure on the bow and heavy pressure on the bow.

The computer decides which gesture to show, and in the relevant cases, what amount of bow pressure to assign to it. Every gesture is represented with a number of files (between two to four), to account for variations within the action, such as intensity or placement of the bow. Each gesture, then, can be shown in a variety of ways and bow pressure levels. For the piece, I chose to use six gestures: normal bowing (side to side motion), circular bowing, scraping along the strings, arpeggiation/tremolo, tapping with the bow (hit once), and ricochet (hit and let the bow bounce).

The process of acquiring and using the augmented bow data was much simpler. Since a Jitter window shows the LED light mounted on the bow as a point in space, videos of each action were recorded directly from the Max patch (figure 3). Following each video, I programmed animations that illustrate the movements of this point in space, that is the motion of the bow. While the cello part of the score shows actions as they are, the augmented bow part of the score shows actions as they appear in the Max patch to the performer monitoring the bow's movements. When the computer decides which augmented bow gesture to show, the relevant animation is created in real-time. The animation of each gesture incor-

2 Jitter Tutorial 25: Tracking the Position of a Color in a Movie, https://docs.cycling74.com/max6/dynamic/c74_docs.html#jitterchapter25

3 A Myo armband is a wearable inertial measurement unit (IMU) device, built with eight medical-grade sensors. It provides motion, rotation, and orientation estimation of the device in space, via a three-axis gyroscope, a three-axis accelerometer, and a three-axis magnetometer, as well as muscle electrical activity via electromyography (EMG) [16].

4 Registering the Myo data was done in Max, using the Myo for Max externals by Jules Francoise: https://github.com/JulesFrancoise/myo-for-max

5 Quaternion values were transformed into Euler angles with the jit.quat2euler Jitter object.

porates random elements, such as direction or pace, for the purpose of variation and interest. Here, too, I chose six gestures, some are closely related to the cello ones: normal bowing, drumming with the bow, circular bowing, tapping on the bow, plucking its hair, and "bow crunch" (holding the bow and making crushing motions with the fingers).

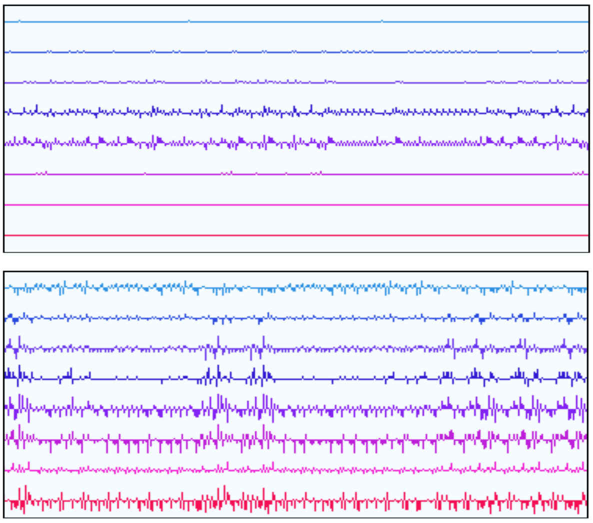


Figure 2. EMG values of bow pressure: light (top) and heavy (bottom).

Figure 3. Video recording of the Jitter window, capturing the motion of the LED diode mounted on the augmented bow.

2.5. Score Design

The score has two levels of function: control function, which is done in Max, and graphic display function, which is done in Processing. The Processing program receives instructions from Max (via Open Sound Control protocol[6]) of which gesture to display and when. While the "when to display gestures?" is governed by fixed timed sections, the "which gesture to display?" depends on the gesture that is currently being played by the augmented bow. Thus, the control function includes a gesture identification module[7]. Once a gesture is identified, a first order Markov chain determines which action to display at a given block of time. The Markov chain receives the current gesture played on the augmented bow, and outputs the next one to display for both instruments. Each identified gesture correlates to different probabilities of which would be the next gesture to display. These probabilities were determined according to natural playing positions and ease of transition between playing actions. For example, if the detected gesture is normal bowing, then there is a 5% chance for the same gesture to be selected next, 30% for drumming with the bow, 20% for circular bowing, 15% for "bow crunch", 15% for tapping on the bow, and 15% for plucking its hair. If circular bowing is detected, the same gesture gets 5% to be displayed next, normal bowing gets 15%, drumming with the bow gets 30%, "bow crunch" gets 15%, tapping on the bow gets 10%, and plucking the hair gets 25%. Finally, the Markov chain's output is sent to Processing at given points in time, according to the fixed formal sections.

The code in Processing comprises different modules for each gesture. When it receives a gesture command from Max, the corresponding module is activated, and the relevant gesture is displayed. The graphic representation of the gestures is generated in real-time. In the case of the cello part, the gesture data from the corresponding text file is also being sent from Max in real-time. Figure 4 shows the overall process of the score's algorithm:

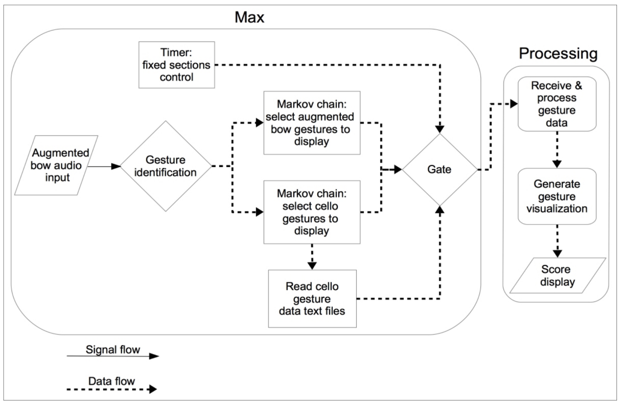


Figure 4. Overall score algorithm flowchart.

The screen itself is divided into two halves, with the cello part on the left side and the augmented bow part on the right side (figure 5). The performers are instructed to position themselves on stage accordingly. They also must sit in such a way that still allows the people in the audience to see the performance actions (i.e. not with the back to the audience).

The graphic notation, in the form of gestures that the performers imitate, looks differently for each instrument. The cello part shows the front of the instrument, including the lower part of the fingerboard and the upper part of the tailpiece. A virtual bow moves on the screen, driven by the gesture data, instructing the cellist what to play, or more correctly, how to play. Pressure on the bow is indicated by the color of the virtual bow: the darker the color, the heavier the pressure (figure 6). The augmented bow part shows a white dot, representing the LED mounted on the bow. The movements of the dot instruct the actions to the performer. Various amounts of delay were added to the white dot, depending on the gesture, for aesthetic pur-

6 The program uses oscP5 library for Processing by Andreas Schlegel: http://www.sojamo.de/libraries/oscp5/

7 While I tried visual gesture identification methods, such as with the MuBu Hierarchical Hidden Markov Models objects [17], the best results were achieved by using audio descriptors, such as a combination of loudness and spectral skewness [18].

poses (figure 7). I also decided to alter the point of view of the augmented bow's part circular bowing animation, showing the circle from above instead of from the performer's actual point of view, which is from the side (figure 8). The change was made in order to show the circular gesture more clearly, since the side point of view could be confusing. This clearer point of view also allows for greater variations in the motion, and it is more aesthetically pleasing than the sideways point of view.

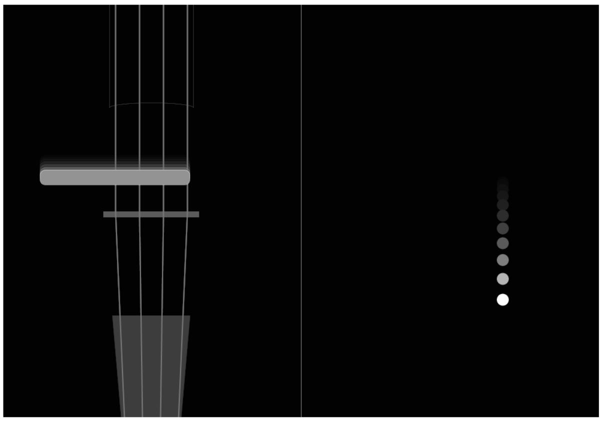


Figure 5. Screen layout: cello part on the left, augmented bow part on the right.

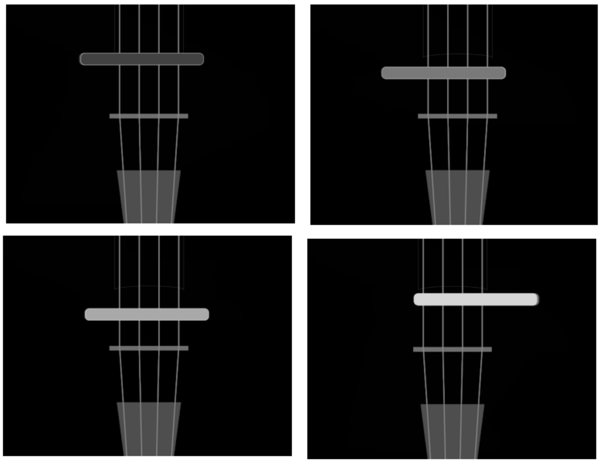


Figure 6. Cello bow pressure range from heavy (top left) to light (bottom right).

Figure 7. Augmented bow part: recreating the LED diode motion, as it appears in the Jitter window, with added delay.

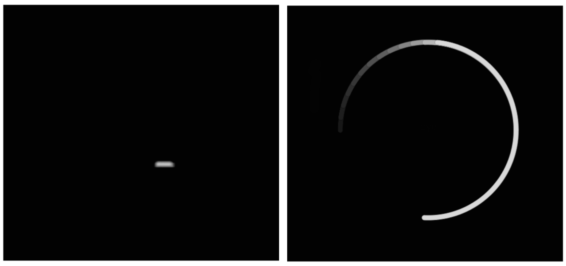


Figure 8. Augmented bow circular motion as captured by video, showing the sideways perspective (left), and as recreated in the score, showing the gesture from above (right).

3. CONCLUSIONS

At the time of writing this paper *Arcos* has yet to be performed. I am unable, then, to determine whether the gesture data approach I took for creating its real-time score was effective in a live performance situation. Only from my own perspective, as the augmented bow performer, I can conclude that interpreting graphic notation that is generated from gesture data is much clearer, and does feel more natural and fluent compared to interpreting text notation. While it is still necessary to learn this type of graphic notation designed for the augmented bow, and to practice interpreting it, I consider it an integral part of the performance practice of this instrument. And, although on a higher level of symbolic representation than the cello part, the bow's notation is still derived directly from the sound morphology of the instrument. It forms a tight relation, or even a sort of an ecosystem, between action, sound, and notation.

Furthermore, this gesture-based type of graphic notation discussed in this paper is especially useful for indicating complicated gestures and different variations within each gesture. However, creating such notation is a lot more energy and time consuming than providing textual instructions.

In addition, the setup of one computer that runs both a Max patch and a Processing program is CPU intensive, and may be too risky in a live performance setting. So far, however, it proved stable in testing. In my opinion, it is also preferable to networking devices, which could be even riskier in live performance.

While this paper concerns a specific composition, it illustrates a possible way to address real-time notation for NIME—using the interface's own data for notation—which other composers can utilize. Additionally, it could offer a direction for future work concerning interpretation by imitation. This attitude towards action notation applied in *Arcos* may prove relevant to real-time scores in general. Research into works that use similar notational procedures could shed more light on this subject as a tool for real-time notation, and maybe even contribute to our overall understanding of the process of sight-reading real-time scores. Such future research might also set the foundations for outlining a conceivable sub-category of real-time action-based notation.

Acknowledgments

Many thanks to David Dalmazzo and Rafael Ramirez for helping with acquiring and processing data, to Leo Morello for recording cello gestures, and to Nir Bitton for advising on the score design.

4. REFERENCES

[1] A . Clay, and J. Freeman, "Preface: Virtual Scores and Real-Time Playing," *Contemporary Music Review*, vol. 29, no. 1, pp 1, 2010.

[2] G. E. Winkler, "The real-time score: A missing link in computer-music performance," in *Proc. of the Sound and Music Computing Conf.*, Paris, 2004, pp. 9–14.

[3] S. Shafer, "Performer Action Modeling n Real-Time Notation," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR'17*, Universidade da Coruña, 2017, pp. 117-123.

[4] S. Shafer, "Terraformation," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR'17*, Universidade da Coruña, 2017, pp. 215-227.

[5] R. R. Smith, *Study no. 15b*, 2013. Accessed on: May 12, 2020. [Online]. Available: http://ryanrosssmith.com/study15b.html

[6] R. R. Smith, *Study no. 15b.1*, 2013. Accessed on: May 12, 2020. [Online]. Available: http://ryanrosssmith.com/study15b1.html

[7] R. R., Smith, "[Study no. 50][Notational Becoming] [Speculations]," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR'16*, Cambridge, 2016, pp. 98-104.

[8] G. Paine, "Gesture and Morphology in Laptop Music *Performance*," in *The Oxford Handbook of Computer Music and Digital Sound Culture* (Ed: R.T. Dean). Oxford University Press, 2009.

[9] T. Wishart, *On Sonic Art*. Routledge, 1996.

[10] D. Smalley, "Spectromorphology: Explaining Sound-Shapes," *Organised Sound*, vol. 2, no. 2, pp. 107-26, 1997.

[11] K. Cascone, "Laptop Music-Counterfeiting Aura in the Age of Infinite Reproduction," *Parachute,* pp. 52-59, 2002.

[12] G. Paine, "Towards Unified Design Guidelines for New Interfaces for Musical Expression," *Organised Sound* , vol.14, no. 2, pp. 142-155, 2009.

[13] *J. Kojs, "Notating Action-Based Music," Leonardo Music Journal*, vol. 21, no. 1, pp. 65–72, 2011.

[14] S. Shafer, "Performance Practice of Real-Time Notation," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR 2016*, Cambridge, 2016, pp. 65-70.

[15] L. Vickery, "The Evolution of Notational Innovations from the Mobile Score to the Screen Score," *Organised Sound*, vol. 17, no. 2, pp. 128–136, 2011.

[16] D. Dalmazzo, and R. Ramírez, "Bowing Gestures Classification in Violin Performance: A Machine Learning Approach," *Frontiers in Psychology,* vol. 10, no. 344, 2019, doi:10.3389/fpsyg.2019.00344

[17] J. Francoise, N. Schnell, R. Borghesi, and F. Bevilacqua, "Probabilistic Models for Designing Motion and Sound Relationships," *in Proc. Int. Conference on New Interfaces for Musical Expression*, London, 2014, pp. 287-292.

[18] M. Malt, E. Jourdan, "Real-Time Uses of Low Level Sound Descriptors as Event Detection Functions Using the Max/MSP Zsa.Descriptors Library," in *Proc. of the 12th Brazilian Symposium on Computer Music,* Recife, 2009, pp. 45-65.

FILTERING AND SYNTHESIS: IT-BASED APPROACH IN THE COMPOSITIVE PROCESS OF GIOVANNI VERRANDO AND FAUSTO ROMITELLI

Luca Befera

Dipartimento di Musicologia e Beni Culturali,
Università di Pavia
luca.befera01@universitadipavia.it

Luca Guidarini

Dipartimento di Musicologia e Beni Culturali,
Università di Pavia
luca.guidarini01@universitadipavia.it

ABSTRACT

To investigate the influences that information technology has brought to musical writing and compositional process, the cases of Giovanni Verrando and Fausto Romitelli – Italian post-spectral composers of the 1960s generation – will be examined. After their studies at IRCAM in Paris, they integrated tools of digital sound treatment and musical form management into their notation, opportunely mediated by software and informatics tools. Two fundamental aspects of their language will be investigated: the cyclical organization of the form, and timbral generation techniques as filtering for Verrando and synthesis for Romitelli. It will be demonstrated how the principles indicated, deriving from specific methods of digital treatment of sound, are applied to the creation of complex timbres reproduced by the orchestra or by the ensemble. In both cases, the results are a writing process, compositional way of thinking and sound outcome hybrid between acoustic and electronic, determining a formal management mediated by sound data and according to a prominently IT-based approach.

1. INTRODUCTION

During the 20th century, technology became an integral part of the sociocultural environment, determining significant changes in various fields of human activity according to a technomorphic process.[1] This reflected a different conception of the cultural objects, considered as data of a heterogeneous nature. The proliferation of software, the web, personal computers or, more generally, digital devices in everyday life, has invested an expanding catchment area, manifesting itself in a different way of conceiving reality [2, 3, 4]. In the theoretical field, this has evolved from studies in the area of information theory and, in parallel, to computer science [5, 6]. The binary code revolution caused a progressive substitution of the digital perspective over the analogue one, towards a widening of the communicative possibilities mediated by virtual platforms [7, 8]. Therefore, cultural objects have also acquired the value of data, which can be managed with the help of computers and which, in an equal and opposite reaction, are influenced by digital language.

The use of informatics or, more generally, electronic devices as a tool to support or expand creative and performing possibilities affected also the musical domain. Starting from the pioneering studies done in the '50s and '60s in well-known centers such as Club d'Essay in Paris, WDR Studios in Cologne, Bell Laboratories in New York and Studi di Fonologia Musicale Rai in Milan [9], the experimentation on music software became a focal point of some institutions' research during the '70s, among which IRCAM held a central role. There, the spectral composers found an ideal place to develop their timbral research: composing "the sound" and not "with sound" [10, 11] became one of the key principles of their studies by means of software for sound analysis and production, or, namely, for the management of the spectral content. At the same time, some synthesis models were used to manage the pitches organization, to reproduce complex sounds by orchestral rendering or generate sequences of sonic events according to algorithms or abstract criteria. Therefore, the mediation of informatic procedures and software such as Max/MSP, PatchWork, OpenMusic, Audiosculpt, and Csound assumed the dual function of

[1] The term, applied by Eric Maestri in the musicological field, was derived from the science and technology scholar Linda Caporael, who defines 'technomorphism' as "the attribution of technological characteristics to humans" [1]. The concept was discussed by the former author regarding the sound parameter in the speech Timbre is a technomorphic thing: A comparative analysis of three case studies, held during the conference Timbre is a Many-Splendored Thing on 4-7 July 2018 at McGill University in Montreal.

means for sound production and tool to manage the formal organization [12].

Within this paper, these dynamics will be investigated in the production of Giovanni Verrando and Fausto Romitelli, Italian composers of the '60s generation who studied at IRCAM and integrated into their compositional approach elements derived from informatics and electronic practice on the basis of what already done by the spectral authors of the previous generation. The parallel formation path of the two composers started from the education at the Conservatory of Milan and the courses of Donatoni at the Civica Scuola di Musica and Accademia Chigiana, then proceeded in the *Cursus de composition et informatique* at IRCAM – Romitelli from 1990 to 1991, then remaining in the Parisian institution as Compositeur en recherche from 1993 to 1996, and Verrando from 1993 to 1997 [13, 14]. The ways of managing sound and form were therefore influenced by common roots, although each one of them would have then developed his specific means of expression.[2]

The cases of the filtering in Verrando's *Filtering* – first movement of *Triptych* (2005-2006) for large orchestra and electronics – and the synthesis in Romitelli's *Hellucination I – Drowningirl* – third movement of *An Index of Metals* (2003) for soprano, ensemble, electronics and video – will be investigated, since these pieces, identifying the stylistic evolution of each author, exemplify the influences of information technology (IT) in their compositional approach. The cases presented in the paper are the results of a research that focuses on the precompositive material analysis, both by a sonic and formal point of view. The analysis of the IT derived sketches and preparatory materials underlines a close relation between their compositional practice, especially in the macro-formal and sonic construction domains. In both the pieces analyzed, the form is conceived in a cyclical way, but with different purposes: in *Filtering*, the sounds are organized as discretized and clearly perceptible units; in *Hellucination I – Drowningirl* the modular construction is conceived in a way that sonically avoids the perception of the cycle. Therefore, each author manifests a compositional approach based on modules, to construct a likewise segmented form or to generate a continuous process where the modules are not even recognizable. Moreover, the sonic perspectives are complementary: in the first case, filtering is a useful process to manage the spectral content while synthesis is a means for sounds production; in the second, filtering is an informatics and symbolic tool while synthesis is a model through which the spectral content is oriented. The specific dynamics within which filtering, and synthesis become part of the compositional process and formal structuring with the help of informatics tools will be further analyzed.

2. GIOVANNI VERRANDO: FILTERING

2.1. Introduction

The experience in Paris was particularly relevant for Giovanni Verrando: from then on, the composer began to implement his research on sound through informatic devices. Since 2005, this aspect has been integrated by experimentation on timbral possibilities of acoustic instruments, in what he defines 'new lutherie' as "the stable introduction of new instruments into the ensemble and a more general openness to the evolution of lutherie" [16]. This research is closely connected with the usage of informatics tools: the new timbral possibilities are also implemented by rendering electronic sounds - often obtained through entirely digital processes - by means of the acoustical ensemble. This process involves the comparison between the acoustic and electronic spectra, conveyed towards a hybrid outcome. Moreover, this approach is mediated by the psychoacoustic science deepened during the IRCAM education, where both the extended techniques and software research became useful tools for a deeper study of timbre and syntactic management of sounds. The notation was therefore influenced by sounds generation and organization, mediated by software such as Audiosculpt for the definition of the timbral content and Max/MSP for the management of the discrete objects that contain it.[3] In the two following sections, these dynamics will be respectively investigated, taking as an example some excerpts from *Filtering*, first movement of *Tryptych* (2005-2006) for large orchestra and electronics.

2.2. Timbral content

In *Triptych*, all the synthesized - and then re-orchestrated - sounds come from 38 predetermined samples [14, 16]. The composer works with Audiosculpt to analyze or resynthesize the sounds, modelling or selecting the spectral components of samples taken from other pieces or previously generated by himself through digital synthesis: from the continuous comparison between the graphic result visible in the sonogram, the effects generated by the instruments and the relative acoustic perception, he orients the compositional process towards the desired sonic result. This process implies a close relationship between electronic and acoustic sounds. To clarify this dichotomy, Eric Maestri defines the categories of 'endogenous movement' (modifiable within itself) and 'exogenous movement' (relative to the gesture): "Electronic sound is characterized by the expansion of the envelope, and in particular of the sustain, which becomes the centre of sound evolution and the moment in which the difference between the instrument and the electronics is decisive. [It

[2] For more information about the interconnections between the two authors, see the thesis *Sincronie: Interconnessioni Formali tra Nova, Verrando, Romitelli e l'Electronic Dance Music negli Anni '90*, which examines their collaboration in the Nuove Sincronie and Sincronie festivals and some aspects of a common compositional approach through information theory and informatics tools, especially at a syntactic level [15].

[3] The composer himself attests the usage of the software in the interview realized by Luca Befera on August 7th, 2019.

becomes fundamental] the control of the evolution of sound over time; the change of energy in the spectrum without re-attacking the sound. In this way, endogenous and exogenous movements change the hierarchy. Where in instrumental music it is the gesture-carried sound that dominates, in electronic music it is the texture-carried sound; the decisive difference between electronic and instrumental sound seems to me to be given by the different evolution of envelope maintenance" [17]. Although computer processing allows multiple gradients between these categories, as well as, in some cases, it is possible to control the envelope through the instrumentation, the categories are useful to describe macro-tendencies of the spectral content, which are expressed in a perceptual dichotomy or - if the composer himself wants to recreate electronic effects through acoustic gestures as in this case study - in hybrid sonic results.

In addition to the dialectic relationship between endogenous and exogenous movement, Verrando manages the formal structure according to the perceptive differences between inharmonic, aperiodic, harmonic and sinusoidal sounds [16]. These principles derive from the psychoacoustics studies made in the '90s by Albert Bregman and Stephen McAdams on the 'auditory scene analysis'. This theory is based on the examination of how the brain processes acoustical images of certain sounds to recognize their location and connotation. Given two simultaneous sounds, the listener will or will not split their perception depending on their sound qualities and, consequently, on the relationships that he established between them. An 'auditory stream' is defined as "our perceptual grouping of the parts of the neural spectrogram that go together" [18] or, in the definition of his proselytes Daniel Pressnitzer and Stephen McAdams, as "a sequence of events that can be considered to come from a same source" [19]. In both cases, listeners segregates the stimuli present in a sound input system through their perception, defined as 'auditory scene'. The streams are discretized according to the spectral composition of the sounds, their location within space and different rhythmic patterns. Verrando uses Audiosculpt to work precisely on the segregation of auditory streams by observing and modelling the spectral content through the software's graphical interface. In the example in Figure 1 (relative to what the composer defines as *Triptych*'s 'sound 31'[4]) it is possible to observe waving bands cancelled in the spectrum to recreate an oscillation of the inharmonic frequencies within the sound. Figure 2 shows another approach, in which Audiosculpt is used for spectral analysis only. The sonogram at the top left refers to a sample taken from Rasite, the sixteenth track of the album Aaltopiri by Pan Sonic: observing and listening, the composer works to achieve the most similar sonic result,

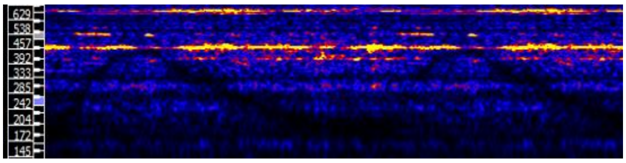


Figure 1. Inharmonicity oscillations in the 'sound 31' of *Triptych*.

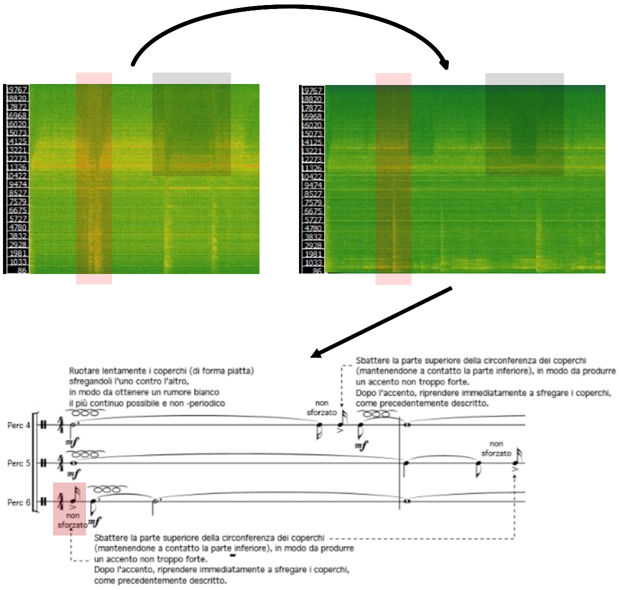


Figure 2. Comparison between Pan Sonic's *Rasite* ex-cerpt (top left) and 'sound 2' of *Triptych* (top right); the squares show the spectral affinities and the relative nota-tion.

using cymbals (top right). The resulting adiastematic notation (bottom) indicates just the durations of the percussive accents and rubbing of the cymbals, while the specific instrumental technique is explained in the word-written instructions. Figure 3 shows a further instrumental resynthesis: starting from an additive synthesis of sinusoidal sounds – loaded in the sampler (bottom-left) – the composer generates a harmonic aggregate (top-left), which is rendered through the ensemble and represented in diastematic notation (right). Therefore, these are not sinusoidal but complex timbres, whose perceptual result tends to be harmonic.

Hence, each kind of notation aims to translate a specific timbral category, through diastematic/adiastematic notation or by simple words explaining the gestures. The peculiar sonic result, focused on the spectral implications, is mainly generated by the materiality of non-conventional objects and instrumental extended techniques derived, above all, from the endogenous/electronic content at the very basis of the piece.

[4] The samples, part of which are shown in Figures 1, 2 and 3, were kindly given to me by Verrando on August 13, 2019. In the file there are four folders: the first three are related to different savings of the transcription on Finale of the orchestral score; the last one, called 'Triptych-Sampler-Max', presents the 38 samples to be used in the synthesizer and the Max/MSP patch for the application of these on the keyboard. The synthesizer reproduces either the electronic sounds that cannot be realized with the instrumentation used (e.g. sinusoidal sounds) or double the orchestral sound prerecorded in the samples. The approach highlights a retrospective assembly of discrete elements previously obtained.

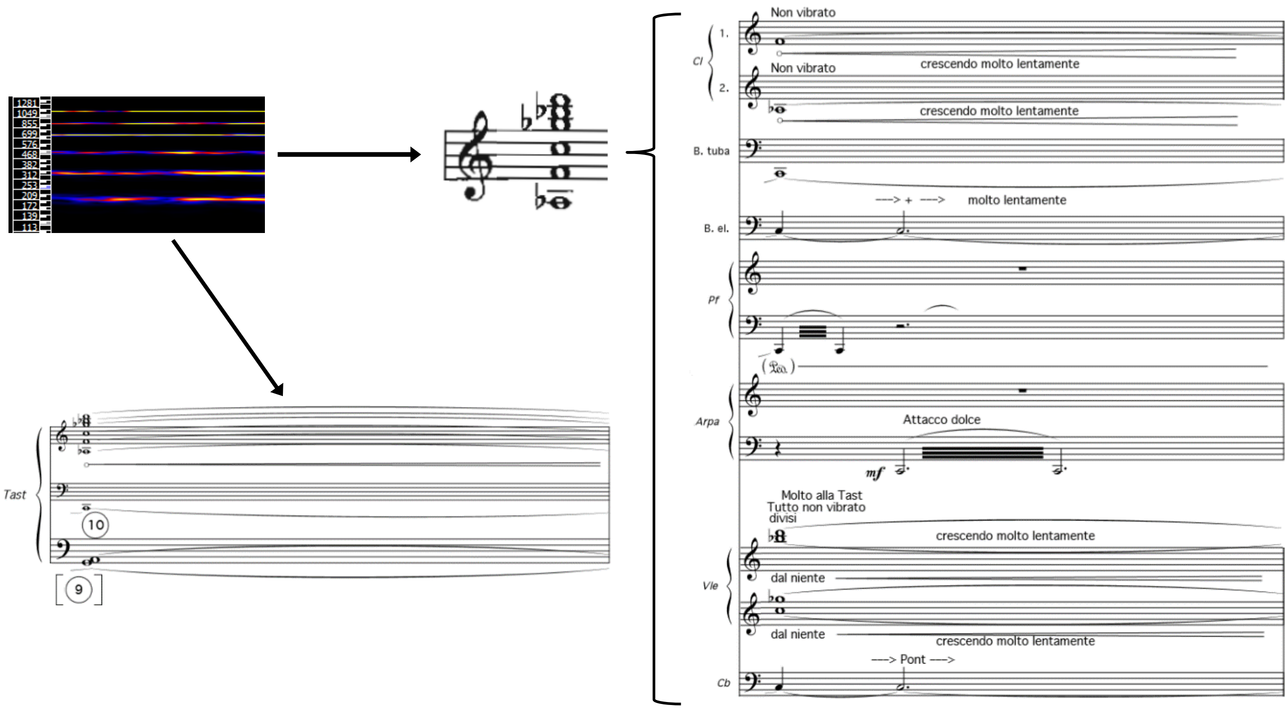


Figure 3. Transcription of the 'sound 10' (sonogram on top-left), loaded in the sampler ('Tast', bottom-left) and re-orchestrated in the score for clarinet, bass tuba, electric bass, piano, harp, viola and double bass (right). The written notes are concert pitches.

2.3. Syntax

The spectral content just described is inserted in clearly defined discrete units. The 38 synthetic sounds, from which the instrumental orchestration comes from, are loaded in a sampler on Max/MSP [15]. Each one of these is marked in the score through the circled numbers (e.g. Fig. 3, sampler score on bottom-left) and reproduced in the whole play. The usage of the pre-compositive material as circumscribed informatic units reflects also the structure of the piece, organized by 'blocks' (numbered in Fig. 4) structured on data of spectral nature. The blocks content is managed by Verrando according to the oppositional categories described in the previous section (inharmonic, aperiodic, harmonic or sinusoidal sounds), which, by contrast or affinity, characterize also the blocks succession. The repetition of the same or similar elements (letters in Fig. 4) plays an important role in the macro-formal distribution, defining a cyclicity that operates on medium- and long-range.

Regarding the organization of sounds, Ingrid Pustijanac has already observed: "the materials are structured according to these two possibilities: juxtaposition or overlapping of only two harmonic planes" [14]. Beyond the simple consequentiality of the timbral blocks already discussed, this means that it is possible to identify the superimposition of multiple auditory streams within them. These streams are segregated according to the 4 spectral categories already discussed. In Table 1, a possible coding of the elements contained in the first 8 blocks is proposed, defined as follows: harmonic sounds (*a*), sinusoidal sounds (*s*), inharmonic sounds (*r* or *p* if generated by the cymbals) and impulses (*i*). The subscripts indicate different types of these categories, whereas the initial aperiodic sounds (*ap*) are recreated by playing the cymbals with the bow, generating sounds between the harmonic and metallic connotation. The sum indicates the overlapping of several elements: although there is a 'polyphony of timbres' on the whole - indicating a sort of 'global sonority' of a musical signal, which consists of several inner components [22] - by separating the internal streams it is possible to argue perceptual interconnections between different blocks. Therefore, in the last two columns, where the predominance of some elements has been inferred [15], it

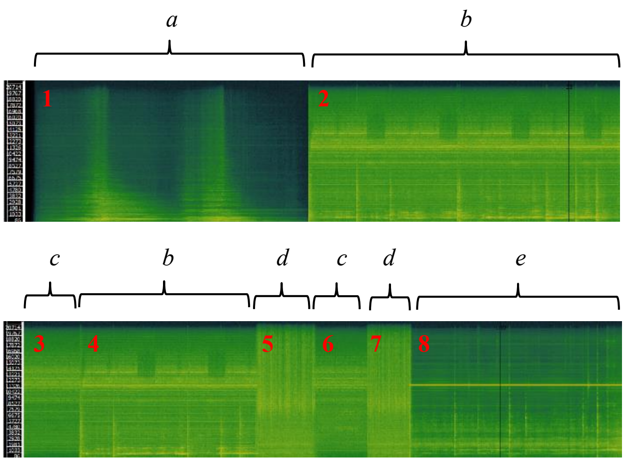


Figure 4. Formal distribution of 'blocks' over time in the beginning of the piece.

Blocks	Duration in score (in quavers)	Elements sequences	Predominant elements	
1	13	(ap_1)	$(a)(p)$	$(a)(p)$
2	20	$(s+p_2+i)$	$(s)(p_2)$	(p_2)
3	4	$(s+r_1)$	(r_1)	(r)
4	12	$(s+p_2+i)$	(p_2)	
5	4	(r_2+r_3)	(r_2)	
6	4	$(s+r_1)$	(r_1)	
7	3	(r_2+r_3)	(r_2)	
8	16	$(s+p_2+i)$	$(s)(p_2)$	(s)

Table 1. Coding of the elements contained in the first 8 blocks.

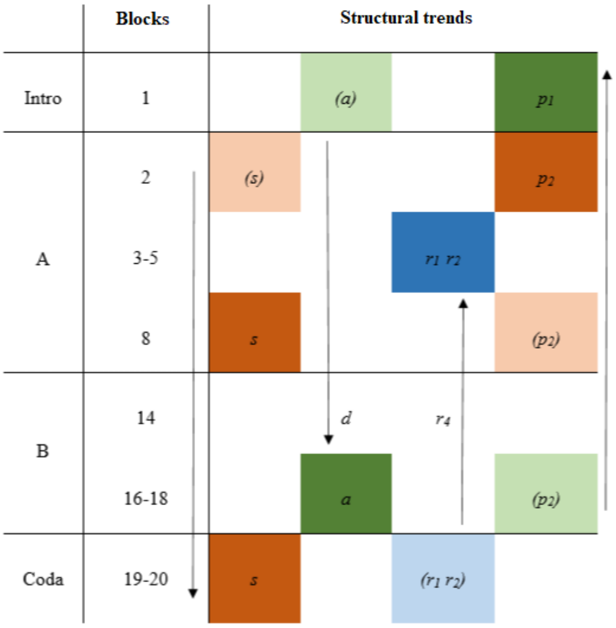


Table 2. Macro-formal distribution of the auditory streams in the first movement of *Triptych*, *Filtering*.

is possible to observe some timbral processes operating on the form as a whole: within the first 8 blocks, a progressive transformation occurs from the initial aperiodic sonority to the sinusoidal sound of block 8, passing through different types of inharmonic sounds. By extending this consideration to the macro-form (Table 2), it is possible to observe the repetition of some internal streams on a wider range. Each time, the different combinations of the same sounds generate specific polyphonies of timbres. Within these, the repeated elements maintain a parallel - though not simultaneous - flow, producing processes of internal mutation of the spectral components.

Summarizing, informatics spread across the whole piece and influences the very routes of the compositional process. First of all, the electronic – or electronic derived – sounds generated by software predominate and are inserted in discrete and clearly recognizable elements. Secondly, the timbral material is managed according to quantitative criteria related to the spectral content and inserted in discrete elements assembled by juxtaposition or overlapping of sounds. Finally, the identical repetition of some structural functions resembles a digital treatment of sound, whose polyphony of timbres is branched into smaller streams or spectral data. The syntactic structure, partly derived from spectral practice, corresponds in Verrando to a specific procedure, which finds in 'new lutherie' a peculiar realization. The resulting notation is related to digital processes, which are later translated into the score to ensure an easy intelligibility by acoustic instrumentalists. However, it always maintains strong informatics and electronic connotation, manifested in sonic results and formal management.

3. FAUSTO ROMITELLI: SYNTHESIS

3.1. Introduction

Since his first approach with the IRCAM cursus in 1991, Romitelli introduces computer aided composition knowledge – a topic of interest at the Institute since the 80s – in his practice. With the help of Laurent Pottier, he developed a personal PatchWork library [23], used for the pieces written in the '90s. In the 2000s, Romitelli started to work with the OpenMusic language[5], mostly to create and organize fixed harmonic fields in a personal compositional discourse. Unlike Giovanni Verrando's case, Romitelli doesn't start from a preexistent spectrum in order to resynthesize the exact auditory image, but rather from a sound-synthesis criterion that symbolically – thanks to PatchWork and OpenMusic – generates pitches aggregates of imaginary complex spectra to be orchestrated. The actual state of research doesn't permit to examine Fausto Romitelli's personal computer – preserved by the Giorgio Cini Foundation in Venice (Fondo Fausto Romitelli, FFR from now on) – but the sketches-driven analysis permits to investigate the symbolic synthesis process that allows him to create the formal and pitch organization of *Hellucination I – Drowningirl*, the third section of *An Index of Metals* (2003), video-opera for soprano, ensemble and electronics.

3.2. Sources

Currently preserved by the Giorgio Cini Foundation in Venice (Fondo Fausto Romitelli), the sources analyzed for this research were conserved in a mixed folder - that contains sources related to *An Index of Metals, Domeniche alla periferia dell'Impero* (1996/2000), *Seascape* (1994), and notes for lessons – and consist in:

[5] Personal call with Luca Guidarini, 12.06.2018. Paolo Pachini was the musical informatic and video maker that helped Romitelli in *An Index of Metals*.

A. Three pages of annotations, schemes, and tables for the control of time, density and dynamics (Figs. 5a and 5b);

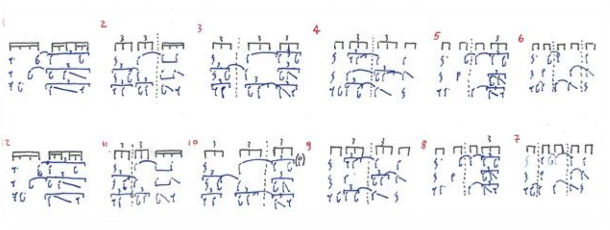


Figure 5a. First two pages of A resumed, FFR, transcription.

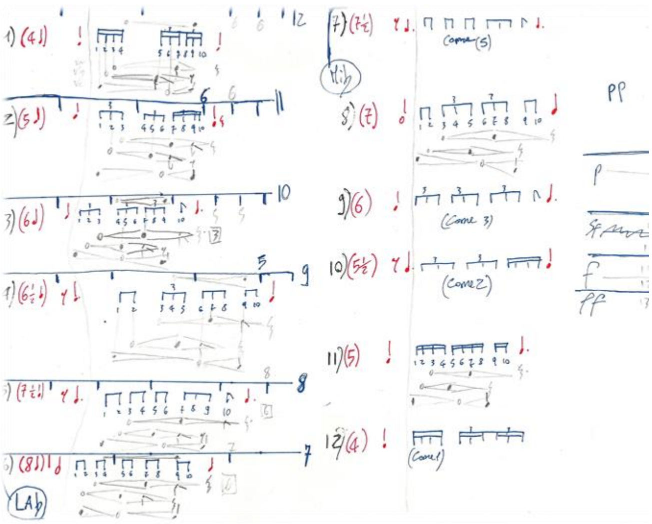
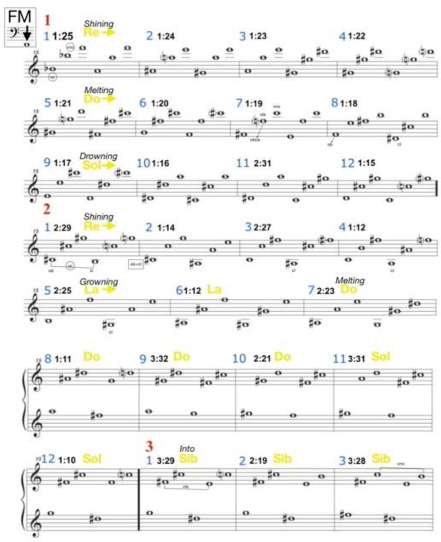


Figure 5b. Third page of A, FFR, transcription

B. Three double-sided pages of sketches, with references to a and c. (see Fig. 6);



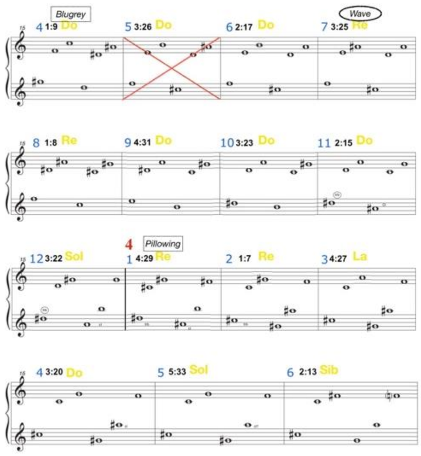


Figure 6. First two pages of B, FFR, transcription.

C. One glossed page of B. Truax, "Organizational Techniques for C:M Ratios in Frequency Modulation" [24].

0:1	1:32	1:31	1:30	1:29	1:28	1:27	1:26
1:25	1:24	1:23	1:22	1:21	1:20	1:19	1:18
1:17	1:16	2:31	1:15	2:29	1:14	2:27	1:13
2:25	1:12	2:23	1:11	3:32	2:21	3:31	1:10
3:29	2:19	3:28	1:9	3:26	2:17	3:25	1:8
4:31	3:23	2:15	3:22	4:29	1:7	4:27	3:20
5:33	2:13	5:32	3:19	4:25	5:31	1:6	5:29
4:23	3:17	5:28	2:11	5:27	3:16	4:21	5:26
6:31	1:5	6:29	5:24	4:19	3:14	5:23	7:32
2:9	7:31	5:22	3:13	7:30	4:17	5:21	6:25
7:29	1:4	8:31	7:27	6:23	5:19	4:15	7:26
3:11	8:29	5:18	7:25	9:32	2:7	9:31	7:24
5:17	8:27	3:10	7:23	4:13	9:29	5:16	6:19
7:22	8:25	9:28	10:31	1:3	11:32	10:29	9:26
8:23	7:20	6:17	11:31	5:14	9:25	4:11	11:30
7:19	10:27	3:8	11:29	8:21	5:13	12:31	7:18
9:23	11:28	2:5	13:32	11:27	9:22	7:17	12:29
5:12	13:31	8:19	11:26	3:7	13:30	10:23	7:16
11:25	4:9	13:29	9:20	14:31	5:11	11:24	6:13
13:28	7:15	15:32	8:17	9:19	10:21	11:23	12:25
13:27	14:29	15:31	1:2				

Figure 7. C:M ratio series of order 32, table from B. Truax [24].

3.3 Control of time and cyclical organization

The first two pages of A contain twelve 3-voices modules (Fig. 5a). A rhythmic figure, which structures the glissato gesture of the strings over time and controls its duration (for a total of 42 quarter notes) in a progressive way, can be seen on the top of each module. The modules mirror around an axis represented by the 6th and 7th modules. They express the cyclical nature of the formal organization: the twelve-module cycle is repeated eleven times during *Hellucination I – Drowningirl*; the last repetition of the cycle is interrupted at the moment of its maximum length (at the 6th module). An ordinate series of numbers (1-13) expresses the formal scheme of cyclical repetition (see Fig. 5b, right side). Every cycle introduces elements that increase the global density in the formal process that expresses a general crescendo, both in the strings and winds/brass. The internal organization of the modules, with the help of the glissato gesture on the strings and of the electronic sounds is conceived in a way that avoids the perception of the cycles that construct the section.

Three gestures (Figs. 8 to 10) that produce a gradual complexity of sonic material are assigned to the winds/brass instruments, and their use depends on the density condition:

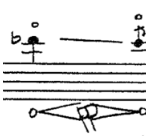


Figure 8. Gesture 1.

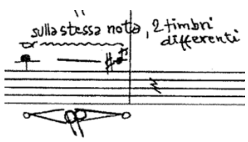


Figure 9. Gesture 2

Figure 10. Gesture 3.

The remainder of the third page of annotations (Fig. 5b) is occupied by a scheme that seems to be useful to organize dynamics peaks over time. Romitelli, for that, uses the same temporal modules of the other two pages.

3.4 Pitch organization and FM synthesis

The source B consists of a double-sided bifolio and a single double-sided page (Fig. 6). It contains a scheme with 150 bars of pitch aggregates, with numerical references to A and C, some orchestration indication, textual references to the soprano part and harmonic references to the guitar, bass and keyboard parts. The genetic criteria of those aggregates are explained by two details of the sketch: a box with the indication “FM D1[notated]” and the ratios written above every aggregate. ‘D1’ indicates the nominal – and hidden through the process – frequency of a Frequency Modulation, whose ratios are expressed over every aggregate. C is a glossed page of Truax, Barry [24] and contains an organizational table representing the C:M ratio series of order 32 (Fig. 7). Truax used the first half of a Farey sequence of order 32 due to creating a gradual descending of the modulator frequency from 1:32 to the harmonic ratio 1:2 that divides symmetrically the whole series.

Frequency Modulation was used in the so-called instrumental synthesis since Murail’s *Gondwana* (1980) [25] as a static technomorphic metaphor useful to create complex sounds; in the case of *Hellucination I – Drowningirl*, FM assumes structural relevance since the organization of the pitch aggregates is driven by the process Truax explained in his table. Starting from that and from the pitch aggregates of the source B it was possible to recreate on Max8 – using *bach* and *cage* libraries[6], especially the object *cage.fm* - the OpenMusic patch he used to create the symbolic FM, from the fundamental frequency and the ratios between the carrier and modulator frequency, with a fixed modulation index of 1 (Fig. 11).

Figure 11. Patch reconstruction on Max8, *presentation mode*.

The patch operates with an approximation of semitones, meaning that the principle behind the instrumental synthesis of Romitelli is not the aural accuracy of a timbre, but the creation of interesting-hearing, and processional driven group of fixed harmonic fields.

3.5 From sketches to score

Figure 12 shows the realization of the first formal cycle in the strings, in which the density is at its minimum point. The numbers of the formal modules correspond to those of Figure 5, the ratios are those of Figure 6.

[6] The *bach* and *cage* libraries are collections of modules to deal with Computer-aided composition in Max, developed by Andrea Agostini and Daniele Ghisi (https://www.bachproject.net/, accessed 18.04.2020).

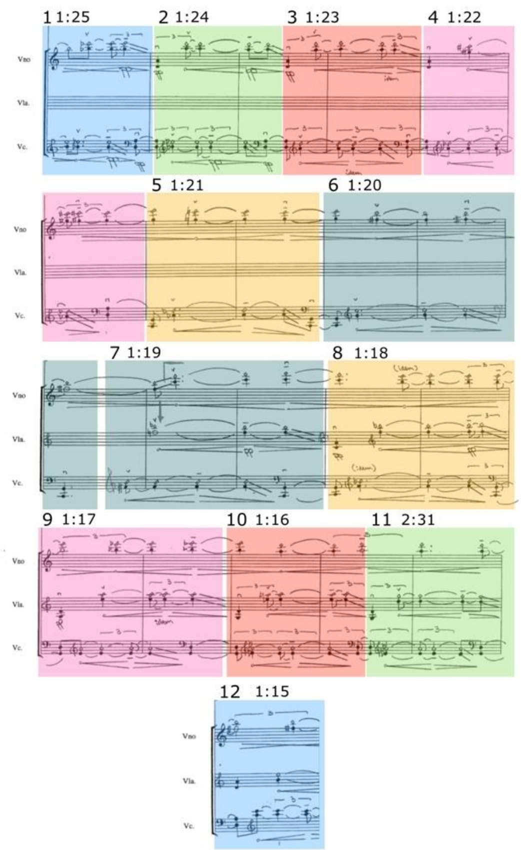


Figure 12. Pages 12-14, strings section score.

The score corresponds to the sketches within three exceptions:

1) The 3rd repetition of the cycle skips the 5th module. As a result, the text is sung without interruption in the vocal part.
2) In the 3rd module of the 7th repetition of the cycle Romitelli changes the pitch content in order to preserve the semitonal descending line of those pitches, without this correction the applications of Farey's ratio (8:31) would create an interruption of the descent.
3) The last two and a half cyclical repetitions present a different pitch organization in the winds/brass. In the previous repetitions those instruments doubled, echoing the pitches of the strings. In those last cases they become autonomous and play independent pitch aggregates from B (Fig. 6). As a result, there is a temporal compression of the process duration and an increase of the textural density.

3.6 Electronics

The 11-channels electronics of *An Index of Metals* has three functions:

1. Integration with the instruments;
2. Articulation of a single event;
3. Sampling from other's music: the *Introduzione* from Pink Floyd's *Shine on You Crazy Diamond* and the *Intermezzi* from Pan Sonic's works.

According to Paolo Pachini[7], who helped Romitelli in the realization of some electronics, *Hellucination I – Drowningirl* refers to the first function. Pachini filtered a sound of a bowed metallic tube with a CSound vocoder, then refiltered and processed it with the FM instructions Romitelli gave to him. Probably the indications were similar to the instrumental one since the pitches of every single spectrum correspond to the instrumental and the simulated ones.

3.7 Multi-layered form

Formally, the piece presents an independent harmonic line formed by the soprano, the electric guitar and bass, and the keyboard runs his own process (Table 3) alongside the instrumental synthesis.

	1(1)	2(1,5)	3(1,5)	4(2)	5(2)	6(2,5)	7(2,5)	8(2)	9(2)	10(1,5)	11(1,5)	12(1)
1	(A)D				C				G			
2	D				A		C				G	
3	Bb			C			(A')D			C		G
4	D		A	C		G	Bb			C		
5	(A'')D	C	A	G	D	A	C	G	Bb		C	
6	(B) D	G	Bb	D		G	Bb	D		G	Bb	C
7	(A'')D	C	A	G	D	A	C	G	Bb		C	
8	(C) D		Eb	F	G	Bb	D		G	Bb	C	C#
9-10	(A'')D	C	A	G	D	A	C	G	Bb		C	
11-12	(C') D		Eb	F	G	Bb	D	G	Bb	D	C	A
13	(D)G	D	A	C	G	Bb	D					

Table 3. Harmonic line of the guitar, bass, and keyboard according to the 12 cycles.

The whole harmonic line can be represented as: A (b. 56 – 103) A' (104 -137) A'' (138 – 158) B (159 – 179) A'' (180 – 200) C (201- 221) A'' (222 – 242) C' (243-263) D (264 – 276). The first A is dilated from the start to the end of the 6th module of the 3rd cycle. A' is a variation of A, compressed in one and a half cycle.

4. CONCLUSIONS

The two analyses, although based on different assumptions, demonstrate the influence in writing and notation of an IT approach. The authors, coming from a similar background where of great importance were the practices studied at IRCAM, moved towards their specific experimentations discussed in relation to the considered pieces. The software usage, strictly connected to a spectral approach, is implemented towards specific modalities of sound generation and control of formal organization. The use of filtering and synthesis processes is emblematic of these dynamics, as practical applications of abstract informatic models. Within the macro-form, the internal processes are stratified in different ways, not explicitly perceived, but organized to generate cyclical repetition and

[7] Personal call with Luca Guidarini (see footnote 5).

continuity. Layering and modularity became two fundamental principles to manage the spectrum possibilities, generated from and structured through an IT-based approach. It is also possible to discern wider considerations through these assumptions: their compositional models belong to a technomorphic environment which, consolidated in art music context in the 70s also thanks to the Parisian institution, took specific ramifications starting from the 2000s. The musical language becomes more and more permeated by digital influences both at structural and computer-aided processes level, as fundamental elements in the authors' poetics.

5. REFERENCES

[1] L. R. Caporael, "Anthropomorphism and mechanomorphism: Two faces of the human machine," *Comput. Hum. Behav.*, vol. 2, pp. 215–234, 1986.

[2] M. Buckland, *Information and Society*. Cabridge: MIT Press, 2017.

[3] F. Webster, *Theories of the Information Society*. New York: Routledge, 2014.

[4] C. May, *Key Thinkers for the Information Society*. New York: Routledge, 2002.

[5] J. Gleick, *The information: A history, a theory, a flood*. London: Fourth Estate, 2011.

[6] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge: Cambridge University Press, 2005.

[7] L. Manovich, *Software Culture*. Milano: Olivares, 2010.

[8] L. Manovich, *The Language of New Media*. Cambridge: MIT Press, 2001.

[9] P. Manning, *Electronic and computer music*. New York: Oxford University Press, 2004.

[10] G. Grisey, "Zur Entstehung des Klanges," *Darmstädter Beiträge zur Neuen Musik*, vol. 17, pp. 73–79, 1978.

[11] F. Romitelli, "Il compositore come virus," in *Il corpo elettrico: viaggio nel suono di Fausto Romitelli*, A. Arbo, Ed. Monfalcone: Teatro Comunale, 2003, pp. 81–83.

[12] G. Assayag, "Computer assisted composition today," in *Applications on contemporary music creation: Esthetic and technical aspects*, 1998.

[13] A. Arbo, Ed., *Anamorphoses: Etudes sur l'oeuvre de Fausto Romitelli*. Paris: Hermann, 2015.

[14] I. Pustijanac, "La ricezione del pensiero spettrale negli ultimi due decenni in Italia," in *Italia/Francia. Musica e cultura nella seconda metà del XX secolo. Atti del Convegno internazionale di studi*, 2009, pp. 347–372.

[15] L. Befera, "Sincronie: Interconnessioni Formali tra Nova, Verrando, Romitelli e l'Electronic Dance Music negli Anni '90," Università di Pavia, Cremona, 2019.

[16] G. Verrando and AA.VV., *La nuova liuteria: Orchestrazione, grammatica, estetica*. Milano: Suvini Zerboni, 2012.

[17] E. Maestri, "Movimento esogeno e endogeno in Professor Bad Trip I," in *Have your trip: La musica di Fausto Romitelli*, V. Santarcangelo, Ed. Milano: Auditorium, 2014, pp. 117–141.

[18] A. S. Bregman, *Auditory scene analysis: the perceptual organization of sound*. Cambridge: MIT Press, 1994.

[19] D. Pressnitzer and S. McAdams, "Acoustics, psychoacoustics and spectral music," *Acoust. psychoacoustics Spectr. Music. Contemp. Music Rev.*, vol. 19, no. 2, pp. 33–59, 2000.

[20] M. V. Mathews and J. E. Miller, *The Technology of Computer Music*. Cambridge: MIT Press, 1969.

[21] G. Castagna, *Object-oriented programming: A unified foundation*. Boston: Birkhauser, 1997.

[22] J.-J. Aucouturier, "Dix Expériences sur la Modélisation du Timbre Polyphonique," Thèse de doctorat de l'Université Paris, 2006.

[23] A. Olto, "Between spectrum and musical discourse. Computer Assisted Composition and new musical thoughts in EnTrance by Fausto Romitelli," *Sounds, Voices and Codes from the Twentieth Century. The critical editing of music at Mirage,* L. Cossentini and A. Orcalli, Ed. Udine: Mirage, 2017, pp. 419-452.

[24] B. Truax, "Organizational Techniques for C:M Ratios in Frequency Modulation," *Computer Music Journal*, Vol. 1, No. 4, pp. 39-45, 1977.

[25] F. Rose, "Introduction to the Pitch Organization of French Spectral Music". *Perspectives of New Music,* Vol. 34, No. 2, pp. 6-39, 1996.

[26] M. Spicer, "(Ac)cumulative Form in Pop-Rock Music", *Twentieth-century music,* Vol. 1, No. 1, 29-64, 2004.

ACTION SCORES AND GESTURE-BASED NOTATION IN AUGMENTED REALITY

Giovanni Santini
Hong Kong Baptist University
info@giovannisantini.com

ABSTRACT

Augmented Reality (AR) is becoming, year by year, an established and well-known technological resource. Experimentations and innovative applications are produced in different areas. In music, there already is some use of such a technology in the fields of education and performance. However, the use of AR features as compositional resources has yet to be deeply explored and leaves room for innovative research. In particular, the possibility of notating the gesture in space, instead of on paper or screen, has been only superficially studied. This research focuses on the development of a new prescriptive notation system for gestures that represents extended techniques requiring direct contact between the performer and the vibrating body. Such a system has been implemented in the composition *Portale*, for small tam-tam, AR environment and live-electronics.

1. INTRODUCTION AND BACKGROUND

The notation of gesture for music scores has been at the center of numerous experimentations at least in the past 60 years. The need to expand the Common Music Notation (CMN) system derived from an enlarged aesthetic panorama which, in many circumstances, was taking into account sounds and performing techniques that had not been considered and included in the standard practices of music. The notation of gesture is often connected to the notion of prescriptive notation (indication of the action) as opposed to descriptive notation (description of the result).

1.1 Action scores

In Helmut Lachenmann's action scores, "...the score is notated as a series of actions without determining their precise pitch content or even their precise sounding result" [1]. In other words, it implies a notation of gesture which leaves room for some unpredictability in the sonic outcome. The panorama is extremely vast and varied and it is impossible to provide a satisfying background in this paper. Three examples from three different authors will be provided.

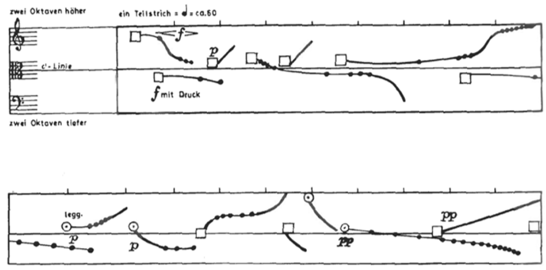


Figure 1: First two lines of Lachenmann's *Guero*

Figure 1 shows the beginning of H. Lachenmann's *Guero* (1969), where the pianist plays the keys themselves (as the emitter of the sound), with no action on the strings. The clefs report the whole keyboard extension (almost as a simplified geography of the instrument) and the score is based on gestural information (e.g., white squares correspond to the white keys played with finger's nails) and relatively free proportional time indications. Some degree of unpredictability is intrinsic in the notational system itself and is regarded as an aesthetic value of the compositional thinking.

A further evolution of gesture-based notation, almost completely excluding CMN elements, can be found, for example, in compositions by Aaron Cassidy, such as the *Second String Quartet* (2010), where notes are completely replaced by lines indicating the left hand's fingers' positions and indications such as trill and vibrato. Moreover, a grey scale is used to deliver pressure information (from full pressure to harmonics pressure). A red line indicates bow position, pressure (using transparency) and strings of contact. A green line indicates the bowing (which portion of the bow is being pressed on the string at a given time). Rhythm of left and right movements is notated on additional staves. The result is a complex and multilayered tablature that "examines the ways in which limited collections of physical action types can 'push against' constructed, dynamic, multi-planar bounding windows". The complexity of their combination "encourages unusual, unexpected, and often unpredictable materials to emerge" [2].

Pierluigi Billone's *Mani.Mono* (2007) (Figure 2) includes drawings of the gestures to perform along with the symbolic prescriptive notation of the gesture and the corresponding descriptive notation of the intended sound result. In this case, gesture-based notation is oriented towards the sound in a very specific and "deterministic" way: there is one expected sound result which is meant to be consistent across different performances. The score indicates how to

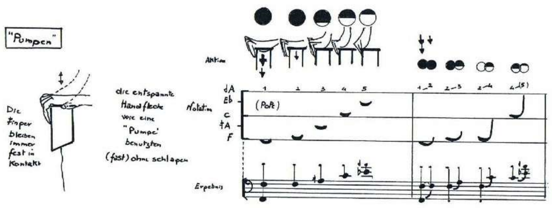


Figure 2: The beginning of Billone's *Mani.Mono*

reach a specific result rather than constructing a set of possibilities through a parametric representation of gesture.

In all the mentioned examples, the notation is coherent with the aesthetic purposes of the composers and is adequate to meet the artistic goals. This research is not meant to point out limits of 2D notation in general. However, new perspectives beyond the already known uses of notation can be pointed out: notation on paper would always show limits in carefully indicating spatial information, as it requires some form of abstraction from the space itself. In other words, notation on paper can't notate spatial events exactly where they are supposed to happen, with their
4-dimensional behaviour. Conversely, AR notation has this capability. As discussed in sections 2 and 4, in cases where an articulated use of gesture needs to be connected with a precise and consistent sound result (with the intention to minimize the intrinsic unpredictability of prescriptive notation), AR can deliver an unprecedented level of precision.

1.2 AR for music education and performance

Before describing the content of the AR-based notation system developed by the author, it is worth pointing out some background research in Augmented Reality for music education, composition and/or performance.

In music instruction, differentiated areas of interest have been taken into consideration. Interest have arisen around Western instruments, such as guitar [3, 4, 5], violin [6, 7], and non-Western instruments such as Guqin [8], Koto [9], and Dombyra (Kazhakh traditional instrument) [10]. The most researched instrument is piano, addressed in a large number of papers (e.g., [11, 12, 13] to cite a few). The piano roll is a quite standard tool: virtual blocks appear in correspondence of the keys to press. As long as a block is visible, the corresponding key has to be held down. When the block disappears, the key has to be released.

According to evaluation studies carried in the cited research papers, AR lowers the barrier of entry for beginners, guarantees higher accuracy and better mnemonic retention. The target audience of applications are students in very early stages and the proposed repertoire includes compositions studied in the first one or two years of learning. Although benefits for beginners have been proven, experts tend to find AR notation systems on traditional repertoire confusing and impractical.

It is important to notice that all the mentioned applications have been developed for allowing students to learn an already existing repertoire, as an aid to traditional instructional means more than as a stand-alone solution.

With the release of devices dedicated to AR, some experiments have risen. In [14], Kim-Boyle presents the concept of immersive score. The score *5x3x3*, previously realized in 3D, is ported into an AR environment and in room-scale size. The performer, wearing the *HoloLens 1* [1] headset, is immersed in a virtual structure (the score), superimposed to the real world and can navigate it. In this context, AR is used mainly in its visual capacity; however, different properties of the score react to timbral nuances, thus introducing interactive functions in the AR notation.

On the other hand, the works of Amy Brandon, such as *Augmented Percussion* [15], present a deeper focus on AR interfaces and interaction: bare hands of a percussion performer are used to activate/manipulate the sound of virtual objects, some of which are embedded in the real instrument. The interpreter makes use of a *Meta 2* [2] headset.

LINEAR [16] is an AR framework for improvisation that allows the creation of a notation-interface hybrid in real-time: a performer, using an iPhone, can generate virtual bodies along the trajectory of his/her gestures. Those bodies have both the function of interface (they are linked to specific samples) and notation (for generating the sound of the original gesture, the performer needs to repeat the same gesture).

[17] describes an AR controller for sound spatialization in n-channels designed both for diffusion automation and for live performance. It allows to position virtual bodies in the space and to draw trajectories with the gesture. Each of these bodies is linked to a sound source moving in the space along the designed trajectories. The tool is designed for HTC Vive Pro [3] .

2. THE AR ACTION SCORE: WHY AND HOW

The concept of AR action score will be explained by analyzing the particular case of the author's *Portale*(2019), for small tam-tam, AR environment and live-electronics.

In the composition, the tam-tam is played in 2 ways:

- by using fingers wearing thimbles;
- by moving one magnet (held in position by another magnet in the back) over the surface of the instrument.

The finger generates a scratchy timbre when it is moved continuously on the surface. It also produces a more "pitched" sound when hitting the tam. The magnets move the perceived pitch and shift spectral components, behaving as masses attached to the vibrating body. The will to realize a clear and intuitive notation for indicating the movement of the magnets (addressed to a specific spectral configuration) is at the core of the development process.

[1] An AR see-through device developed by Microsoft.
[2] An AR see-through device developed by MetaVision.
[3] Non-see-through head-mounted display headset allowing AR content thanks to the use of front-facing camera for visualizing the real environment

2.1 How the magnets modify the tam's spectrum

Figures 3a and 3b report two different results (spectrogram and pitches of the loudest partials) generated by the percussion of a finger (with thimble) hitting one specific tam-tam in a specific position (white cross) with the magnet(s) in another specific position (white circle).

Although some difference has to be expected among repeated hits, the spectral content of sounds is consistent for every position of the magnet and of the hitting finger, given a specific tam-tam, a specific magnet and a specific thimble. Figure 4 shows the spectral shift obtained with a magnet moved on the tam-tam. Every trajectory performed on the surface of the instrument results in a different shifting effect (although with some similarities between similar movement shapes).

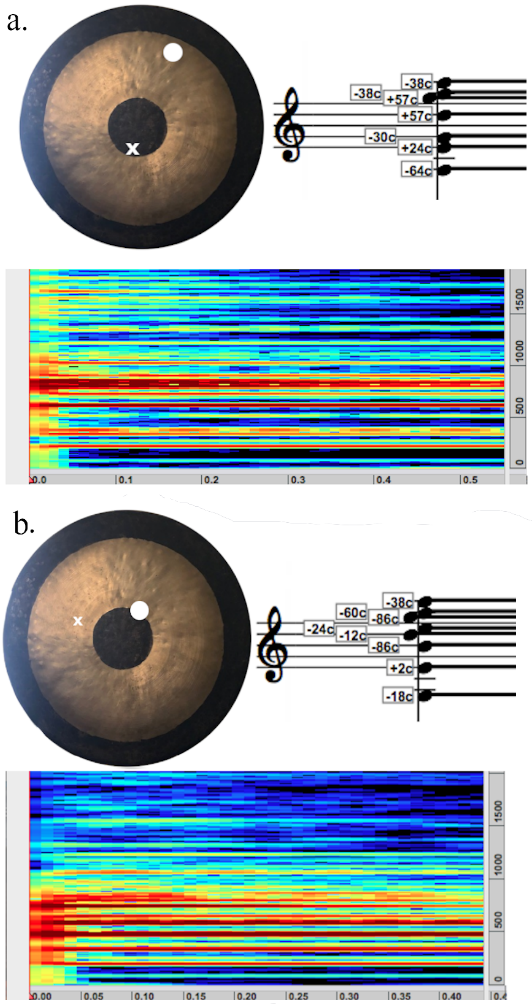


Figure 3: Analysis of the excitation of the tam by a metal thimble at the location indicated by the white X with the magnet at the location of the white circle. 3a and 3b indicate two different positions both for the thimble and for the magnet. Spectrogram shows the first 0.6s of the attack, from 0-2kHz. The seven highest amplitude frequencies are showed in common notation, with cent deviations.

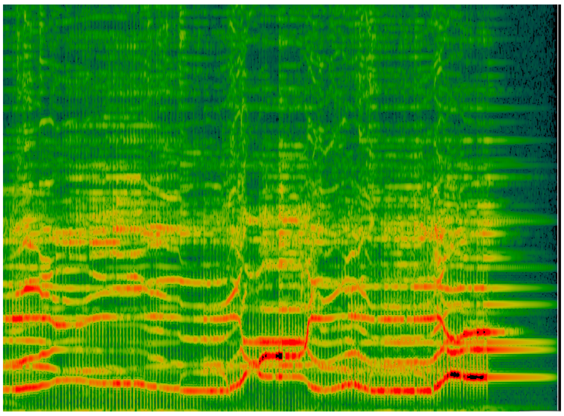


Figure 4: Spectrogram displaying the shift of partials during the movement of the magnet. When the magnets stop moving the partials become visibly more stable in pitch distribution.

2.2 Notate the effect by notating the action

In general, the accurate identification of sound results in the context of extended techniques poses well-known limitations [4]: instability, unpredictability and limited control over the result (although inside a well-defined timbral world).

The rationale behind the notation system presented here is that, by being able to indicate the specific point in space where the interaction has to be realized, it is possible to increase the level of accuracy, while decreasing the complexity of the deciphering process.

In fact, if any position of the magnet and of the thimble on the instrument correspond to a quite consistent class of spectra, the identification of the precise point of access is enough for also indicating the result with a satisfying precision. In addition, the notation is almost self-sufficient, requiring few preliminary information. On the contrary, a notation on paper could need to be complemented with additional audio material and performance notes (especially in circumstances where a precise spectral result is required).

The notation system implemented consists in virtual points indicating hit spots and in virtual lines following pre-designed trajectories projected directly on the tam-tam. The performer can see those objects by wearing a headset for AR rendering.

Figure 5 shows the two playing modes on tam (with fingers and with magnets). The line in the picture on top illustrates the trajectory the performer has to follow on the tam with the index finger. Such a line is not static but moves (therefore, the direction to follow is made obvious by the direction of the line along the trajectory). The picture at the bottom presents the AR indication corresponding to the configuration presented in Figure 3a: the AR model of the magnet (black virtual sphere) indicates the intended magnet position and the light blue effect representing the hit point for the finger.

[4] In some cases, considered as aesthetic resources.

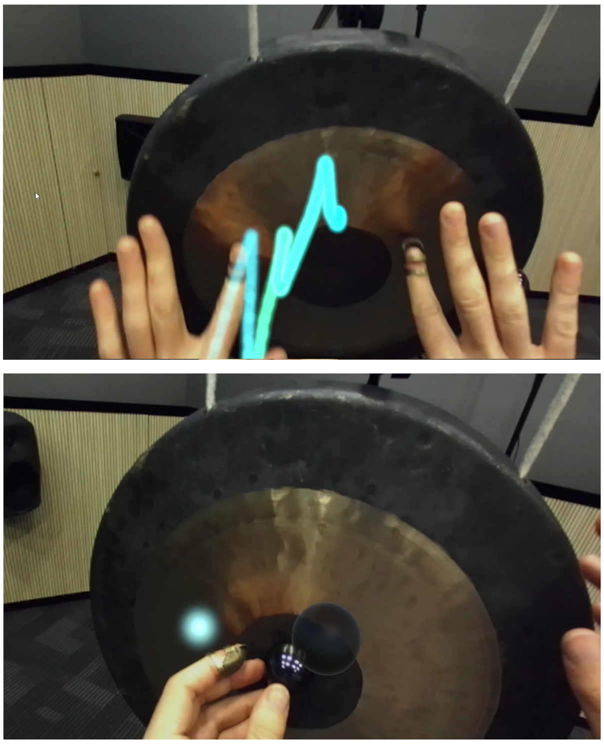


Figure 5: The two ways the performer is meant to play the tam: fingers with thimbles and magnet(s). On top, the AR line representing a trajectory the performer has to follow. At the bottom, the AR notation of the configuration in Figure 3a.

2.3 Notation of time and notation of intensity

In a prescriptive notation system as above, that consists in a line mimicking the gesture to perform, the notation of rhythm has two specific aspects.

The first one is that the notation is not symbolic (it does not use a figure that resembles something else from itself) but mimic (the movement is actually performed in advance by a virtual object and subsequently repeated by the performer). We are used to adopt some kind of symbol (or position on paper) to resemble durations (or time proportions). In general, we divide greater time values in smaller ones, or create longer durations by adding up shorter ones. In the AR notation system introduced here, time is not represented as multiples (or fractions) of a fixed amount of duration; on the contrary, it is represented as a fluid alternance of internal speed articulations of the gesture. We could call it *continuous rhythm* as opposed to *discrete rhythm* (the rhythm expressed by metrical values). Such fluid notation of time can take into account the differences of speed inside a gesture: in fact, a real performance act can have an unstable velocity, connected to differences in sound result (e.g., different speeds of the bow on a string or, as in Portale, different speeds of a magnet on a metal surface). Although the metric notation in CMN can become extremely specific in defining different durations, it will always describe the articulation of a gesture as a sequence of finite durations (for going from a point A to a point B) and not as a fluid change in the gesture's velocity.

The second aspect is related to the performer's reaction time. In fact, rehearsed rhythm learned on a score is precise. On the contrary, in this system the information is conveyed to the interpreter in real-time and every action is notated in the moment in which it should happen. Clearly, the performer cannot realize the required actions as soon as they appear on the instrument. The execution needs some delay time. For this reason, a certain fuzziness in the rhythmical outcome is an intrinsic component of such a notation system.

At the current stage there is no clear indication of dynamics (which has a noticeable impact on the spectral result); adding some indicators for that parameter is a forecasted enhancement of the software. However, the system already notates width and speed of movements and hits, which have a close relation with dynamics (faster and wider movement for *ff*, opposed to small and slow movements for *pp*).

3. TECHNICAL ASPECTS

The notation system used in *Portale* relies on a technical environment requiring different hardware and software components. In addition to microphones, speakers, sound interfaces and pc(s) for audio, visual and positional tracking processing, the framework requires:

- 1 Head Mounted Display (HMD) [5];
- 1 stereo VR camera [6];
- 2 motion capture trackers [7];
- a software developed in Unity 3D for AR processing.

3.1 Headset and trackers

The performer is wearing the HTC headset, which allows the representation of virtual bodies in space by detecting the real world with front-facing cameras and representing it on the internal screens (one per eye). Virtual bodies are rendered on the same screens. The front-facing cameras natively installed on the headset deliver a poor image quality (420p per eye) and have a high latency (200 ms), making it problematic for the performer to follow the notation accurately. For this reason, the *ZED Mini* Stereo VR Camera has been mounted on the headset for replacing the native one, bettering the resolution to 720p per eye and lowering the latency to 60 ms.

The two trackers are positioned:

- on the tam for detecting the position and orientation of the instrument. This way, the AR score always follows the tam's movement;
- on the right hand of the performer. This tracker is used as an input device for interacting with virtual bodies (this component of the composition is not a part of the notation system described in this paper and therefore its function will not be further addressed).

[5] *HTC Vive Pro* headset.
[6] *ZED Mini* VR Camera.
[7] *Vive Trackers*, devices used for detecting the position and orientation in space of objects in the real world.

3.2 Software

The software side of *Portale* is articulated in two components:

- an AR program created and compiled in Unity;
- a Max/MSP project (its functioning is not further analyzed as it is not implied in the notation system).

The AR software is responsible for scheduling, rendering and positional tracking processing. The software also sends control information to Max/MSP when the interaction of the physical performer with virtual objects (detected through motion tracking) is meant to produce some sound outcome (generated in Max/MSP).

3.2.1 Gesture design

In the application, gestures are resembled by a virtual object (the blue line in Figure 5) following a trajectory with a certain speed.

The creation of an Augmented Reality score poses two issues: how to draw a 3D gesture and how to move a body along that trajectory in time.

The trajectory of the gesture is created with a 3D cubic Bezier's curve (a parametric curve used in computer graphics), whose shape and bending can be adjusted by shifting the position in space of nodes (the white squares in Figure 6) and control points (the red squares in Figure 6, two per node). Trajectories are designed in advance and cannot be changed during the performance.

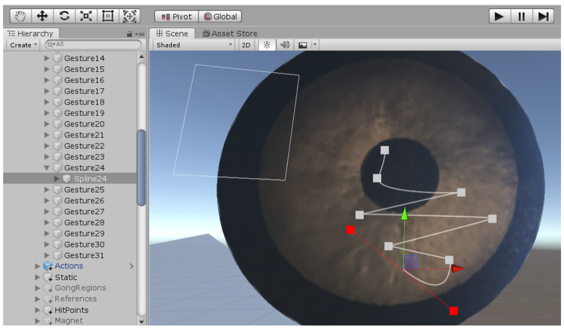


Figure 6: The creation of one trajectory with a sequence of cubic Bezier's curves inside the Unity editor.

Two *scripts* [8] allow the composer to control the starting point in time, the internal speed articulation of the gesture (how long does it take for the line to go from one node to the next) and its total duration.

4. DISCUSSION

The concept of AR action score and AR gesture-based notation is conceived in the frame of extended techniques and timbral research. Its main focus lies in the delivery of mimic (not symbolic) prescriptive information in 3D space which, while ensuring a certain degree of intuitiveness, allows an accurate control over the result. In fact, in every circumstance where the result is consistent given a specific position of exciters and/or preparation [9], the system provides a univocal way of notating that result (as a function of the positions or gestural behavior). The particular nature of time indication in this context produces a condition which could be defined as continuous rhythm, as a consequence of its capability of notating speed changes in gesture instead of events happening in relation to fixed rhythmical values or in relation to time proportions linked to the position on paper.

In *Portale*, the formal development is focused on the evolution of different forms of interaction between the physical performer and a virtual object (the blue animated line that runs across trajectories) which, in the last section, becomes AR gesture-based notation. Describing the other stages of interaction would go beyond the scope of this paper. However, it is worth mentioning that, in its implementation for *Portale*, the notation system has been used for a restricted set of pre-designed actions with a relatively narrow space for development: only two playing techniques, only one trajectory at a time, the impossibility for the physical performer to interact with the notation itself (but only follow it). These constraints have been implemented mainly for two reasons:

- *Portale* is structured around different possibilities of interaction between the physical performer and virtual objects (and viceversa) and, in this context, AR notation is one of them;
- the physical conformation of the small tam did not allow to safely and/or effectively use some solutions [10] .

These constraints are not to be considered limitations of the system itself, but rather choices of implementation. Other versions of the system for future compositions focusing exclusively on AR gesture-based notation will include broader sets of possibilities.

Although, in the opinion of the author, such a quite unprecedented possibility shows potentials for future musical research, there are some intrinsic limitations.

While the system can be considered precise in static situations, when movements (especially for the magnets) are considered, the instrument itself and the magnets attached to it oppose some resistance to the performance gestures. In fact, the irregularities of the surface of the tam might sometimes prevent the magnet from keeping its position or following the desired path. As a result, a precise indication of movement does not automatically translate into a precise movement.

Another issue consists in the difficulty of adaptability of the notation. For example, playing *Portale* on a tam different from the one on which the composition has been developed, would make the sound result (at least slightly) different. In fact, every tam, even if of the same size and

[8] A script is a custom programming file written in C# that can be attached to virtual objects in order to control their behavior.

[9] Preparation is here intended as a modification of the usual behavior and timbral result of a vibrating body by the addition of extraneous masses to the vibrating body itself.

[10] E.g., the use of two sets of magnets could have created problems given the small size of the instrument. In fact, there is a high chance of having the two sets near enough to generate magnetic attraction, thus making impossible to have the control over the required techniques

from the same constructor, could present significant timbral divergences. Those dissimilarities would be particularly obvious in case of tams with different sizes. This limitation would also hold for other instruments on which this notational system might be applied.

All of this said, the notation system itself would not present particular problems of scalability in size (e.g., changing from an 18" to a 32" tam): the same score would easily fit a bigger or smaller instrument. That is because the reference system can be scaled inside the AR software, and therefore the score itself and all the gestures can be instantly resized accordingly.

One additional constraint is constituted by the organological nature of the instruments on which the system could be used. It would only work for instruments whose vibrating part can be directly manipulated by the performer with the gesture (surface-like instruments): most of percussion instruments and strings; conversely, it could not fit for woodwinds and brasses (where the vibrating body is constituted by the air column).

An informal evaluation carried on December 1st 2018 at the *AR/VR Retreat* in Berkeley (CA) showed that non-musicians were able to perform a 30 seconds AR score producing a result reasonably close to the intended outcome after just one repetition after an instruction process that lasted around one minute. Although only a structured evaluation (which will be presented in future research) could provide trustable results, the informal one was encouraging.

5. CONCLUSIONS

The notation system presented in this article, developed for the composition *Portale*, allows one to draw in space and time the intended gesture addressed to a specific sound result. The score is formed by a series of static points (indicating specific spots on the instrument) and lines moving along pre-designed trajectories rendered on the tam. This notational system guarantees to preserve immediateness and intuitiveness of notation, while being accurate on the result. In fact, as shown in Session 2, it is possible to compose the harmony derived from the movement of the magnet across the surface of the tam without needing extra sound material or additional spectral/harmonic information. The indication of the position of interaction or of the trajectory is sufficiently accurate to determine quite consistently the result.

Another point of interest can be found in the notation of rhythm, not realized through symbols referring to discrete values (such as quaver), or through a proportional graphic distribution, but coming from the internal speed articulation of the gesture. AR notation, as implemented in *Portale*, mimics the behavior of gesture over time and represents visually and in real-time the fluid alternation of velocities with a level of similarity that notation on paper could not possibly achieve. Such a particular dimension of temporal indication might be referred to as *continuous rhythm*.

The system requires the use of a specific headset for AR rendering (HTC Vive) and is realized through a custom software created in Unity 3D. The trajectories used for the score can be generated with a graphic editor and custom scripts allow to compose the internal speed of movement on each trajectory. Trajectories are then automatically placed on the real tam through the use of the Vive Tracker.

Main limitations consist in difficulties of adaptability (different tams provide a slightly to greatly different sound result given the same interaction positions and trajectories). Additionally, the notation is fruitful only on instruments providing a surface for interaction (while it could not work on instruments using the air column).

In future work, the realization of a formal evaluation experiment will provide more information on the actual usability, precision and intuitiveness of the system.

6. REFERENCES

[1] A. Heathcote, "Liberating sounds: phylosophical perspectives on the music and writings of Helmut Lachenmann," Ph.D. dissertation, Durham University, 2003.

[2] A. Cassidy, "Constraint Schemata, Multi-axis Movement Modeling, and Unified, Multi-parametric Notation for Strings and Voices," *Journal for new music and culture*, 2013.

[3] C. Kerdvibulvech and H. Saito, "Vision-Based Guitarist Fingering Tracking Using a Bayesian Classifier and Particle Filters," in *PSIVT'07 Proceedings of the 2nd Pacific Rim conference on Advances in image and video technology*, vol. 14, 2007, pp. 625–638.

[4] M. Löchtefeld, S. Gehring, R. Jung, and A. Krüger, "guitAR: supporting guitar learning through mobile projection," in *CHI 2011*, 2011.

[5] J. R. Keebler, T. J. Wiltshire, D. C. Smith, and S. M. Fiore, "Picking up STEAM: Educational implications for teaching with an augmented reality guitar learning system," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013.

[6] F. D. Sorbier, H. Shiino, and H. Saito, "Violin Pedagogy for Finger and Bow Placement using Augmented Reality," in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2012.

[7] H. Shiino, F. De Sorbier, and H. Saito, "Towards an augmented reality system for violin learning support," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013.

[8] Y. Zhang, S. Liu, L. Tao, C. Yu, Y. Shi, and Y. Xu, "ChinAR: Facilitating Chinese Guqin learning through interactive projected augmentation," in *ACM International Conference Proceeding Series*, 2015.

[9] M. Doi and H. Miyashita, “Koto learning support method considering articulations,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[10] M. K. Zhaparov and U. Assanov, “Augmented reality based on Kazakh instrument ”Dombyra”,” in *8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014 - Conference Proceedings*, 2014.

[11] M. Weing, A. Röhlig, K. Rogers, J. Gugenheimer, F. Schaub, B. Könings, E. Rukzio, and M. Weber, “P.I.A.N.O.: Enhancing Instrument Learning via Interactive Projected Augmentation,” in *2013 ACM Conference on Ubiquitous Computing, UbiComp 2013*, 2013, pp. 75–78.

[12] X. Xiao, B. Tome, and H. Ishii, “Andante: Walking Figures on the Piano Keyboard to Visualize Musical Motion,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2014)*, 2014, pp. 629–932.

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

[14] D. Kim-Boyle and B. Carey, “3D scores on the HoloLens,” in *TENOR 2019 International Conference on Technologies for Musical Notation and Representation*, Melbourne, 2019.

[15] A. Brandon, “Augmented Percussion,” 2019.

[16] G. Santini, “LINEAR - Live-generated Interface and Notation Environment in Augmented Reality,” in *TENOR 2018 International Conference on Technologies for Musical Notation and Representation*, Montréal, 2018, pp. 33–42.

[17] ——, “Composing space in the space: an Augmented and Virtual Reality sound spatialization system,” in *Sound and Music Computing 2019*, Malaga, 2019, pp. 229–233.

SYMBOLISING SPACE: FROM NOTATION TO MOVEMENT INTERACTION

Katerina El Raheb, Marina Stergiou, Akrivi Katifori,Yannis Ioannidis
National and Kapodistrian University of Athens
"Athena" Research Center
kelraheb@di.uoa.gr

ABSTRACT

The last decades the development of whole-body interaction technologies, as well as XR (Extended Reality) technologies, including Augmented, Mixed and Virtual Reality, created a strong potential for embodied and immersive experiences to support learning and the use of notation while moving. In our ongoing work, we explore this potential on the user case of familiarizing dance experts and amateurs with movement notation in general and Labanotation in particular. By applying methodologies of user-centered design, including co-design workshops with notation and dance experts, interviews, focus-groups, questionnaires supporting the iterative design of our prototype, we focus on how we can meaningfully transfer the concepts related to space from notation to full body interaction instructions. So far we have developed two prototypes following two paradigms: a. the augmented mirror paradigm using Kinect and b. the immersive paradigm using HTC-Vive, that we have used as technology probes to interact with dance experts in the context of our co-design work. We reflect on this experimentation and we document the emerging challenges of transferring a symbolic language that is meant to be transmitted through paper, into spatial semantic queues. We discuss the challenges that arise between the gaps of symbolically referring to space, within a rich conceptual framework, such as Laban Movement Analysis (LMA), experiencing space kinaesthetically, and transferring these into a digital experience, always within the limitations of the current technologies.

1. INTRODUCTION

While traditional notation of music is an integral part of music education, at least for classical and western genres, using notation for movement and dance practice is rather an exceptional case. Notation and the creation of scores in dance education, creation and practice is a rather rare and idiosyncratic process, unless we are talking about systematic choreological analysis with the participation of dance researchers and trained notation experts. While more than 80 notation languages are mentioned in literature, still Labanotation, remains the most well-established system for writing and analysing movement. Although it does not represent an everyday language for dance educators and practitioners of any genre, it is still a powerful tool for movement scholars. In addition, the last decades many researchers and developers in movement computing as well as Human Computer Interaction have found in Labanotation and Laban Movement Analysis, a powerful tool for conceptualising human movement [1].

Playful technologies show great potential for making learning experiences much more interesting and fun for both adults and digital native young students, through embodied experiences [2], especially in the case of teaching more complex, analytical knowledge, such as dance notation. Previous works have discussed the opportunity of cultivating kinesthetic awareness, i.e. "the perception of our position and movement in space" through interaction [3, 4], using audio feedback.

So the question is, what are the implications of exploring symbolisation of directions in space using notation within a three dimensional XR experience? We argue that moving in space to familiarise oneself with concepts about space on a cognitive level can make the whole process more enjoyable and intuitive than studying on paper. On the other hand, while current XR technologies offer a great opportunity to learn or read notation while moving, there are many implications when it comes to cultivating spatial awareness.These implications originate both from the limitations of current technologies (e.g., limited precision, visual feedback on flat screen or small field of view), and from the complexity and diversity of the notion of Space, both on a cognitive and embodied level.

In this experimental, qualitative study, we focus on the design challenges that emerged during our iterative, participatory approach of communicating and capturing simple directional concepts through the Laban symbols, using two XR application paradigms. As we have observed, many of the usability and user experience challenges emerge from the complexity and expandability of embodied perception of space in movement practice vs. the strict, geometrical representation of it in the digital environment. Following a research through design logic ([5], we have proposed an experimental whole-body interaction application that evolved through-out the process. The objective of the application is to teach the basic Laban direction symbols through a playful embodied experience, implementing two paradigms that are inspired by dance practice. The first one is the mirror paradigm, implemented in our case with the use of a Kinect motion sensor device and SDK [6] and the second one is the immersion paradigm, implemented using virtual reality equipment, HTC Vive [7]. Both prototypes have been developed using the Unity 3D platform [8]. Section 2 provides an introduction to the concept of space in dance practice, Laban Movement Analysis and Notation and the idea of the imaginary cube as an extension of the personal space to practice directions. It is important to note that unlike music, practicing dance while reading and taking notation is not a common practice. In Section 3 we relate our work with previous research and efforts to teach dance or notation using XR technologies or the paradigm of augmented mirror. In Section 4 we describe the scenario of use and the setting of the installation, while in section 5 we explain our methodology during the process of design and provide details about engaging with the dance and user community. In Section 6 we present the findings of our research and in section 8 we conclude the work.

2. SPACE AND DANCE PRACTICE

The concepts of space and its perception is prominent in contemporary dance and other movement practices and the awareness of orientation of the whole body and limbs is a skill relevant to most dance practices. Its understanding, but also re-thinking, re-constructing and developing ideas around it and its metaphoric and poetic nature is in the core of both choreography and learning. Therefore the definition of space in movement practices extends the definition of the measurable Cartesian space [9].

2.1 Labanotation: a symbolic language for Movement

Based on Laban Movement Analysis (LMA), the theoretical framework to analyzing movement, Rudolf Laban has introduced Labanotation, the symbolic language for notating different aspects of movement, such as directions, body parts, dynamics, timing and others. It is mostly used for dance and motion activities and translates motion into symbols, where change of symbols constitutes motion [10]. Labanotation, apart from being the most wide-spread system for notating dance, is also a valuable medium for the cognitive representation of the structure of movement, especially at the beginning stages of movement learning [11, 2]. The last decades, several digital applications have been proposed for choreography and dance documentation and automatic analysis [12, 13, 14, 15]. The potential of LMA and Labanotation, has also been explored in other domains such as the design of movement-based interaction [16], design of expressive animation characters [17], gestural design [18], and artistic installations [19].

2.2 Laban Movement Analysis and Space

Laban Movement Analysis [20, 21, 22], consists of four basic elements that concern various movement aspects: *Body* (what), *Space* (where), *Effort* (how), and *Shape* (in relation to what). *Space* is further categorized in personal,

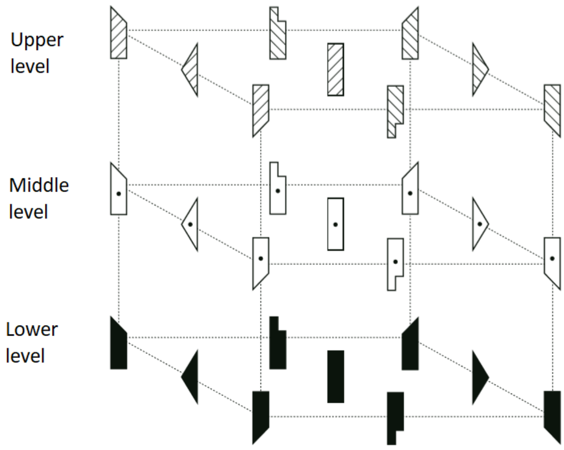


Figure 1. Laban's cube with directional symbols on the corresponding points in 3-dimensional space

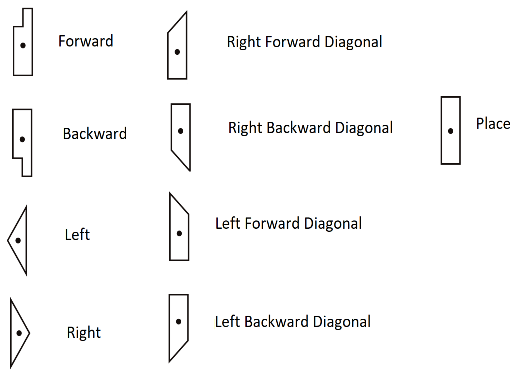


Figure 2. Direction symbols of Labanotation

interpersonal and general space. Personal space, the volume created and occupied by each person or *kinesphere* is defined by Laban as "the sphere around the body whose periphery can be reached easily with extended limbs" [23, 24]. The center of this volume is the center of the mover's body. The kinesphere, the imaginary sphere around the body, can expand or shrink according to the mover's will and mood.[17]. The use of space and the relation to the kinesphere is one of the major aspects in LMA that can apply to the analysis of both functional as well as expressive aspects of movement. Further division and transformation of the imaginary kinesphere into geometrical shapes and relationships in LMA are used both as educational tools to cultivate spatial awareness, as well as for analysing and "reading" existing dance works and performances [25].

Inside the kinesphere the mover's body can create various formations which can be seen as polyhedrons. These polyhedrons are explored through the *movement scales* which are pathways that connect specific points of interest of the polyhedrons (planes, edges, corners, diagonals etc.) and can be outlined with each body part. Laban connected specific polyhedrons of interest with elements like dimensions, planes and diagonals. The most basic polyhedron is the cube (see Fig.1) and is correlated with the concept of diagonals. The cube can be furthered divided in three levels, upper, middle, lower; each level has nine points of

Figure 3. Recordings (video and Optical Motion Capture) of Directionality exercises in the context of the WhoLoDancE project

interest, eight in the periphery and one in the center(see Fig. 1 and Fig. 2).

2.3 Moving in a Virtual Cube

Some dance practices highly rely on the use of the frontal point-of reference usually referred as the "audience" and a mirror is used both for helping young students to get oriented in the space and also to check the correctness of the posture or movement. Practicing Laban scales gives practitioners and dancers the chance to enhance their cognition and perspective of moving in 3-dimensional space. Moreover, they can experience the coherence of kinesphere spatial structure. Apart from the scales, the Laban cube offers the possibility to the mover practitioners and dancers to practice various directionality concepts, enhancing in that way their perspective of the 3-dimensional space. In addition, besides LMA, the concept of the cube, as a starting point to create movement forms in choreography, has been used by other choreographers such as Trisha Brown, which she has sketched as the imaginary cube and its points in her work "Locus" [26]. The choreographer William Forsythe in "improvisation technologies" [27] provides several examples for geometries in space and how they can be used in spatial thinking in a creative context.

In the WhoLoDancE project [28] several exercises, using the concept of the imaginary cube have been proposed by the contemporary dance experts to cultivate the sense of space and self-practice directionality, such as following specific sequences of pointing directions by particular body parts, as shown in Figure 3. In particular, during the motion capture of those exercises, which aimed to create a movement library with useful educational content, a metallic physical cube was set in the studio to make more concrete the idea of orienting body parts towards points in a visible, tangible cube.

3. TEACHING SPACE IN A VIRTUAL ENVIRONMENT

It is true that some dance practices heavily rely on the use of the mirror both for supporting orientation and for self-correcting posture and motion. In these cases, the metaphor of the "augmented-mirror" and the use of Kinect [29, 30] have shown positive results regarding both usability and effectiveness in self-practice [31, 32, 33]. Mirror, however, as well as, video and Kinect, unless combined with more than one devices, provide only one 2-dimensional perspective of the body and movement. In this case, a combination of Kinect with extended reality equipment might be a solution [34]. Familiarizing someone with Labanotation concepts and symbols of space includes the understanding of high level space concepts and the architecture of the body. It is definitely an important aspect in dance teaching that extends to other domains such as movement literature, bodily knowledge, but also analytical skills regarding space, that is useful for both children and adults, and non-dancers. In movement practices, on one hand there is the improvement of the kinesthetic awareness, on the other hand there are the cognitive skills such as memorization, and analytical conceptualization on space, aspects that can extend to other domains such as interdisciplinary thinking, geometry, architecture etc. Recent advances in technology for gaming and motion tracking, as well as implementing Extended Reality (XR) environments have the potential to create effective training environments and compelling entertainment experiences. It is commonly accepted, that learning Labanotation can be hard and frustrating not only due to its novel vocabulary but also because Labanotation tries to describe motion in 3-dimensional space with 2-dimension symbols. Ballas et al. [35] describe a Kinect-based system for teaching Labanotation in mixed reality. The proposed way of teaching the symbols is by following an avatar that performs according to the desirable Laban score. In that way, the user is intended to learn the symbol by seeing and mimicking a mirrored, avatar teacher. In addition, mobiles apps have been developed either to create a score [36], to visualize and explain the Laban scales [37] , and to read notation while moving using Augmented Reality (AR) glasses [38].

4. SCENARIO AND GENERAL IDEA

The cube is one of the simplest polyhedrons of Labanotation to explore three-dimensional space and conceptualize about levels and directions. The exercises proposed by the

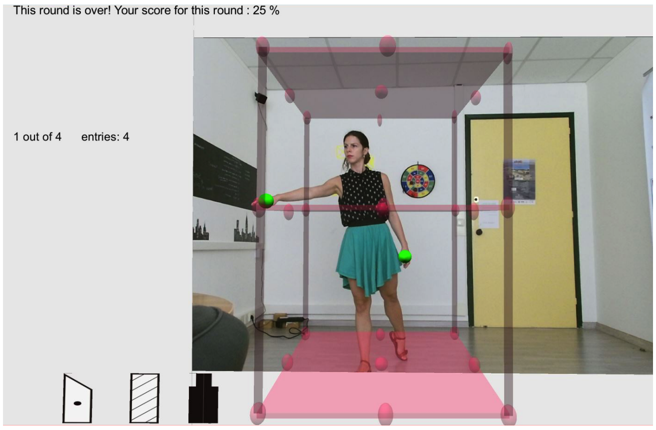


Figure 4. Cube exercise in Augmented Mirror setting using Kinect

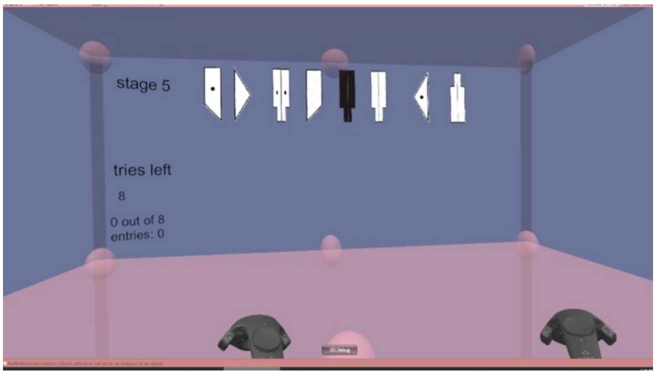


Figure 5. Screenshot from the first person perspective in VR.

dance experts for directionality with the cube, are considered as very simple, basic, generic exercises which can escalate into more complex combinations depending on the level of the performer. However, with the fact that Kinect is based on a depth camera, providing visual feedback on a two-dimensional screen is a challenge.

4.1 The cube in a VR setting

Taking into account the experience with the Kinect and its specific characteristics and limitations, we decided to migrate and test the whole scenario in a virtual reality setting, using virtual reality equipment, to experiment with a more immersive experience. The application was adapted to be used with the HTC Vive VR equipment, including a VR headset attached to a cable and two hand held controllers which ensure that the Vive sensors record the position of the users' hands . The use of immersive VR was greeted with excitement by the collaborating dance and LMA experts who explored its possibilities in comparison with the experience with the Kinect (Figure 4).

However it was immediately clear that the new setting brought also a new set of challenges. As it is to be expected, this type of immersive VR required a cumbersome headset and holding the controls, which seemed to the experts as a step backwards from the unencumbered use with the Kinect. Although we had foreseen that in the Vive setting it would be much easier for the users to complete the exercises by reaching the correct points on the cube, due to the higher accuracy of the motion capture, there were still difficulties. We suspect that this was due to the fact that the cube was placed in an otherwise empty virtual environment which might have been a bit disorienting for the users, as P5 suggested. On the whole, they were very enthusiastic with the tool so maybe this was the reason they overlooked its limitations. The next section will attempt a further reflection on the use of the Vive comparatively with the Kinect version.

4.2 Setting and workflow

The scenario of the exercise is as follows: In the Augmented Mirror version the users see on screen their video camera captured self within a virtual cube, as it is shown in Figure 4. In the VR (Virtual Reality) version the users are immersed in the virtual cube. In both versions, the users are asked, by us orally, to point with their hand to the direction that is given, having the directional symbol, as a "semantic aid" [39] to reach this direction. The direction is the edge of a cube which is virtually attached to their personal space. There are three phases in this task: the a) Exploration, b) the One-to-one, and c) the Memorization.

In the exploration mode the users are asked to point to the directions and the symbols appear in order to let them explore and familiarize. When their hand enters the area that represents a specific direction in the cube-space, the symbol of this direction will appear in that place allowing them to observe it as long as their hands stays in that area. This stage is called "exploration" mode, since the users are free to go ahead and discover all the directions, as many times as they want. Therefore, they get informed about the directions that they are following as well as about which symbol represents each direction.

In the one-to-one phase of the exercise, the users see the symbol and they are asked to "reach for" the virtual ball attached on the corresponding direction and edge of the cube. If they succeed, the system displays the next symbol. A "win" sound and the temporary color change of the cube from purple to green are used as corrective feedback to the user, apart from the symbol change, every time they succeed. A score is displayed at the end of this stage to inform the user about their performance. The type of the score given, is based on how many "hits" they had in the total amount of the symbols that were displayed. The number of the symbols that the users have to go through in this stage is flexible and it depends on the desired difficulty, the available time schedule etc. An indicative number would be all twenty-seven symbols, therefore all symbols, one time each.

After trying the previous stage, during the memorization mode, the users are now ready to try what they have learned so far by performing a sequence of moves in order to go through the combination of symbols-directions that appear on the screen. The first sequence consists of four symbols, the second of eight and the third, of twelve. Only if the users succeed in finding all symbols of a group, they can move to the next one. If they fail in finding one symbol, they have to go through the whole sequence again. When they go to the right direction that the symbol indicates, a win sound is played, the cube changes color from purple to green for two seconds and the symbol displayed is replaced with a grayed-out one to indicate that the users can move to the next symbol. In this stage, the user can see live information of how many symbols they have found correctly out of the total, for each group (four, eight and twelve).

5. METHODOLOGY

Our approach is based on of a longitudinal co-design collaboration with four dance and LMA experts that lasted a year and consisted of a series of co-design sessions. This long term collaboration was combined with once-off hands-on demonstrations to a varied user group, involving children and adults, dance and movement experts and technology experts, in a total of more than 30 users in a series of different events. Through this iterative design approach, we aimed to document and discuss the potential and the challenges in conveying the spatial concepts through the proposed kinesthetic experience.

5.1 Working with dance experts

Our co-design group involved five dance and LMA experts:

- P1: semi-professional dancer and teacher, having 25 years of dance practice in ballet, contemporary and other types of dance as well as theoretical Labanotation knowledge,
- P2: a dance practitioner, contemporary and ballet dancer, with 8 years of experience and basic knowledge of Labanotation,
- P3: a dance high education professor and researcher, expert in LMA and Labanotation as notator,
- P4: a renowned contemporary choreographer, with more than 25 years in making and teaching movement for the stage to children, adults, actors and dancers, using LMA for educational and creative purposes,
- P5: young dance professional teacher and choreographer, expert in community dance, having degrees in dance and Psychology.

The co-design sessions with the experts included several three to five hour sessions which alternated theoretical discussions and bodystorming [40] on the concept and method, initially, and later, as the design and implementation progressed, hands-on evaluation of the prototype. The experimentation with the tool was at points guided through specific tasks, at others more free-form so as to explore its different perspectives and identify challenges.

5.2 Involving other user groups

In order to test the prototype and concept with a more varied user group with or without any background in dance

Figure 6. Choreographer (P3) and Expert Labanotator, participant (P4) using Vive during the co-design session

or LMA, and/or with or without experience with Kinect-based applications in games, we organized hands-on demonstrations of the tool in the lab and also in the context of different conferences and events. We involved five users who were both dance and movement experts, five technology experts and seven people of the wider public, with special focus to children and teenagers as digital natives. More specifically, we involved 15 young digital natives, aged 8-15 years old, with some of them having experience in ballet, but none in LMA or Labanotation.

The installation was offered as a hands-on demonstration. In all sessions an introductory stage with a presentation preceded the main tasks to explain the general context and objective of the study and to give to the participants a short introduction to Laban s cube and Labanotation. Feedback was collected while observing the users with the tool and also, in the form of questionnaires and brief interviews, also recording their previous experience with movement practices, as well as with Kinect based games.

6. OUTCOMES AND DESIGN CHALLENGES

In this section we document design challenges that emerged during the co-design process as well as the remarks and outcomes from the evaluation with other user groups during the demo sessions.

6.1 Supporting memorization

All the dance experts during the co-design sessions, and also the adults with dance or technical background expressed their interest in the memorization tasks. After the experiment, even users who had briefly used the application and had no previous experience with LMA showed that they actually memorized a good number, and in some cases all, of the symbols. For the young digital natives, the tasks were definitely the most clear and fun phase according to their interview answers and our observations.

6.2 The need to encourage 3-dimensional movement

Although the exercise, and the idea of exploring the cube, is designed to encourage the three-dimensional use of the body, this was not fully accomplished with the application. This was due to the use of the 2D screen combined with the fact that most people, especially those having experience with this type of device were expecting to have a more static, upper body, gestural interaction with the system. Regarding the main experience, users who were not familiar with the Kinect and tended to turn and bend their body very often, faced some inconsistencies in the results of their movement. Kinect works best when users face the camera and make simple movements that don't involve much bending or turning.

6.3 User's familiarity with MS Kinect

We observed that users who were familiar with using and working with MS Kinect were significantly more successful in completing the tasks than the ones that were familiar with movement and Laban concepts but had no experience in using Kinect. This was mostly because they knew the correct way to perform certain moves like bending and turning that Kinect couldn't capture with great fidelity. Movement experts, on the other side, were more focused on the representation of their movement and seemed frustrated that Kinect didn't always respond well. Therefore, this lack of familiarity with the medium made it quite hard for the movement experts to focus on the tasks and complete them.

6.4 Self-image on screen vs. seeing your space in first person view

The objective of this installation was to teach the basic Laban direction symbols during a playful embodied experience. Through the co-design sessions with the dance experts and taking into account the outcomes of the involvement of other user groups, we reached the following realization: On one hand this type of installation is in fact effective to support memorization and learning of the symbols. But on the other, the Kinect motion sensor device in fact implements the mirror paradigm which implies that the users have to be aware of their own surrounding 3D space while at the same time focusing on a two-dimensional screen to get feedback. This constant shift of attention between the screen and the physical space is not the optimal solution for cultivating spatial awareness. As a conclusion, the MS Kinect hardware might be affordable and portable, but was not fully serving the idea of the cube as an extension to the user's own body and personal space. It is true that some dance practices heavily rely on the use of the mirror for both supporting orientation and self-correcting posture and motion. The mirror, as well as, video and Kinect, unless combined with more than one devices, provide only one 2-dimensional perspective of the body and movement. In these cases, e.g., for ballet, where usually a more traditional teaching approach with the mirror is applied in the physical world, the metaphor of the "augmented mirror" and the use of Kinect [29, 30] have shown positive results regarding both usability and effectiveness in self-practice [31, 32, 33]. In such cases, the students looking at themselves and their posture and correcting it is the objective. In our case, however, they need to also consider the symbols themselves and link them to the 3D direction and body posture they correspond to looking at a 2D screen. So the cognitive load is greater.

7. DISCUSSION

7.1 Mirroring vs. Immersion

Kinect and Vive experiences, as it was revealed during the working sessions with the experts, were different in many aspects. Firstly, the time needed to familiarize with the environment was significantly less in the Vive, as the whole exercise seemed more self-explanatory, since the body is immersed in the kinesphere or the cube rather than presented on the 2D screen. Observation revealed that their body posture and movement was different, more free and natural. The feeling of immersion was strengthened by the fact that in the Vive experience the users see the symbols in their own physical environment rather than placed on a two-dimensional representation of their bodies. As one of the dance experts, choreographer and teacher explains: "In the Vive version, I definitely had more conscious feeling of my body and kinesthetic awareness, the attention was on my own body, rather than on the screen. In the Kinect I was searching for my body on the screen, so I somehow lost my sense of embodiment, it looked more like a funny game, but I was more connected with the image of my body rather than the sense of it." Another expert also notes that this feeling of immersion, might be an interesting way to trick non-dancers and people who are uncomfortable or shy with movement into some type of dancing. "Here you are not able to see the real world, you can't see if others are watching, you are lost in your own space, and this fact, combined with some playful elements might be a way to motivate people who do not usually feel like moving or dancing, since the focus is on the goal".

7.2 Free hands vs. controllers

Overall the use of the controllers in Vive, vs. having free hands in Kinect was not annoying or at least was compensated by the immersion and freedom of movement in the three-dimensional space, according to the dance experts. P4 notes: "It was strange though that I could see the controllers but not my hands, however this is a fact I very easily forgot and overcame." In fact using the controllers, brought to the discussion the metaphor of drawing in space, and further creative ideas, such as coloring the different directions. Another minor issue that we had in the Vive setting was the presence of the cable.

7.3 Continuous vs. discretised space

Another point that all the dance experts commented on was the continuity of space and continuation of movement as a feeling. While Laban's cube, and the 27 directions can be seen as benchmark points in space, that allow abstract communication, and thinking of the geometry of the body, by creating linear visual metaphors, there is much more in exploring space, both in Laban's theory and in kinesthetic awareness in general. One of the experts expressed the concern that looking for points and lines, rather than areas, volumes or even texture of the air that covers the space might convey a very linear, or "empty" way of moving. In fact, for the cultivation of movement literacy both levels are needed: there is one cognitive, analytical aspect of thinking on movement, that is where even as a choreographer you create the "skeleton" of the space and then there is kinaesthesis, the qualities and textures of the movement that you fill this skeleton with to make it a complete physical, embodied experience.

As P4 adds: "If I have to compare studying Laban symbols and the concepts using this application, rather than paper reading, then I would definitely vote for it, it is embodied, it is clear, it is fun and effective. Though I would never say that someone can learn to dance with this application, I see much more potential for learning the Laban symbols: it can also be a tool for teaching geometry, or architecture especially for young, digital natives.". P1 and P4 agree on the following: "The process of analysis and learning a conceptual framework is very important but completely different from the real, embodied experience although in a continuous dialogue. The concepts of *Body, Space, Effort and Shape* are very useful in helping young dancers understand the tools that they have, the range of possibilities, but in the real world in the embodied experience they all happen at the same time. For example: the focus might be on *Space*, and the question is to go from point A to point B having already a specific *Effort*, a specific quality."

7.4 Memorizing symbols vs. improving natural movement

During the completion of the tasks the dance experts were much more demanding, they needed to explore the difference of "pointing at" the direction, vs. "reaching out", to be in but also go out of their kinesphere or cube. During the interviews, some of them admitted that the Kinect was fun, but somehow restricted their movement, while others had the feeling that they had to adjust their movement to the system, though it was fun as a game. One of the experts observes "at certain points I had the feeling that I needed to move in a very restricted space, in a very particular way, this certainly affected my qualities". This observation reminds us of the question raised by Norman "how natural is natural interaction". To this point we need to add that this expert had never used Kinect or Vive.

7.5 Beyond the Cartesian space

During the co-design and evaluation sessions with dance experts, we have discussed ideas for transforming the experience into a more imaginative, creative and playful environment where other modalities of kinesthetic awareness related to space can be cultivated. We have discussed both the use of visual metaphors such as the one of drawing in space with traces while moving from one edge of the cube to the other. P3 highlights that "the edges of the cube, the places and the diagonal are important but what is also important, is the in-between space and its volume, its texture, especially if our focus is to use this for young dancers". P4 proposes to add another mode where the symbols will be replaced by images of tangible objects and the mover can create small stories by pointing at, reaching, or grasping these objects". Taking into account the impact of story-making as semantic aid for directional gesture interaction [39] a next version of the tool combining Kinect and Vive can be explored both for memory practice and creative context. Another important idea that emerged during the process, is that of "constructing" space. Usually virtual reality technologies are used as a means for entering a ready-made new environment, however, in many movement as well as somatic practices, one is asked to fill this place with their imagination.

Moreover, a very interesting area for future research is the connection between movement and sound. Special designed sound and audio cues could be given as a sonification feedback for example to reflect directions but also other qualitative aspects of movement (e.g., Effort). Furthermore, a correlation of directions and musical notes or musical scales can be considered since each direction could be a specific note and each level a specific octave. In that way the users "compose" a musical theme while they are moving. This bridge between music and movement could possibly help in understanding Laban symbols.

8. CONCLUSIONS

This paper contributes to the field of interdisciplinary research intersecting movement practice and analysis and embodied interaction design. We articulated concrete HCI challenges on conveying spatial kinesthetic and embodied knowledge. Through an experimental, co-design process with dance experts and practitioners we explored and documented the opportunities and limitations that are available in current commercially available technologies such as motion sensing and VR. We conclude that interactive technologies, can play an effective role in conveying conceptual, analytical knowledge such as teaching a symbolic language to the young digital natives as well as adults. In particular, motion depth cameras like the MS-Kinect, although they are effective for a dance learning context where the mirror-paradigm is used, according to the literature, they might be problematic when there is a stronger need to cultivate the sense of 3-dimensional space. In fact, they can lead the users into a more gestural, 2-dimensional pattern of moving. However we cannot overlook the advantages of such settings in relation to the low cost and complexity, as well as the fact that, as the experiment showed, they can become a powerful tool for memorization exercises and foster interdisciplinary, informal learning by making "paper subjects" such as learning notation symbol, more fun, interactive and engaging [2].

The fact that for the wider audience these types of settings are considered as "electronic games" is an advantage and a weakness at the same time: it is an advantage as they can attract young digital natives and engage them in analytical subjects in a more embodied and fun manner. On the other hand, there is always the risk of the digital medium imposing its own qualities on the movement, which sometimes derives from the limitation of the technology itself, rather than the initial intention of the design.

The fact that a playful embodied task oriented experience can "trick" people into moving or even dancing without realizing it is also a very important observation, made by the movement experts. This opens a wide range of applications to rehabilitation, and the potential of targeting user groups that are not keen on moving, nor convinced easily to start an activity.

Last but not least aesthetics and look and feel of the environment are of high importance. Though this aspect was not within our main focus in this study, interesting ideas have emerged during the co-design sessions for future development, in order to make it visually more attractive. We believe that although the setting is simple and the focus is on the cube, which is something that was considered a strength by the participants, our digital environment can definitely benefit from a future collaboration with visual and 3D artists to better convey the metaphoric aspects of space in a more inspiring manner.

Moreover, a very interesting area for future research is the connection between movement and sound. Special designed sound and audio cues could be given as a sonification feedback that reflect if a movement is right or wrong, for example. Furthermore, a correlation of directions and musical notes or musical scales would be very interesting since each direction could be a specific note and each level a specific octave. In that way the users "compose" a musical theme while they are moving. This bridge between music and movement could possibly help in understanding Laban symbols.

During this work we acknowledge that the continuous dialogue with dance and movement practitioners can open pathways in terms of perception and therefore representation of spatial aspects and can enrich the field of whole-body interaction as well as the design of innovative applications for notation. Dance and movement practitioners and researchers, can bring innovative insights in embodied and multi modal experiences through challenging and rethinking the relationship between the Cartesian, measurable, perceived and metaphoric space. Based on the outcomes, we consider a new co-design cycle to develop a more complete scenario of teaching Laban Movement Analysis using XR technologies. In addition, more systematic evaluation experiments need to be held in order to further study the outcomes of this initial experimental study.

Acknowledgments

We thank the dance experts that dedicated long hours throughout the process, as well as all the participants for their valuable feedback.

9. REFERENCES

[1] K. El Raheb and Y. Ioannidis, "From dance notation to conceptual models: a multilayer approach," in *Pro-ceedings of the 2014 International Workshop on Move-ment and Computing*. ACM, 2014, p. 25.

[2] A. Dania, "Dania , A ., Tyrovola , V ., & Koutsouba , M . (2011). Proposal for a method for teaching dance skills . Proceedings of the 27th Biennal ICKL Conference , Institute for Musicology , Budapes ... Proposal for a Method for Teaching Dance Skills," *Proceedings of the 27th Biennal ICKL Conference, Institute for Musicology*, no. January 2011, 2014.

[3] Y. Candau, J. Françoise, S. Fdili Alaoui, and T. Schiphorst, "Cultivating kinaesthetic awareness through interaction," *Proceedings of the 4th International Symposium on Movement and Computing*, pp. 35–40, 2017.

[4] J. Françoise, Y. Candau, S. Fdili Alaoui, and T. Schiphorst, "Designing for kinesthetic awareness: Revealing user experiences through second-person inquiry," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 5171–5183.

[5] J. Zimmerman, J. Forlizzi, and S. Evenson, "Research through design as a method for interaction design research in hci," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2007, pp. 493–502.

[6] "Microsoft kinect sdk," https://developer.microsoft.com/en-us/windows/kinect.

[7] "Htc-vive, virtual reality headsets," https://www.vive.com/eu/.

[8] "Unity cross platform game engine," https://unity.com/.

[9] J. S. Longstaff, "Re-evaluating rudolf laban's choreutics," *Perceptual and motor skills*, vol. 91, no. 1, pp. 191–210, 2000.

[10] A. H. Guest, *Labanotation: the system of analyzing and recording movement*. Routledge, 2013.

[11] J. A. Adams, "A closed-loop theory of motor learning," *Journal of motor behavior*, vol. 3, no. 2, pp. 111–150, 1971.

[12] M. Nakamura and K. Hachimura, "Development of Multimedia Teaching Materialfor LabaNotation," in *ICKL 2001*, 2001.

[13] L. Wilke, T. Calvert, R. Ryman, and I. Fox, "From dance notation to human animation: The LabanDancer project," *Computer Animation and Virtual Worlds*, vol. 16, no. 3-4, pp. 201–211, 2005.

[14] W. Choensawat, M. Nakamura, and K. Hachimura, "GenLaban: A tool for generating Labanotation from motion capture data," *Multimedia Tools and Applications*, vol. 74, no. 23, pp. 10 823–10 846, 2015.

[15] K. El Raheb and Y. Ioannidis, "A labanotation based ontology for representing dance movement," in *International Gesture Workshop*. Springer, 2011, pp. 106–117.

[16] L. Loke, A. T. Larssen, and T. Robertson, "Labanotation for Design of Movement-Based Interaction," *Second Australasian Conference on Interactive Entertainment*, vol. 1996, no. November, pp. 113–120, 2005.

[17] L. Bishko, "Animation principles and laban movement analysis - movement frameworks for creatinc empathic character performances," *Nonverbal Communication in Virtual Worlds: Understanding and Designing Expressive Characters*, pp. 177–203, 2014.

[18] M. M. Mainsbridge, "Body as instrument: an exploration of gestural interface design," Ph.D. dissertation, 2016.

[19] S. F. Alaoui and R. Nesbitt, "Self and self portraits," in *Canadian Society of Dance Studies Symposium on Embodied Artful Practices*, Vancouver, Canada, 2014.

[20] A. L. de Souza, "Laban movement analysis scaffolding human movement to multiply possibilities and choices," in *Dance Notations and Robot Motion*. Springer, 2016, pp. 283–297.

[21] L. Ullmann, *Choreutics*. London: Macdonald & Evans, 1966.

[22] J. Newlove and J. Dalby, *Laban for all*. Taylor & Francis US, 2004.

[23] R. Von Laban, *Choreutics*. London: Macdonald & Evans, 1966.

[24] R. von Laban, *The language of movement: A guidebook to choreutics*. Plays, inc., 1966.

[25] T.-T. Prinsloo, M. Munro, and C. Broodryk, "The efficacy of laban movement analysis as a framework for observing and analysing space in rosas danst rosas," *Research in Dance Education*, pp. 1–14, 2019.

[26] M. Sulzman, "Choice/form in trisha brown's locus: A view from inside the cube," *Dance Chronicle*, vol. 2, no. 2, pp. 117–130, 1978.

[27] W. Forsythe. (2012) Improvisation technologies: a tool for the analytical dance eye.

[28] K. El Raheb, S. Whatley, and A. Camurri, "A conceptual framework for creating and analyzing dance learning digital content," in *Proceedings of the 5th International Conference on Movement and Computing*. ACM, 2018, p. 2.

[29] Z. Marquardt, J. Beira, N. Em, I. Paiva, and S. Kox, "Super mirror: a kinect interface for ballet dancers," in *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2012, pp. 1619–1624.

[30] A. Aich, T. Mallick, H. B. Bhuyan, P. P. Das, and A. K. Majumdar, "Nrityaguru: A dance tutoring system for bharatanatyam using kinect," in *Computer Vision, Pattern Recognition, Image Processing, and Graphics: 6th National Conference, NCVPRIPG 2017, Mandi, India, December 16-19, 2017, Revised Selected Papers 6*. Springer, 2018, pp. 481–493.

[31] M. Trajkova, "Usability Evaluation of Kinect-Based System for Ballet Movements Usability Evaluation of Kinect-Based System for Ballet," no. June, 2015.

[32] M. Trajkova and F. Cafaro, "E-ballet: designing for remote ballet learning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 213–216.

[33] "Takes Tutu to Ballet : Designing Visual and Verbal Feedback for Augmented Mirrors," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol*, vol. 10, no. 20, pp. 1–30, 2018. [Online]. Available: https://doi.org/10.1145/3191770

[34] S. Yan, G. Ding, Z. Guan, N. Sun, H. Li, and L. Zhang, "Outsideme: Augmenting dancer's external self-image by using a mixed reality system," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2015, pp. 965–970.

[35] A. Ballas, T. Santad, K. Sookhanaphibarn, and W. Choensawat, "Game-Based System for Learning Labanotation Using Microsoft Kinect," in *IEEE 6th Global Conference on Consumer Electronics (GCCE 2017) Game-Based*, no. Gcce, 2017, pp. 2–4.

[36] "A mobile app for dance notation," https://u.osu.edu/kinescribe/, accessed: 2019-12-30.

[37] "Moving Space: the laban scales," https://itunes.apple.com/us/app/moving-space-the-laban-scales/id396089751?mt=8, accessed: 2019-12-30.

[38] "Dance like the future is watching," https://insights.osu.edu/arts/dance-augmented-reality, accessed: 2019-12-30.

[39] B. Fruchard, E. Lecolinet, and O. Chapuis, "Impact of semantic aids on command memorization for on-body interaction and directional gestures," in *Proceedings of the 2018 International Working Conference on Advanced Visual Interfaces*, 2018.

[40] E. Márquez Segura, L. Turmo Vidal, and A. Rostami, "Bodystorming for movement-based interaction design," *Human Technology*, vol. 12, 2016.

AIDING THE PERFORMANCE OF TEMPO CANON: NEW TECHNOLOGY IN STUDIES 1, 2 AND 3 FOR STRING QUARTET

James Opstad
University of Birmingham
Birmingham, UK
jamesopstad@gmail.com

ABSTRACT

This paper discusses the application of new score-reading technologies in a set of string quartet studies written for Apartment House. I highlight how the use of technology facilitates complex polytemporal relationships within the ensemble and allows these to be conveyed in a simpler and more direct manner. I demonstrate the current state of the score application and draw attention to design features that differ from existing approaches as well as changes that have been made in response to performer feedback. This follows a brief discussion of tempo canon in both my own work and its broader historical context.

1. INTRODUCTION

The three string quartet studies discussed in this paper are examples of tempo canon. This is a type of canon where the same melody is superimposed at different speeds. As a form, it poses particular challenges to an ensemble as each performer must maintain their own unique pulse that is nonetheless in a precise relationship to the other players. Individual performers may also be required to shift suddenly to distant tempi with a high degree of precision. I have developed a new networked score-reading application that overcomes these obstacles by embedding a visual metronome in each player's part. It is distributed as a Progressive Web Application and is optimised for tablet devices.

The historical precursor of the tempo canon is the prolation canon (or mensuration canon), a musical form that first became popular in the early Renaissance. Leading exponents of this style included Johannes Ockeghem and Josquin des Prez. The *Missa Prolationum* by Ockeghem is the most rigorous exploration of this form and it has been said that it "may well be the most extraordinary contrapuntal achievement of the 15th century" [1]. A section of the original manuscript is included in Fig. 1 and a transcribed excerpt is shown in Fig. 2. A single voice is written out for each canon and the mensuration markings indicate the respective alterations of the written durations. Fig. 3 translates this into modern notation, revealing the intricate ways

Figure 1. Manuscript page of the first "Kyrie" from the *Missa Prolationum* by Johannes Ockeghem

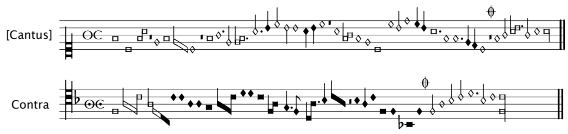


Figure 2. Excerpt from the *Missa Prolationum* (mensural notation)

in which the parts overlap and intertwine. It should be noted that the mensuration markings do not alter all note values and it is this that distinguishes prolation canon from tempo canon. In a tempo canon, all durations are scaled by the same ratio. The Agnus Dei from Josquin des Prez's *Missa L'homme arme* maintains a 2:1:3 ratio throughout and can therefore be considered a tempo canon as well as a prolation canon. [2]

During the 20th century, tempo canon was used extensively in the compositions of Conlon Nancarrow. In his works for player piano, the relationships between tempi reached ever greater levels of complexity and included rational, irrational and even transcendental numbers. An excerpt from the piano roll for *Study 49c* is shown in Fig. 4. A defining characteristic of this music is that it demands performance by machines. Though there have been significant attempts to perform some of these works in recent years [1] , the extremely intricate layering of tempi is difficult to perform accurately and Nancarrow's music for ensembles therefore favoured simpler relationships. Nancarrow did, however, theorise about the possibility of using syn-

[1] A key example is *Nancarrow: Studies and Solos - Bugallo-Williams Piano Duo*.

Figure 3. Excerpt from the *Missa Prolationum* (modern notation)

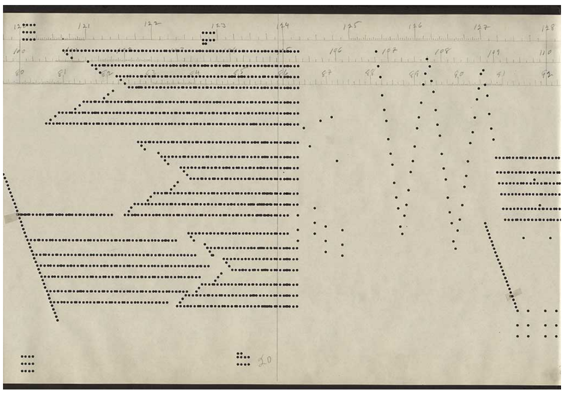


Figure 4. Excerpt from *Study for player piano no. 49c* by Conlon Nancarrow

chronised video conductors to assist performance.

> And I've got the idea of each performer having a small television screen, with something imitating a conductor that comes to the beat, so they can see it coming, whatever it is. I don't think it would be too complicated. It would probably be expensive, each one having his own screen. [2]

Nowadays it is commonplace for each performer to have their own screen in the form of a tablet computer. Nancarrow's idea can therefore be realised and the challenges implicit in the performance of tempo canon can be overcome.

2. THE MUSIC

My own music differs significantly in style from Nancarrow's but shares some of the same underlying principles. In particular, the concepts of *convergence period* and *echo distance* have structural importance. The convergence period is described by Kyle Gann as "the hypermeasure that exists between (potential) simultaneous attacks in voices moving at different tempos", while the echo distance is "the temporal gap between an event in one voice and its corresponding recurrence in another" [2]. These two concepts play an important role, both during the compositional process and in the way the music is perceived.

Each of the three string quartet studies has a global base tempo to which the other tempi relate. The correspondence between a local tempo and the base tempo is always a rational number expressed as a ratio of two integers.

Figure 5. Excerpt from *Study 1* (start)

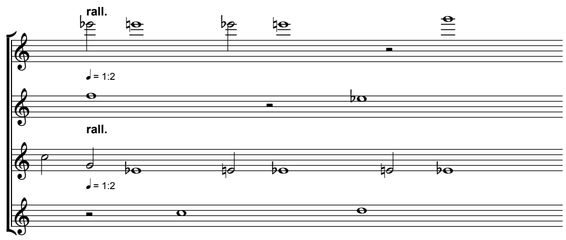


Figure 6. Excerpt from *Study 1* (second rallentando)

2.1 Study 1

The four instrumental parts of *Study 1* are derived from a two-part canon. The upper part is played by the violins while the lower part is played by the viola and cello. For the first section of the piece, violin 1 and viola maintain a static tempo while violin 2 and cello perform a controlled rallentando. The opening is shown in Fig. 5. At the tempo marking in Fig. 6, violin 2 and cello reach half the original tempo and remain there for the duration of the piece. Simultaneously, violin 1 and viola begin a rallentando, continuing until all parts coincide at the final chord. Within each rallentando, each beat has a different tempo and the durations are scaled accordingly using hidden tuplets. These are shown in Fig. 7 and reveal the underlying structure. The score-reading application allows this complexity to be hidden from the performers and for the intention to be conveyed very simply and directly. The unidirectional nature of the tempo changes in *Study 1* means that the echo distance between the two instrumental pairs continues to grow throughout. The expansion of two-part counterpoint to four independent parts is reminiscent of the *Missa Prolationum* excerpt shown in Fig. 3.

2.2 Study 2

Study 2 is the only one of the three studies that begins with staggered entries, as is typical for a canon, and is also unique in the set for having all four parts play the same melodic line. The changes in tempo are less predictable, however, with each part adhering to its own temporal scheme. The gradual changes of *Study 1* are replaced by marked changes in tempo every fifty four quaver beats. These initially become slower, once again creating a feeling of rallentando, before increasing in speed. By this point, the four parts are intertwined in a complex web of convergence periods and echo distances. An excerpt from

Figure 7. Excerpt from *Study 1* (start) with tuplets shown

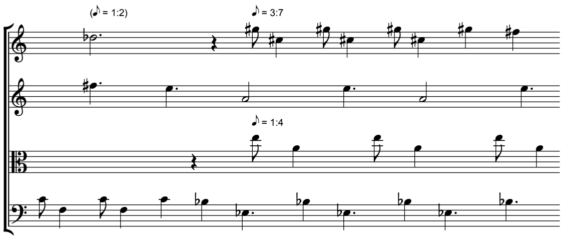


Figure 8. Excerpt from *Study 2*

the score is shown in Fig. 8.

2.3 Study 3

Whereas *Studies 1 and 2* are unbarred, *Study 3* has bar lines and a static 4/4 time signature. In contrast to *Study 2*, all four parts play different melodic material. These are derived from the same six bar melody and constitute the four specular transformations: original, retrograde, inversion and retrograde inversion. This piece is influenced by the music of Aldo Clementi, a composer who made extensive use of canonic techniques. The transposition levels were decided with the aid of an algorithm derived from analysis of Clementi's compositional methods [3]. As in *Study 2*, each part moves between tempi independently, though here the order of tempi is itself canonic. An excerpt from the score is shown in Fig. 9.

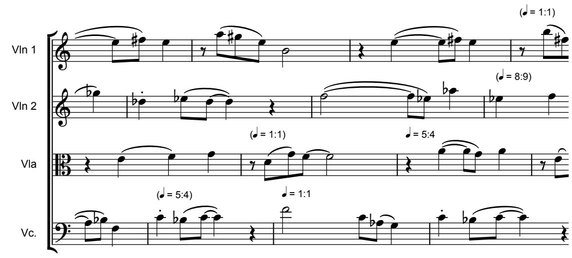


Figure 9. Excerpt from *Study 3*

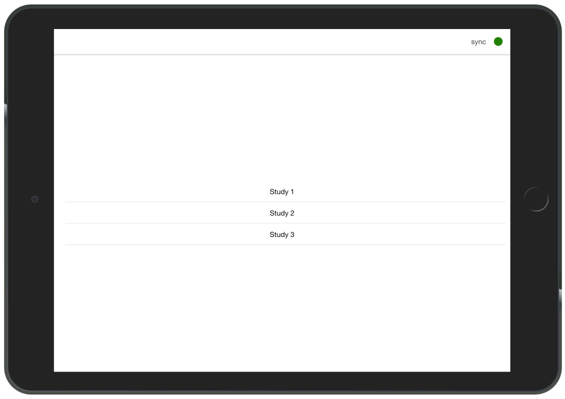


Figure 10. Application home screen

3. THE APPLICATION

When designing the score-reading application, a range of criteria were taken into consideration. I wanted it to function both standalone and in a networked environment so that it would be equally well suited to practice, rehearsal and performance. It was also important for the timing to be as accurate as possible and for the embedded visual metronome to maintain a consistent frame rate of 60 frames per second. For these reasons, I decided the score and associated timing data should be preloaded onto the device and all events should be scheduled locally. This contrasts with other existing approaches such as INScore [4] and Drawsocket [5] where each event is communicated in real-time over a network. In my application, network traffic is kept to a minimum and the only data that is transferred are transport commands (start, stop and snap) and a periodic synchronisation algorithm. This avoids potential issues with network latency and has the added benefit that the system is very robust in performance. If the network fails, the performance can still continue. The application is packaged as a Progressive Web Application and can be hosted on the internet for easy access by individual performers. It is well suited to use without the composer present as no additional technical knowledge is required.

3.1 Navigation and Layout

The home screen of the application is shown in Fig. 10. The synchronisation status is displayed in the header. The circle next to this is green when the device is connected to the server and red when disconnected. When red, it can be tapped to reconnect. In the centre of the screen, the available scores are listed. Selecting a score takes the user to the screen shown in Fig. 11. This displays the available parts as well as adding a back button and the selected score to the header. On selecting a part, the user is taken to the score-reading screen shown in Fig. 12. The main area is then occupied by two systems of score. Additionally, transport controls and the selected part name have been added to the header, and a footer is displayed containing a slider. This slider displays the current position during performance and otherwise allows the performer to navigate the score.

3.2 Rehearsal and Performance Considerations

There are several respects in which the practical aspects of a rehearsal are taken into account in the design of the application. Each player can navigate the score independently in order to look through their part. If a player wishes to move everyone to the same location, they can use the snap feature. In addition, the start and stop controls are available to all players, giving each member of an ensemble

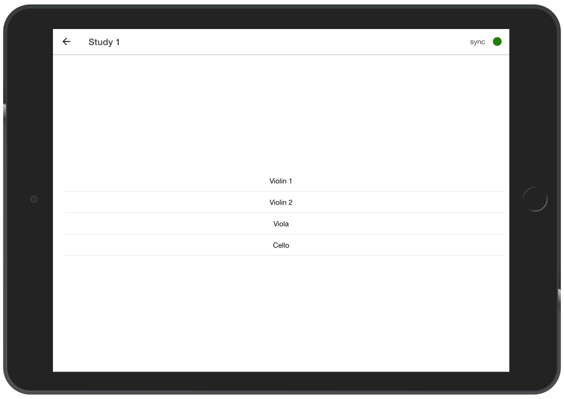


Figure 11. Part selection screen

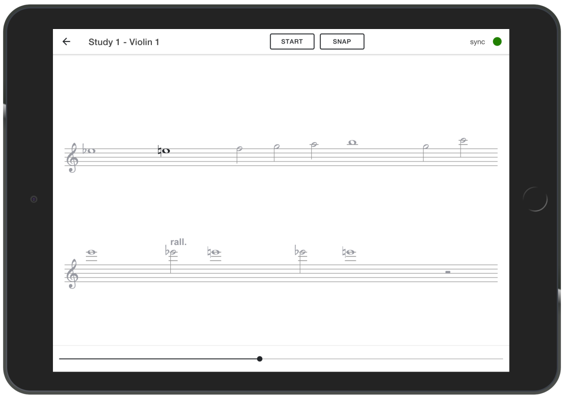


Figure 12. Score-reading screen

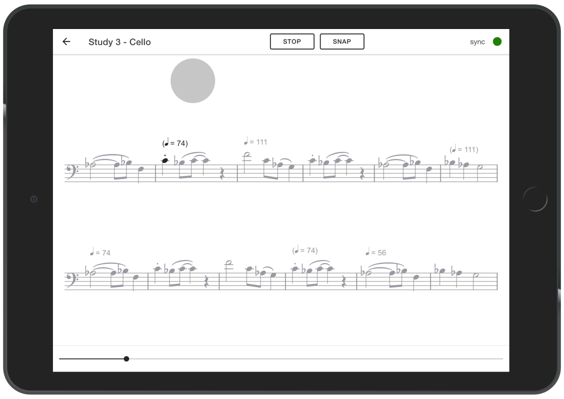


Figure 13. Score-reading screen during performance

equal control. The original tempo markings, expressed as ratios, are translated into literal metronome marks determined according to the base tempo. The base tempo can be altered and this will automatically adjust the metronome marks for all players. This allows the overall speed to be adapted during rehearsals.

There are other design features that ease both rehearsal and performance. The performer always reads from the top system. The bottom system allows the player to look ahead in the music and the top system is replaced by the bottom system as the music proceeds. Performers find this to be very natural and appreciate being able to look ahead in the music. It was decided that the current system should always be at the top rather than alternating as this maintains a consistent distance between the notation and the visual metronome. Additionally, the current event in the score is always highlighted (as shown in Fig. 12). This is useful, both when navigating the score and during performance, and overcomes the main drawback of unbarred music, that the absence of regular visual cues makes it harder to keep one's place.

3.3 Visual Metronome

The key distinguishing feature of the application is the visual metronome. This is crucial to the performance of the three studies as it provides a visual reference for the current pulse and this pulse is often unique to each part. The design of the metronome went through several versions of varying degrees of complexity. These explored different kinds of motion and some simulated more closely the movements of a real conductor. In the end, however, I reverted to one of the simplest designs. This is a dark grey circle that fades linearly to white over the duration of a beat. I settled on this after establishing some important criteria in response to performer feedback. These were: it must be clear and simple enough to be used as a reference in peripheral vision; the start of each beat must be completely unambiguous; and the speed of the beat must be clearly and quickly discerned. Practical comparisons and performer feedback led to the current design being favoured over the 'bouncing ball' approach used in applications such as ZScore[6] and Comprovisador[7]. It also became evident through working with the musicians that the visual metronome works in combination with the event highlighting in conveying a clear sense of pulse. Fig. 13 captures the metronome partway through a beat.

3.4 Technology

The score-reading application was created using web technologies and is distributed as a Progressive Web Application. This allows it to be accessed on any platform with a standards-compliant browser, including both iOS and Android mobile devices. The app can be added to the user's home screen and will then open in a full-screen window.

3.4.1 User Interface

The user interface is built using *React* [2], an open-source Javascript library that was originally developed by Facebook. *React* makes it possible to build an application by combining small, encapsulated components and declaring how these respond to changes in state. It is well suited to musical scores where the state is changing over time or in response to events. The score itself is rendered using the SVG format. Each system comprises a top-level group and, within each system, the score elements are grouped in the hierarchical tree structure shown in Fig. 14. The score SVGs were created using the *Dorico* [3] music notation software. The groupings were applied manually using *Affinity*

[2] https://reactjs.org/
[3] https://new.steinberg.net/dorico/

```
<staff>
  <layer>
    <event>
      // event data
    </event>
    ...
  <\layer>
  ...
</staff>
...
```

Figure 14. Score data structure

```
interface EventInput {
  duration: number
  tempo: {
    numerator: number
    denominator: number
  }
}
```

Figure 15. Typescript interface for event input

Designer [4] as *Dorico* does not include semantic data in exported SVGs. A compilation step interprets this structure to assign a unique identifier to each musical event. The entire SVG is loaded into the DOM and the bounding box coordinates are used to set the SVG viewBox to the current system. This has significant performance advantages as adding and removing lots of elements to and from the DOM can be slow, causing a drop in the browser frame rate and visible stuttering.

3.4.2 State Management

The application logic is built with *Redux* [5], an open-source Javascript library for managing application state. This enabled me to keep state management distinct from the user interface layer, preventing overdependence on a single library and easing testing and development. This decoupling also allows the core timing logic to run in different environments, including both client and server. In my composition *Eluvium*, for clarinet and live electronics, this meant I could use the same core application to create an electronic score, running in *Node.js* [6], that sent commands to *SuperCollider* [7]. The score timing data is stored in the JSON format and mirrors the hierarchical structure used in the SVG (Fig. 14). Each event is an object with the shape shown in Fig. 15. The duration is a decimal expressing the number of beats and the tempo is a fraction expressing the relationship to the base tempo. The client application then converts this into an object with the shape shown in Fig. 16, determined according to the current base tempo. The values are all decimals representing seconds.

[4] https://affinity.serif.com/en-gb/designer/
[5] https://redux.js.org/
[6] https://nodejs.org/
[7] https://supercollider.github.io/

```
interface EventOutput {
  start: number
  duration: number
  end: number
}
```

Figure 16. Typescript interface for event output

3.4.3 Server

In an ensemble setting, each performer uses WiFi to access the same local area network and connect to a WebSocket server. This is running in *Node.js* and is responsible for relaying transport commands to all connected clients and synchronising each client to a master clock. The clock synchronisation is done using the open-source *@ircam/sync* [8] Javascript library. This periodically compares the client clock to the master clock and compensates for any drift [8]. In practice, this means that all connected clients can share a common master clock to synchronise events. When a performer presses ‘play’, this sends the command to all connected devices along with a start time in the future. By default, the start time is a tenth of a second after the ‘play’ command is sent. This ensures that all devices start at the same time. The local clocks continue to stabilise as the performance proceeds.

4. CONCLUSIONS

The score-reading application presented in this paper has proved itself to be a practical and effective aid in the performance of tempo canon. Practice and rehearsal scenarios have been considered on an equal footing with performance and the performer experience has been carefully factored into the design. New works will present further challenges and I look forward to continuing to use and develop the application. One feature that has been frequently requested by performers is the ability to annotate a score and I hope to incorporate this in the future. I would also like to make the application publicly available so that it can be used by others. At present, the main obstacle to wider use is the process of grouping events in the score SVG, which is laborious and error prone. Though there are applications such as *Verovio* [9] that can encode semantic musical data in the SVG output [9], I find these too restrictive when compared to full featured score-writing applications such as *Dorico* and *Sibelius*. I will therefore continue to research and develop methods to streamline this process.

Acknowledgments

I would like to thank Midlands4Cities for their financial support and the members of Apartment House, Mira Benjamin, Gordon MacKay, Bridget Carey and Anton Lukoszevieze for their playing and input.

[8] http://collective-soundworks.github.io/sync/
[9] https://www.verovio.org/

5. REFERENCES

[1] L. L. Perkins, "Ockeghem [okeghem, hocquegam, okegus etc.], jean de," 2001. [Online]. Available: https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-0000020248

[2] K. Gann, *The music of Conlon Nancarrow*. Cambridge University Press, 1995.

[3] M. Zaccagnini, "Aldo Clementi's System; and an original composition, Variazioni su AlDo ClemEnti for chamber orchestra," Ph.D. dissertation, 2014.

[4] D. Fober, G. Gouilloux, Y. Orlarey, and S. Letz, "Distributing Music Scores to Mobile Platforms and to the Internet using INScore," in *12th Sound and Music Computing Conference (SMC15)*, Maynooth, Ireland, Jul. 2015, pp. 229–233. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01348511

[5] 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'19*, C. Hope, L. Vickery, and N. Grant, Eds. Melbourne, Australia: Monash University, 2019, pp. 15–25.

[6] S. Zagorac and P. Alessandrini, "Zscore: A distributed system for integrated mixed music composition and performance," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 62–70.

[7] P. Louzeiro, "Improving sight-reading skills through dynamic notation – the case of comprovisador," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 55–61.

[8] J.-P. Lambert, S. Robaszkiewicz, and N. Schnell, "Synchronisation for Distributed Audio Rendering over Heterogeneous Devices, in HTML5," in *2nd Web Audio Conference*, ser. Proceedings of the 2nd Web Audio Conference (WAC-2016), Atlanta, GA, United States, 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01304889

[9] L. Pugin, "Interaction perspectives for music notation applications," in *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music*, ser. SAAM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 54–58. [Online]. Available: https://doi.org/10.1145/3243907.3243911

THE NOTATION OF SOUND FOR COMPOSITION AND TRANSCRIPTION–AN ADAPTATION OF LASSE THORESEN'S SPECTROMORPHOLOGICAL ANALYSIS

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

ABSTRACT

This paper details my adaptation of Lasse Thoresen's spectromorphological analysis notation for the sake of composition and transcription, re-imagining the analysis symbols for use over a spectrum staff system over which pitch and spectra can be indicated with great detail, and possibly interpreted by musicians and computers for performance. A sound object is notated with regard to its spectral width, density, centroid frequency, significant sound components, modulation and amplitude envelope. It can also have a spectrum reference. The symbols are placed over a spectrum grand-staff with a frequency scale to show each parameter both from a frequency and pitch perspective. Also included are suggestions for the visual representation of spatialisation where positions and movements are displayed in two or three dimensions above the sound notation while constant rotations are notated as modulations.

1. INTRODUCTION

Lasse Thoresen's spectromorphological analysis was the result of years of teaching Schaeffer's typomorphology at the Norwegian State Academy of Music [1]. As part of the development of early electroacoustic music, Schaeffer developed a vocabulary and a typology culminating in the famous TARTYP [2] diagram with 28 categories that in combinations were to describe most if not all sounds of a music not only concerned with pitch structures. This was an important, if not necessary, development with regard to a traditional musicology not ready for a music of recorded trains and saucepans [1] . As the name suggests, Thoresen's research also builds on Denis Smalley's influential theory of *spectromorphology*–a highly developed framework for studying structural relations in music over time.[3].

In order to make practical use of Schaeffer's typomorphology Thoresen streamlined Schaeffer's diagram and reduced the 28 categories to 15, keeping the nine core categories as well as the six extremes of unpredictable nature. New categories were then introduced to fill the gaps between the core categories and the extremes. Also, Schaeffer's normative vocabulary was removed since he had, in his typomorphology, incorporated ideas relating to the suitability of sounds for musical use. But most importantly, graphic symbols were introduced making possible graphic analyses of music with a detail and consistency not possible before. To my knowledge, the *Spectromorphological analysis of sound objects* (hereinafter SASO), as described in [1] by Thoresen assisted by Hedman, remains the most detailed and developed symbolic system for analysis of sound structures to this date.

The background for my adaptation of SASO comes from teaching electroacoustic music since 2004, first at EMS and later at the Royal College of Music in Stockholm. When teaching composition students how to analyse the organisation of sound through aural sonology, I found it fruitful to also focus on the act of organising sound. However, neither Schaeffer's research nor Thoresen and Hedman's development of the same were aimed for composition. Their tools were developed to describe what is heard. Categories and symbols used for the composition of musical structures, on the other hand, are not only descriptive, but also meant to communicate a musical composition for performance. I found that in order to make full use of Thoresen and Hedman's notation in a compositional context, their symbols needed to be translated into acoustic properties that can be communicated and interpreted by musicians and computers. This translation means both reducing and expanding the symbol palette while developing a practice for placing the symbols over a fixed time-frequency-oriented staff system.

2. THE NOTATION SYSTEM IN DETAIL

Unless indicated otherwise, the notation symbols described here were all originally developed by Thoresen assisted by Hedman. Refer to [1] for a more detailed description of their system, here abbreviated as SASO.

2.1 Sound spectrum

2.1.1 *The spectrum staff system*

The first major adaptation of SASO is the placement of symbols over a hybrid frequency-staff system where specific pitches are easily identified while a frequency scale

[1] *Étude aux chemins de fer* and *Étude aux casseroles* are two parts of Pierre Schaeffer's genre-originating work *Cinq études de bruits*

helps relating spectral data to the actual frequency contents of the sound (see Figure 1). The system covers the complete listening range of our ears but may of course be decimated to using fewer staves if the notated information does not make use of the full spectral range. This kind of grand staff system is common in software for computer-assisted composition for control and/or display of data as pitch, e.g. the *nslider* in Max [2]. One can of course remove the staff systems and rely solely on the frequency scale for music where exact pitch relations have no significance. For such passages I would still use grey horizontal lines to mark the boundaries of each octave.

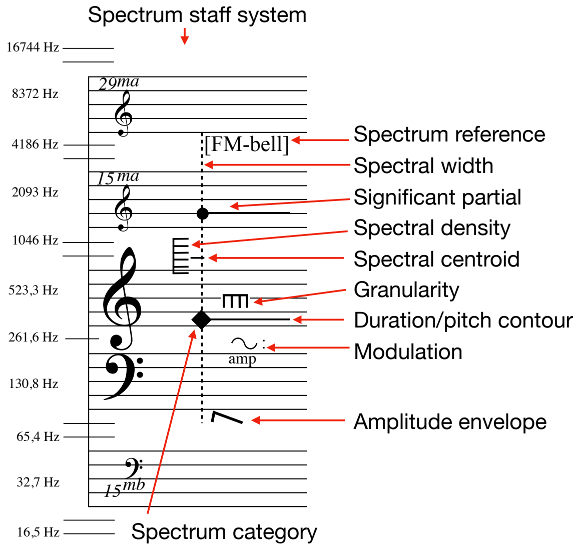


Figure 1. Spectrum staff and an example sound object with several indicators of sound spectrum and energy articulation.

2.1.2 *Spectrum categories*

The three core spectrum categories, pitched, dystonic and complex retain their meaning: *pitched* sounds are sounds with pitch, *dystonic* sounds have inharmonic spectra or are clusters while *complex* sounds have no pitch. We need different strategies for notating the "pitch contour" of these types of spectra since we perceive them differently. Pitched sounds, which have harmonic spectra, are (naturally) notated at the position of the root frequency of the spectrum. Dystonic sounds, which have inharmonic spectra or are clusters of pitched sounds, are placed at the position of the most significant partial. Complex sounds are notated at the position of the spectral centroid, marking the centre frequency of the spectral content of the sound. Figure 2 shows the three main spectrum category symbols and what spectral features that define their placements over the staff system.

2.1.3 *Spectral width*

Different from SASO, spectral width indicates the significant frequency range of a sound's spectrum with a dashed

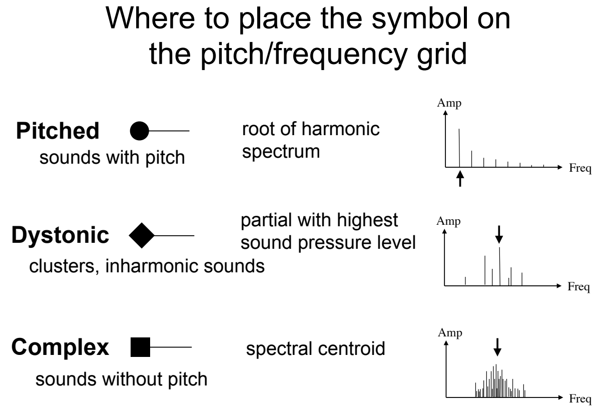


Figure 2. The three spectrum categories, their notation symbols and what spectral feature that determines their vertical positions on the spectrum staff

vertical line across the spectral staff system. For filtered sounds this may be equal to the full frequency range of its spectrum while for non-treated acoustic sounds it makes sense to indicate a frequency range of the relatively louder portion of the spectrum. Figure 3 shows three examples of spectral width: A is a pitched sound with spectral width 440 Hz–2,6 kHz. Since a pitched sound is notated at its root, the spectral width is never below its symbol. B is a dystonic sound with spectral width 164 Hz–2,1 kHz and C is a complex sound of width 147 Hz–1 kHz. The straight horizontal line marking the higher limit of the width of example C indicates that there is no spectral content above this line. In SASO, spectral width is presented as a continuum of different sound spectra from sine tone to white noise [1].

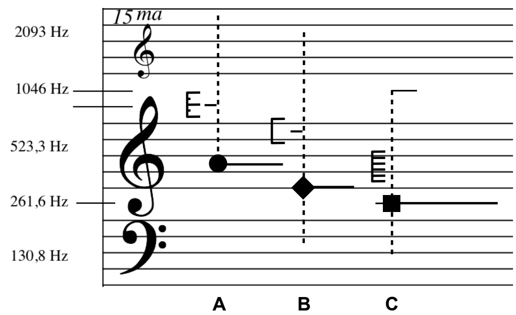


Figure 3. Three examples of spectral width in the pitched (A), dystonic (B) and complex (C) spectrum categories, which are the vertical dashed lines. Spectral density is indicated with the comb-like symbols to the left of the width lines, while spectral centroid is shown as small line indicators on the left side of the width lines for A and B.

2.1.4 *Spectral centroid*

The spectral centroid is the centre frequency of a sound's spectral energy. It is one valuable descriptor for the perceived brightness of a sound, though not the only one. Other factors, such as frequency range, also play a part. It is indicated as a small horizontal line indicator on the left side of the spectral width vertical line. Combined,

[2] https://cycling74.com/products/max

the width and centroid indicators resemble the symbol for spectral brightness in SASO denoting the perceived brightness of a sound. Figure 3 includes three examples of spectral centroid indications: examples A and B have specific centroid indicators, while C is from the complex spectrum category and has therefore its symbol (a solid square shape) at the centroid frequency position.

2.1.5 Spectral density

A comb-like symbol represents spectral density, as a value of the density of partials with high amplitude in a sound's spectrum, contingent on the sound's spectrum category. For purely pitched sounds (with harmonic spectra) maximum density in terms of positions of partials is dependent on the root frequency ant its multiples, while the spectra of dystonic (inharmonic or clusters) and complex sounds can be saturated to the extent that they eventually turn into noise (both ending up in the complex category). The number of teeth of the vertical comb-like indicator provides a relative value of density from lowest possible (two teeth) to maximum (six teeth). There are also two special cases: a maximum saturated spectrum, i.e. noise, indicated with a thick toothless comb, implying that the teeth are too close to separate, and a particular comb-symbol for indicating spectra with only every other partial–a spectral phenomenon commonly referred to as having a "hollow" sound quality. See Figure 4 for an overview of the density symbols. These are placed over and to the left of the spectrum category symbol, unless there is a spectral centroid indicator in which case the symbol is placed to the left of this indicator. Figure 3 shows three examples of density: A has only every other partial, B has a sparse spectrum while C has a very dense spectrum though it is not pure noise which would have yielded the maximum density symbol. The every-other-partial symbol can also be configured to convey different degrees of density as one sees fit. SASO's equivalent symbol is placed on the extension line of the sound object to indicate spectral saturation [1].

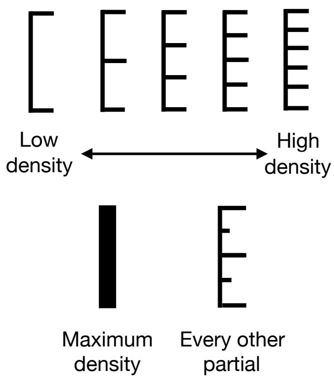


Figure 4. Overview of spectral density symbols

2.1.6 Significant partials and components

Particularly dystonic sounds, like bells with inharmonic spectra, can have multiple significant partials that are clearly audible and whose frequencies are highly relevant when composing for them. While the partial considered the most significant provides the position of the main spectrum symbol, any other significant partials are indicated with small symbols. Any spectrum category symbol can be used to represent significant components of a sound. E.g. an electric fan may have both a noise component and a humming pitched component. When used as a pitched sound for composition, the noise would be considered a complex component of the pitched sound.

2.1.7 Spectrum reference

Not part of SASO, spectrum reference is a text label for the spectrum to indicate a particular spectrum identity. It may be the spectrum recognised from our shared bank of culturally conditioned references such as the sound of an alto saxophone or a large church bell, or it may be a spectrum identity established during the course of a single musical work. These references can be indicated in two ways, either as *being* the spectrum of a known reference or as something *resembling* a certain reference. Both are indicated as text labels within square brackets as shown in Figure 5. The difference is that a reference of resemblance is italicized, within simple quotes and has no capital letter. The reference is positioned to the right of the higher limit of the spectral width vertical line as shown in Figure 1.

[Trumpet]	is the spectrum of a trumpet
['*trumpet*']	sounds like the spectrum of a trumpet

Figure 5. Spectrum reference, indicated as either being or resembling a known spectrum

2.2 Energy articulation

2.2.1 Pitch/spectral contour and extension lines

The extension line from the spectrum category symbol both serves to show how the parameter indicated by the symbol's vertical position changes over time, and the duration of the sound.

Any frequency-dependent sound spectrum indicator, such as the high and/or low value of the spectral width, can have dashed extension lines to indicate changes. These are then treated as time-dependent breakpoints when represented as data.

2.2.2 Granularity and iterative sounds

Granular or iterative sounds is the phenomenon when a chain of rapidly repeated sound grains form one continuous sound. The symbols used are basically the original horizontal comb-like symbols from SASO with different numbers of teeth for different granularity speeds, though with a five-step scale rather than three. The roundness of the symbol's angles can be varied for large, small and moderate granularity coarseness. However, I introduce another level of severe granular coarseness, when there are perceived silences between the grains. This is indicated with the back of the comb-symbol removed to clarify the separation of grains. For Thoresen, when adapting Schaeffer's ideas, granularity and iterative sounds are two different concepts [1] but I find it more useful in this context to treat them as one and the same. See Figure 6 for an overview of granularity symbols of different degrees of coarseness and velocity and Figure 1 for its placement over the extension line. Different sound components may have different granularity symbols.

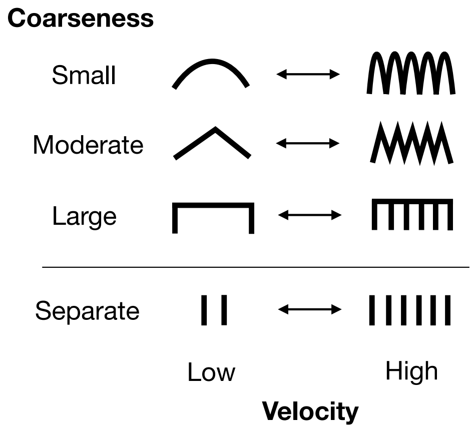


Figure 6. Granularity symbols of different degrees of coarseness and velocity

2.2.3 Accumulation

Accumulations are hordes of sound objects, not to be confused with sounds with granularity where a chain of grains form one continuous sound. The sounds involved in an accumulation are notated as a group of small spectrum category note heads embraced by a bracket with an extension line. The number of note heads included depends on what amount of information is necessary to understand the behaviour of the accumulation. The major difference from SASO is how the placement over the spectrum staff system affects the positioning of the individual sounds included in the accumulation. Figure 7 shows three examples of accumulations. A represents a horde of very short complex sounds, B consists of slightly longer pitched sounds. C also has a random indicator specified for the vertical axis, in this case representing 100 % random positions of the individual sound particles. The extension lines are positioned at resulting spectral centroid frequency of the accumulation. Spectral width is indicated in the same manner as individual sounds, but for the whole accumulation.

For accumulations, levels of randomness can be indicated both for the spectrum and time axes using a percentage value and a question mark with arrows indicating the axis affected as seen in Figure 7. Introducing randomness to the description of textural sounds is in line with the findings of Grill et al where ordered-chaotic and homogenuous-heterogenuous were found to be defining characteristics relating to the perception of sound textures with over 50 % agreement among expert listeners [4].

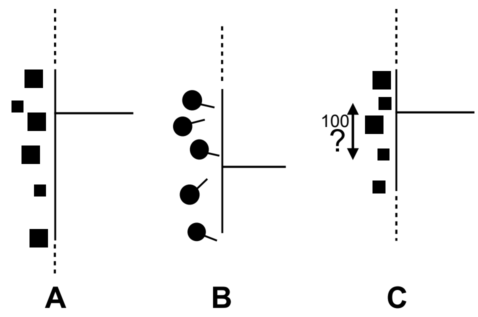


Figure 7. Three examples of accumulations–these can look very different depending on what sound objects are accumulated

2.3 Variation

Godøy and Thoresen both suggest *gait* as the English equivalent of Schaeffer's *allure* - a way of moving forward [5]. Besides granular gait covered above as granularity, SASO has indicators for variation with regard to pitch gait, dynamic gait and spectral gait. Approaching these concepts from a sound synthesis perspective, I choose to treat them as different forms of modulation with a standardised and flexible mode of representation:

2.3.1 Modulation

Modulation is change as articulation rather than structural changes of values. These can be of any kind but common in the music literature are vibrato and tremolo though these terms are not used here since they are ambiguous because of their connections to music instrument practice. A small line shape placed below the extension line of the sound component affected indicates the modulation curve with a short written label below describing the nature of the modulation. The line shape is to be interpreted as describing a change covering the full duration as indicated by the extension line under which it is placed. A small colon mark means a repeated curve/wave which would be the case for a vibrato. Further information can of course be introduced as one sees fit, e.g. a frequency value next to the colon mark specifying the speed of a repeated variation and the height of the symbols can be used to indicate various degrees of modulation. Also, one sound object can have several modulations and these may vary over time.

Since the main contour for pitch (for pitched and dystonic sounds) and centroid (for complex sounds) is indicated by the extension line of the object, modulation of these parameters are notated on the extension line. See Figure 8 for examples of modulation.

2.3.2 Amplitude envelope

The amplitude envelope of a sound is a special case of variation, indicated with a line shape thicker than the modulation shapes and is placed below the whole sound object. No text label is necessary. Figure 8 D shows an example of an amplitude envelope.

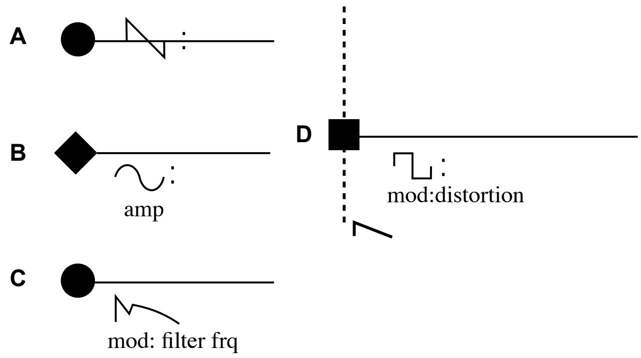


Figure 8. Examples of modulated sounds: A has a repeated sawtooth modulation of pitch, B has a sine wave modulation of amplitude, C has an envelope modulation of a filter extending over the length of the extension line, and D has a pulse wave modulation of distortion as well as an indicator of amplitude envelope at the bottom.

2.4 Rhythm

The visual representation of rhythm was covered in more detail in [6]. I recommend notating rhythm using traditional notation on separate staves below the spectrum staves so that each system of sound notation is mirrored by a layer of rhythm notation on the separate rhythm staff. Traditional notation is still the best way for communicating complex rhythmical relations in terms of note onsets. However, for the individual durations of each note I rely on the extension lines of each sound object and its components. These lines are used to indicate the duration variations treated as articulation in traditional notation, such as staccato and legato. Figure 9 shows an example of rhythm notated below the spectrum staff.

2.5 Dynamics

While the amplitude envelope can be used to define the amplitude shape of an individual sound object, we also need to address the overall dynamic development of the contents of a staff system. Depending on the notation purpose I suggest using either traditional relative dynamic notation as in the example analysis in the Addendum II of the Thoresen and Hedman paper [1], or a continuous line graph below the staff system, similar to track volume automation in Digital Audio Workstations.

2.6 Spatialisation

The representation of spatialisation is not part of SASO, and there is no common standard for notating spatialised sound [7] though there are some interesting solutions, see e.g. [8]. There is much to take into consideration when visually representing the spatial aspects of music, which I covered in greater detail in [9]. An important aspect is how a space and its characteristics can be described from different perspectives. For notating structurally significant movements and changes of position I suggest a 2D image placed above the staves displaying all numbered/labelled notated layers from a top view for horizontal movements. New images are introduced when necessary and/or whenever a layer starts or ends a trajectory, which is indicated using arrows to indicate the change to be performed until next indicator appears. As in some graphical user interfaces for surround panning, positions can either be introduced as coordinates on a cartesian coordinate system or as angles and distance related to the listener position. This 2D-notation was influenced by the work of Ellberger et al [8] [10] and Garcia et al [11].

For 3D positioning one can either introduce a front view image below the top view to account for the added dimension, or (when applicable) use a combination of colour and brightness for elevation when more exact readings of elevation are not necessary. My suggested colour scheme was inspired by the artificial horizon of airplane controls where blue and brown represent the areas above and below the horizon respectively. In my notation, symbols have brighter shades of blue as they ascend above the centre position, while they have brighter shades of brown as they descend. Figure 10 shows examples of a cartesian style representation (A), 3D positions using top and front views (B) and 3D positions using colour shades for elevation. The colour scheme itself is also represented next to the indicator (C). Figure 11 is a short example of notated movements of two numbered layers of sound notation.

Important for the notation of a musical parameter is to distinguish structural changes from elements of articulation. As with traditional musical parameters, certain spatial aspects of sound can also be considered articulations of a sound rather than positional changes, such as a sound rotating around the listener. The movement is experienced as a sound in orbit rather than a sound changing from one position to the other and should therefore be notated as a case of modulation (See Figure 12).

Sometimes sounds have their own dedicated reverb effects, functioning as resonators for those sounds rather than providing artificial room acoustics for the entire sound world. This is notated as a grey shadow behind the main extension line of the sound, reflecting the amplitude envelope of the effect. A shadow of lesser width than the spectrum category symbol (as shown in Figure 13 A) represents reverb with lower amplitude than the original sound while a reverb with the same width as the symbol (Figure 13 B) represents a reverb of equal amplitude to the original sound.

2.7 Change and transformation

What Denis Smalley introduced with spectromorphology in contrast to Schaeffer's typomorphology is a framework for describing sounds even as they change, in all their dimensions [3]. SASO accounts for this in various ways for the sake of making detailed analyses possible. Since my adaptation is supposed to work also for algorithmic composition, all individual parameters defining a sound object can be thought of as an array of one or several breakpoints. This is reflected in the continuation of spectrum category extension lines and/or dashed lines extending from the indicators subject to change. For example, in electroacous-

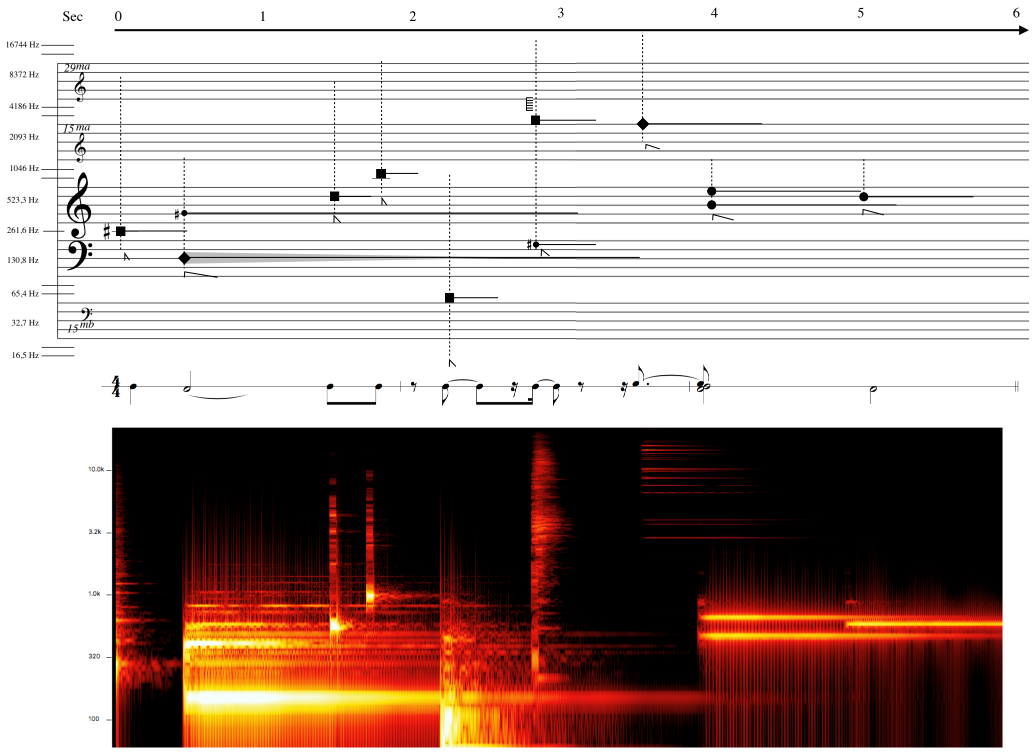


Figure 9. Short notation examples placed over a sonogram to show the correlation.

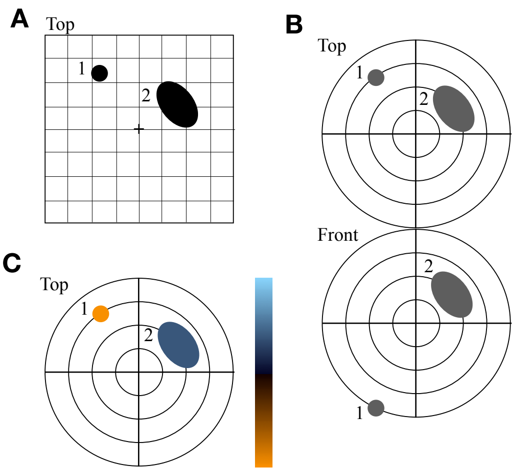


Figure 10. Examples of representation of the positions of two numbered sound layers: A) a cartesian coordinate system for horizontal placement, B) as angles relative to the centre positions in three dimensions with a top view and a front view, and C) like B but with elevation represented as colours with the colour scale shown to the right.

tic music changes to the spectral centroid and the spectral width of a sound object would be expected results of the use of automated filters which would result in dashed lines tracing the changes of width values and centroid frequency for the duration of the changes. That being said, the composer must decide what is the reasonable level of detail for notating a sound object.

2.8 Performance

For performance scores of traditional music notation, symbols are added and/or reinterpreted to accommodate the various features of each instrument. Similarly, the notation presented here can not necessarily be performed as is, but needs to be adapted and in many cases translated to action notation that makes sense for the performers and/or sound sources involved. For computer playback this means converting the notation to midi-like data to be interpreted by the computer's sound-producing software. For performances with acoustic instruments, an exploratory work may be necessary to find the actions needed to produce the sounds prescribed by the notation. Such explorations may invite the musician to take a more active part in the final design of the work. The actions can be indicated on a separate line below the sound notation in the same way that guitar tablature is often positioned below a staff of traditional notation. Transcriptions of music with unusual playing techniques as in the works of Helmut Lachenmann [12], will result in scores with major differences between the notated sounds and their indicated actions.

3. CLOSING REMARKS

What has been described here is not exhaustive but an introduction to my work with adapting Thoresen's analysis tools for composition and transcription related to the acoustic properties the system could be imagined to represent. As with any use of composition tools I expect that users of this system will make the necessary tweaks and additions to make it useful for the situation at hand. Exper-

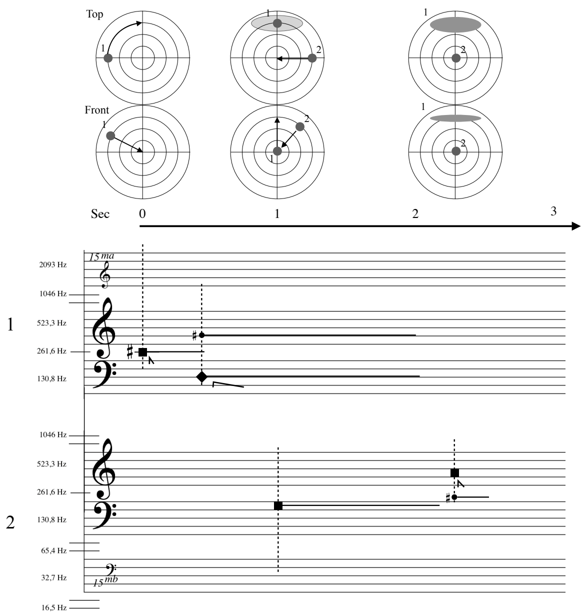


Figure 11. The notation of two numbered layers' movements

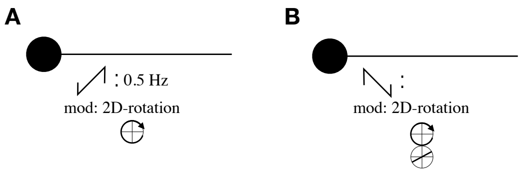


Figure 12. Spatialisation as articulation expressed as a form of modulation: A) is a clockwise steady rotation in two dimensions with one rotation every other second (0.5 Hz) while B) is a counterclockwise 2D rotation tilted in the 3D space.

imental music composers have shown to be quite inventive in terms of modifying traditional music notation in order to convey their musical ideas to musicians. Nevertheless the foundation of traditional notation remains the Western chromatic scale and their positions over a staff system constructed with diatonic scales in mind though there are plenty of suggestions for replacement systems [13].

Starting from a notation system aimed for analysis, the most important difference from notation used for composition is not the symbols themselves, but their interpretation. Neither in analysis nor composition do graphic symbols serve as a complete manifestation of the sound. But for analysis it can sometimes suffice that certain significant features of the music can be assessed, while a composed score may be the carrier of a work's identity and must contain sufficient information for its performance. A consequence of this difference in level of detail may be that the composed score needs to be broken down into different layers on several systems, while analysis scores of electroacoustic music are often displayed as one score.

There are several ways in which I will explore these ideas further, one being the development of software tools for using this notation for algorithmic composition in similar

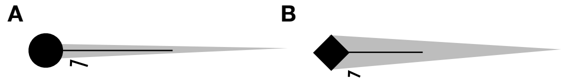


Figure 13. Two examples of reverb added to specific sounds: A) represents a reverb tail of weaker amplitude than the original pitched sound while B) represents a reverb of equal amplitude to the dystonic sound.

ways and/or in conjunction with existing solutions like the MaxScore [14] and Bach [15] libraries for Max.

I will presently compose new music with this notation system to further explore its strengths and weaknesses in relation to the music creation process together with singers, musicians and electronic devices. Also, pedagogy remains an important goal for this work. Following three years of case studies with first year bachelor students in composition at the Royal College of Music in Stockholm using this notation at different stages of its development, I have found this use of a hybrid system for pitch, inharmonic and noise components a useful and intuitive way of bringing composition students with classical and electroacoustic background together. The results of the first case study was presented at TENOR 2018 [16] and a more detailed account of the three studies will be presented in an upcoming article.

Acknowledgments

I would like to thank the anonymous reviewers for their valuable suggestions and comments. The work presented in this paper was made possible through a collaboration between The Royal Institute of Technology in Stockholm and The Royal College of Music in Stockholm.

4. REFERENCES

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

[2] P. Schaeffer, *Treatise on Musical Objects: An Essay Across Disciplines*. Univ of California Press, 2017, vol. 20.

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

[4] T. Grill, A. Flexer, and S. Cunningham, "Identification of perceptual qualities in textural sounds using the repertory grid method," in *Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound, Coimbra, Portugal*. ACM, 2011, pp. 67–74.

[5] J. Dack and C. North, "Translating pierre schaeffer: Symbolism, literature and music," in *Proceedings of EMS06 Conference, Beijing*, 2006.

[6] M. Sköld, "Visual representation of musical rhythm in relation to music technology interfaces-an overview," in *CMMR 2019: 14th International Symposium on Computer Music Multidisciplinary Research, Oct 2019, Marseille, France*, 2019, pp. 725–736.

[7] N. Peters, G. Marentakis, and S. McAdams, "Current technologies and compositional practices for spatialization: A qualitative and quantitative analysis," *Computer Music Journal*, vol. 35, no. 1, pp. 10–27, 2011.

[8] E. Ellberger, G. T. Perez, J. Schuett, G. Zoia, and L. Cavaliero, "Spatialization symbolic music notation at icst," in *Proceedings of the joint International Computer Music Conference / Sound and Music Computing conference ICMC—SMC—2014, Athens, Greece*, 2014.

[9] M. Sköld, "The visual representation of spatialisation for composition and analysis," in *Nordic Sound and Music Computing Conference, Stockholm, Sweden*, 2019, pp. 70–77.

[10] E. B. Ellberger, G. T. Pérez, L. Cavaliero, J. Schütt, G. Zoia, and B. Zimmermann, "Taxonomy and notation of spatialization," in *Proceedings of the International Conference on Technologies for Notation and Representation, Cambridge, UK*, 2016.

[11] J. Garcia, T. Carpentier, and J. Bresson, "Interactive-compositional authoring of sound spatialization," *Journal of New Music Research*, vol. 46, no. 1, pp. 74–86, 2017.

[12] D. Alberman, "Abnormal playing techniques in the string quartets of helmut lachenmann," *Contemporary Music Review*, vol. 24, no. 1, pp. 39–51, 2005.

[13] "The music notation project," http://musicnotation.org, accessed: 2019-12-30.

[14] G. Hajdu and N. Didkovsky, "Maxscore: Current state of the art." in *International Computer Music Conference, Ljubljana, Slovenia*, 2012, pp. 156–162.

[15] A. Agostini and D. Ghisi, "Bach : an Environment for Computer-Aided Composition in Max," in *Proceedings of the International Computer Music Conference*, Ljubjana, Slovenia, 2012, pp. 373–378.

[16] M. Sköld, "Combining sound-and pitch-based notation for teaching and composition," in *TENOR'18–Fourth International Conference on Technologies for Music Notation and Representation, Montreal, Canada*, 2018, pp. 1–6.

GEOMETRIC NOTATIONFOR TIME-BRACKET WORKS, APPLICATIONS AND PERFORMANCE: THE CASE OF JOHN CAGE'S *MUSIC FOR___*

Mikhail Malt
Ircam - STMS
mikhail.malt@ircam.fr

Benny Sluchin
Ircam
benny.sluchin@ircam.fr

ABSTRACT

The interpreter who approaches the music of John Cage composed after the middle of the 20th century is often disconcerted by a great freedom of execution, associated with a set of precise instructions. In previous work [8] we modeled these time brackets (TB) by parallelograms to build computer interfaces for interpretation assistance in the context of Cage's *Two*[5]. Over time ([9], [10], [11], [13]), we realized that the shape used to represent TB, brought important information for the interpretation and musical analysis. In this paper we apply previous research to a computer display conception of John Cage's *Music for___* (1984-87).

1. INTRODUCTION

The interpreter who approaches the music of John Cage composed after the middle of the 20th century is often disconcerted by a great freedom of execution, associated with a set of precise instructions. The result is that, each time, the musician is led to determine "a version," and to decide on a choice among the free elements proposed by the piece. A fixed score is thus created, which can be used several times. The musician interprets "his version" while thinking that it conforms to the composer's intentions. But in fact, most works of Cage composed after the 1950s should not be preconceived, prepared, "pre-generated" for several executions. Each interpretation should be unique and "undetermined." It is in this sense that the use of the computer can help the performer: a program will allow the latter to discover without being able to anticipate what and when he plays. The performance of the work thus escapes the intention of the musician to organize the musical text.

2. OVERVIEW OF EARLIER RESEARCH

In previous work [8] we modeled these time brackets (TB) by parallelograms to build computer interfaces for interpretation assistance in the context of Cage's *Two*[5]. Over time ([9], [10], [11], [13]), we realized that the shape used to represent TB, brought important information for the interpretation and musical analysis. The unusually long duration of this piece, 40 minutes, and the use of TB shows that the temporal question, and its representation, is essential in the *Number Pieces*.
The first step in the process was to model a graphic representation of each part as a succession of musical events in time. For this purpose, the temporal structure of the piece has been represented as quadruples on a timeline. $(s_l(k),\ s_u(k),\ e_l(k), e_u(k)\ )$.

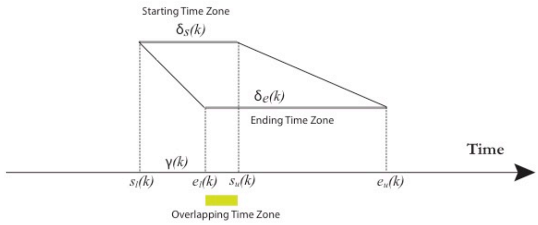


Figure 1. Graphic representation for a generic time event

To obtain a graphic representation of each event in time we consider the quadruple: $(s_l(k),\ s_u(k),\ e_l(k), e_u(k))$ where $(s_l(k),\ s_u(k))$ is the *Starting Time Zone* and $(e_l(k),\ e_u(k))$, the *Ending Time Zone*. As the two intervals have, in our case, a designed superposition, we prefer to distinguish starting and ending zones by using two parallel lines (Figure 1). In this representation we define the "overlapping time zone" the value $s_u(k) - e_l(k)$.
The graphic event obtained by connecting the four points has a quadrilateral shape. The height has no particular meaning. The *starting duration* $\delta_s(k)$ is defined as the difference $(s_u(k) - s_l(k))$, which is the time span the performer has to start the event. In the same way the *ending duration* $\delta_e(k)$ will be the time span given to end the event $(e_u(k) - e_l(k))$. In the general case, these values are not the same, and the form we get

is asymmetrical. When dealing with Cage's *Number Pieces*, one generally has: $\delta_s(k) = \delta_e(k) = \delta(k)$, both durations are the same, and the figure to represent is a trapezoid (Figure 2). We call this duration $\delta(k)$, the "Cage Duration" of the event. This is the case in the majority of the corpus we are dealing with. Special cases will be mentioned later on.

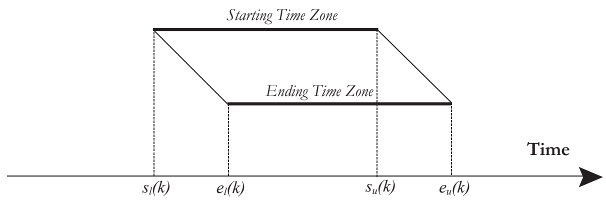


Figure 2. Graphic representation for a time event in Cage's *Number Pieces*

There is mostly an overlapping of the two time zones, $(s_l(k),\ s_u(k))$ and $(e_l(k),\ e_u(k))$ but it can happen that those are disjoined. We can define a variable $\gamma(k)$ where: $s_l(k) + \gamma(k) = e_l(k)$. In Cage's *Number Pieces*, $\gamma(k)$ depends generally on the event duration. Thus, we don't have a huge variety of forms.

An alternative way to present a quadruple will be: $(s_l\ (k),\ \delta_s(k),\ \delta_e(k),\ \gamma(k))$ where $\gamma(k)$ is the value previously discussed. This representation can easily display the regularity in the TB construction (Figure 3).

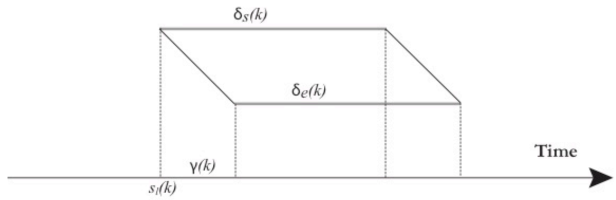


Figure 3. An event represented as $(\boldsymbol{s_l(k)},\ \boldsymbol{\delta_s(k)},\ \boldsymbol{\delta_e(k)},\ \boldsymbol{\gamma(k)})$

Concerning the placement of two contiguous events *k* and *k*+1 we can define a variable $\varepsilon(k)$, the gap between the elements *k* and *k+1* where:
$\varepsilon(k) = s_l(k+1) - e_u(k)$ (Figure 4).

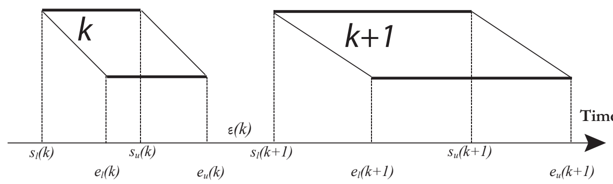


Figure 4. $\boldsymbol{\varepsilon(k)}$**,** The gap between the elements *k and k+1.*

The geometric presentation described here has been proved useful in the case of John Cage's *Number Pieces* [8]. A global view of the piece is available, and the time management, while performing, is improved. In these pieces, the TBs are filled with only few musical elements (in a majority of cases only one note).

3. *MUSIC FOR___* (1984–7)

3.1. Presentation

Between 1984 and 1987 John Cage composed a work (family of works) called *Music for___*. The principal of this (these) composition(s) is the same: musical events are spaced over a total duration of 30 minutes, using TB (a technic he has also used in other works, as well as in the *Number Pieces*, the works he has composed in the last period of his life).

There are 17 individual parts (flute, oboe, clarinet, trumpet, horn, trombone, 2 violins, viola, cello, 4 percussions, 2 pianos), which can be performed individually or together in any combination. The number of performers involved then completes the title. Thus, *Music for two* will be the title for any combination of two instruments from the parts (146 pieces). The principal of this construction is the same as that of *Concert for Piano and Orchestra*, an earlier work of Cage (1957–8). Versions of shorter durations can be made.

The variety of combinations that can be created, shows that we are dealing in fact with a family of works. Not only does the choice of the instruments permit a large number of realizations, but also each part individually gives the performer a lot of interpretative choices.

3.2. Brackets design and pitch material

In *Music for___* one finds two types of TB: the usual flexible ones (TB that have variable times within which the performer begins and ends playing) he calls here *Pieces*, and the fixed TB (TB that has specific start and stop times) he calls *Interludes* (Figure 5).

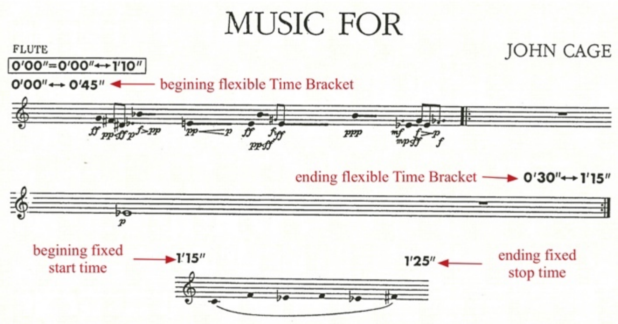


Figure 5. *Music for___*, two types of TB

Music for___ is comprised of three kinds of music. Two of which happen in the *Pieces*: repeated quiet sustained tones separated by rests, and dense spatially (proportionally) notated music characterized by a wide range of quickly shifting dynamic levels (referred later as to "A" and "B" music, respectively). The third material is the one that fills the *Interludes*, a chant-like notation free of rhythmic specificity (referred later as "C"), Figure 6.

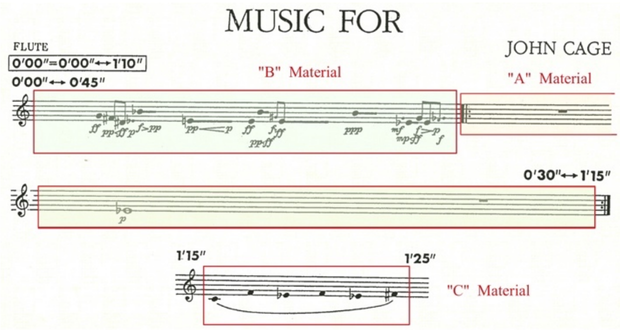


Figure 6. *Music for___*, materials used

Regardless the duration choice within TB, it is clear that a fourth category of music present must be silence (Represented as classical rest musical notation in material "A", and as proportional empty space).

The seventeen parts of *Music for__* share several characteristics. The parts use two types of TB: flexible for the *pieces*, and fixed for the *interludes*. The total number of the TB varies from part to part, as well as the type. This design is easily detected from our timeline presentation. Weisser [2] mentions the fact that in the *Number Pieces* the internal overlap (the parameter we defined as $\gamma(k)$ is closely related to it) has a proportional relationship of (1:3) to the Cage duration of an event. For example, a time bracket with duration of 60" will have a 20" of "overlapping time zone". Haskins [1] enumerates 6 types of brackets commonly occurring in Cage's *Number Pieces* lengths of it are: 15", 30", 45", 60", 75", and 90".

In the parts of *Music for__* the flexible TB falls into 4 types (Figure 7). At the time of composition Cage's sketches show hand calculations. It is only later on, with the help of the software developed by Andrew Culver [2], that the production of the *Number Pieces* became more "industrial". Observing carefully the TB used in *Music for__* shows that the durations are 30", 45", 60", and 75" (Table 1) while the "overlapping time zone" is a constant of 15". Here therefore, the inner overlap does not follow the rule of (1:3) mentioned earlier but that of (1:2), (1:3), (1:4), and (1:5).

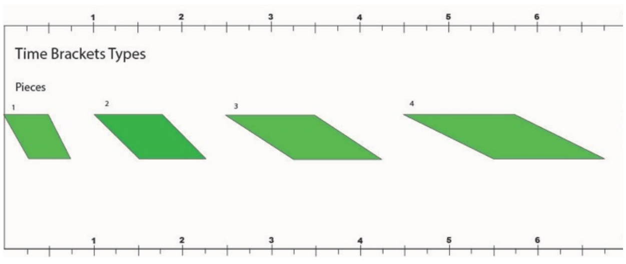


Figure 7. Geometric figures of the TB used in *Music for___*

Type	$\delta_s(k) = \delta_e(k)$	$\gamma(k)$
1	30	15
2	45	30
3	60	45
4	75	60

Table 1. Flexible TB types used in *Music for___*

The durations of the fixed TB vary between five, ten, or fifteen seconds (Figure 10). Concerning the overlap ($\varepsilon(k)$) between the events: fifteen seconds overlap occurs between successive flexible TB, but no overlap occurs between a flexible one and a fixed one.

An initial time bracket noted at the beginning of each part is in fact the time left at the end of the part, after the last event and the total duration of 30'. For example, the trombone last event (an interlude) has 29'15" as ending time. That gives a remaining 45", which will constitute the initial time bracket. This initial time bracket permits each performer to determine his own start time within this time bracket. In this way any relationship between the event's placements is dependent on this initial shift. This personal duration permits the performers to get to their placement in the hall1. The following table (Table 2) displaces the data of each of the parts.

	Initial TB	Number of TB	Flexible TB	Fixed TB
fl	0" <-> 70"	31	16	15
ob	0" <-> 40"	33	20	13
cl	0" <-> 20"	35	18	17
tpt	0" <-> 05"	35	19	16
hn	0" <-> 45"	26	17	9
tbn	0" <-> 45"	38	19	19
vl	0" <-> 20"	34	24	10
vl 2	0" <-> 15"	32	17	15
vla	0" <-> 30"	35	20	15
vc	0" <-> 30"	38	19	19
pn 1	0" <-> 35"	36	18	18
pn 2	0" <-> 70"	28	19	9
perc I	0" <-> 80"	32	19	13
perc II	0" <-> 05"	32	23	9
perc III	0" <-> 00"	30	18	12
perc IV	0" <-> 20"	35	23	12
voice	0" <-> 20"	38	21	17

Table 2. Data for the individual parts of *Music for___*

3.3. Time brackets graphical representation

In an analog way to our previous work, we generate (Figure 8) the timeline of each individual part with the geometric figures (parallelograms for the regular TB and straight lines for the fixed ones).

[1] …They are then to be played as though from different points in space. The players may sit anywhere within the auditorium with respect to the audience and to each other. *Music for___*, instructions for the players in all individual parts.

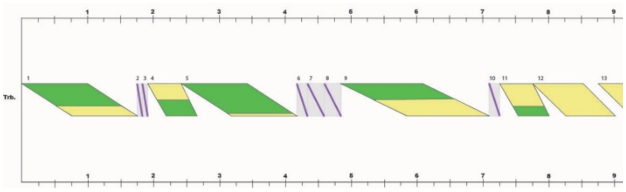


Figure 8. Timeline of the trombone part, first 8 minutes

In this case the different materials are coded by color: yellow for the “A” material (long held tones eventually repeated), green for the virtuosic one “B”, and the slanted violet lines for the “C” material of the Interludes. The graphic proportions of the first two materials are respected in the sharing of the figure. Thus, before approaching the element 12 of the trombone part (Figure 4), the musician is aware of the fact that it is composed only of long tones, and the starting point can be chosen later than that of the left time bracket. On the contrary, element 5 (Cage’s duration of 30”) contains approximately half of virtuosic material, and choosing the starting point early situated will enable ending in a less panic way.

As with the earlier interfaces we have proposed, this timeline presentation helps time management. This aspect is more vital in this case as the events are densely filled with material (in most of the number pieces, the events contain only one note) and also the gap between the elements (the parameter we have defined as $\varepsilon(k)$) is sometimes positive, creating a lapse of time of a guaranteed silence. In the Music for___ parts, this parameter is either =0 in the case of Interludes and <0 in the case of the pieces.

3.4. Parameters deduced from representation

So, one concludes that in the *Music for___* parts the fourth element (silence) will have smaller rate in comparison to the average *Number pieces*. For this reason, we represent a density factor for the green material (“B”). This is a positive number giving the ratio of the number of notes to be performed to the proportional time of that figure.

For example, element 5 shown here (Figure 9) will have the number 10 associated with relatively large proportion of the green material (86%). The Cage duration of this element is 60 seconds we obtain: 10: (0.86x60) = 0.19.

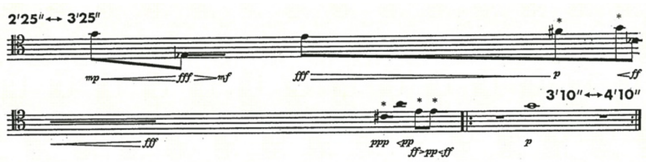


Figure 9. Event 5, in trombone part.

While the preceding element 4 (Figure 10) displays the number 29 associated with a lower proportion of green (44%). As the nominal Cage duration for this event is 30” we obtain the density factor to be 2.19.

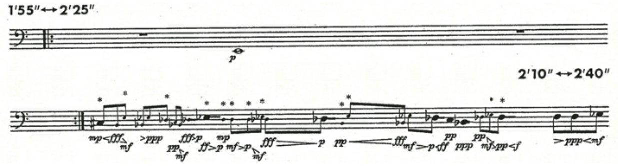


Figure 10. Event 4, in trombone part

In all the parts of *Music for___* (except for the vocal part), the *pieces* are written on two systems. For this reason, it is hard to estimate, for the musician, the time given, and the ease to perform.

In analogue way for the *interludes*, we calculate the ratio between the number of notes to duration for the element (5, 10, or 15 seconds).

For example, for element 21 (Figure 11) we get the small value of 0.2 (there is one note for 5”). This is not the smallest one can encounter, as we get 0.06 for element 36 (one note for a 15” *interlude*), and 0 for element 37 (Figure 12).

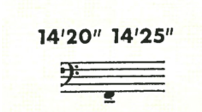


Figure 11. Event 21, in trombone part

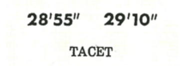


Figure 12. Event 37, in trombone part

A higher value occurs when we have a short interlude containing many notes, as in event 3 (Figure 13). Here the density factor is 1.6.

Figure 13. Event 3, in trombone part

These factors are incorporated in the data we compile for each part: a table in which the temporal data is given in seconds (Table 3). In the left part we display the TB data in an equivalent way ($(s_l(k)$, $\delta_s(k)$, $\delta_e(k)$, $\gamma(k))$ and $\varepsilon(k)$, see Figure 3 and Figure 4), with both Cage’s durations (which are almost exclusively equal[2]).

In the last three columns, T1 and T2 display the materials (“A”, “B” and “C”) that composes each event, and the last column displays the corresponding density parameter, $\rho(k)$. By its nature the material “A”, one held repeated tone, does not display any density parameter useful for performance.

[2] The only exception occurs in the Percussion II part, for element 27. The starting time bracket indicates 26’15”<–>27’00” (45”) while the ending time bracket is 26’45”<–>27’15” (30”). This may be a print- or calculation- mistake.

	$s_l(k)$	$\delta_s(k)$	$\delta_e(k)$	γ(k)	ε(k)	T1	T2	ρ (k)
1	0	60	60	45	0	**B**	A	0,46
2	105	0	0	5	0	**C**		0,60
3	110	0	0	5	0	**C**		1,60
4	115	30	30	15	0	A	**B**	1,58
5	145	60	60	45	-15	**B**	A	0,19
6	250	0	0	10	0	**C**		1,10
7	260	0	0	15	0	**C**		0,20
8	275	0	0	15	0	**C**		0,40
9	290	75	75	60	0	**B**	A	0,40
10	425	0	0	10	0	**C**		0,20
11	435	30	30	15	0	A	**B**	2,03
12	465	45	45	30	-15	A		–
13	525	45	45	30	-15	A		–
14	600	0	0	10	0	**C**		0,80
15	610	60	60	45	0	A		–
16	715	0	0	10	0	**C**		0,30
17	725	45	45	30	0	**B**		0,42
18	800	0	0	10	0	**C**		0,20
19	810	30	30	15	0	**B**	A	1,40
20	855	0	0	5	0	**C**		1,00
21	860	0	0	5	0	**C**		0,20
22	865	45	45	30	0	A		–
23	940	0	0	10	0	**C**		0,70
24	950	45	45	30	0	**B**		0,71
25	1010	75	75	60	-15	A	B	0,69
26	1130	45	45	30	-15	A	B	0,48
27	1205	0	0	15	0	**C**		0,20
28	1220	0	0	5	0	**C**		0,60
29	1225	60	60	45	0	**B**	A	0,02
30	1330	0	0	10	0	**C**		0,60
31	1340	60	60	45	0	**B**		0,33
32	1445	60	60	45	0	**B**		0,75
33	1535	60	60	45	-15	**B**	A	0,38
34	1640	0	0	5	0	**C**		0,80
35	1645	45	45	30	0	**B**		0,58
36	1720	0	0	15	0	**C**		0,07
37	1735	0	0	15	0	**C**		0,00
38	1750	0	0	5	0	**C**		1,20

Table 3. Numerical data for trombone part in *Music for___*

4. GEOMETRICAL FORM AND PARTS PROOFING

We have been using the graphical time line of Cage's Number pieces, to notate the part in a different space, different from the printed page. The events are prepared as score stripes (the events) and are displayed in connection with the cursor and his advancement in the timeline.

In the case of Music for___, this formal form permits the detection of anomalies. Playing the part of the original presentation, and the display of the TB as numerical data on the timeline, does not permit to grasp particularities, and especially when overlap occurs. For example, when we perform the events 25, 26, and 27 in the viola part (Figure 14. **Events 24, 25, 26, and 27 from the viola part, *Music for___***) we simply have difficulties to use all the material. But seeing the form (Figure 15) explains the origin of this difficulty. Not only the element 26 has an "abnormal" form (Cage wouldn't have dared bother the musician, while his instructions show respect) the overlapping created makes the passage very hard to play.

Figure 14. Events 24, 25, 26, and 27 from the viola part, *Music for___*

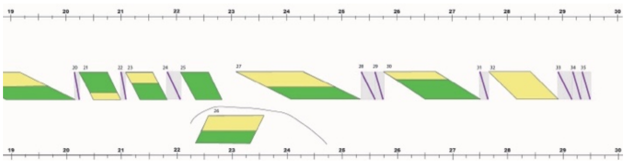


Figure 15. Timeline of the viola part, last 10'

A proper solution will be changing the duration of event 25 (whose density is quite high) and disregard the abnormal event 26.

As another example one has the violin 1 part, where different impossibilities occur (Figure 13). Event 6 overlaps with event 5, event 8 with event 7. Here a solution should be found before the performance. One could simply omit events 6, and 7 or generate more elaborate solutions.

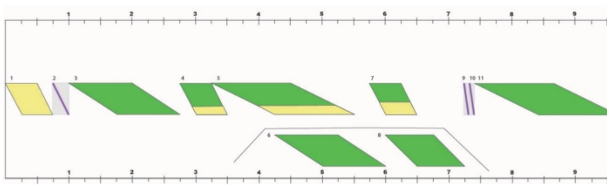


Figure 16. Timeline of the violin 1 part, the first 8 minutes

5. DISPLAYING THE DENSITY FACTOR

The density factor is rather graphically displayed. The digital information is hardly perceived during performance. The eye grasps a tendency without having to get the right number involved.

As a matter of fact, for the moment we display the density factor (only for B material) as saturation parameter for the filling color. This is a display option for the musician. There are different ways to take advantage of this knowledge and to anticipate the handling of a musical event while performing. Purely filled pieces ("A" or "B" type) generally won't create performance conflict. They could be handled by choosing an adequate starting time, taking advantage of the TB. In compound pieces, ("AB" or "BA" type) one could use the information on density in order to anticipate, manage time and resources for the performance. The parameter enhances the preparation of the part.

It is interesting to note that John Cage, consulted different performers and their knowledge when composing the parts. But he also gave the procedure for preparing the parts for performance:

Each player should prepare his part by himself and learn to play it with his own chronometer. There should be no joint rehearsal until all parts have been carefully prepared. [14, instructions for the players]

6. CONCLUSIONS

At the present time we work to offer the musicians a way to approach other pieces from the same family, constructing a generic interface. The task may be somewhat complicated. The works called Number Pieces, share the same principal described earlier, but often contain particularities and exceptions in the instructions for performance. The interface then has to be adapted to cover these.

The interface is a substitute to the printed score. It reveals the structure of the work and provides the performer with the tool to achieve the "meditative concentration" needed. The few instructions given by Cage are integrated in the interface.

Considering the graphic representation, we presented above, our main goal was to find geometric properties and strategies to enhance the performance of these pieces through computer interfaces. John Cage's works have been the target of our work for several years now. We have developed computer tools for the interface, and used it in practice. Both concerts and recordings have been the tests for the usefulness of the approach towards performance. The modeling process is transformed in a pragmatic analysis of the musical phenomena that leads us, step by step, to model some of Cage's concepts. Mentioning first the Concert for Piano and Orchestra (1957), an earlier work that has become important step of his output [7]. Followed by two of his number pieces for a small number of performers [8]. These works were also the object of a recording and performance sessions ([9], [10], [11]).

7. REFERENCES

[1] R. Haskins, *The Number Pieces of John Cage*. DMA dissertation, University of Rochester (2004). Published as Anarchic Societies of Sounds, VDM Verlag, 2009.

[2] B. Weisser, *Notational Practice in Contemporary Music: A critique of three Compositional models*. Dissertation.

[3] J. Pritchett, *The Music of John Cage*, Cambridge University Press, Cambridge, 1993.

[4] A. Popoff, "Indeterminate Music and Probability Spaces: The Case of John Cage's Number Pieces," *MCM (Mathematics and Computing in Music) Proceedings*, pp. 220-229. Springer Verlag, 2011.

[5] A. Popoff, "John Cage's Number Pieces: The Meta-Structure of Time-Brackets and the Notion of Time," *Perspectives of New Music,* Volume 48, Number 1, pp. 65-83, 2010.

[6] J. Retallack, *Musicage: Cage muses on words, art, music*, Wesleyan university Press, p. 182 (1996).

[7] B. Sluchin, M. Malt, "Interpretation and computer assistance in John Cage's Concert for piano and Orchestra (1957-58)," *7th Sound and Music Conference* (SMC 2010), Barcelona, 2010.

[8] B. Sluchin, M. Malt, "A computer aided interpretation interface for John Cage's number piece Two5," *Actes des Journées d'Informatique Musicale* (JIM'12), pp. 211-218, Namur, Belgium, 2012.

[9] J. Cage, *Two5*. On John Cage *Two5* [CD]. Ut Performance, 2013.

[10] J. Cage, *Music for Two*. On John Cage, *Music for Two* [CD]. Ut Performance, 2014.

[11] J. Cage, *Ryoanji*. On John Cage, *Ryoanji* [CD]. Ut Performance, 2017.

[12] S. Rappaport, B. Sluchin, *On Panorama* [CD]. Liner notes. Ut Performance, 2019.

[13] B. Sluchin,M. Malt, "John Cage's Number Pieces, a geometric interpretation of "time brackets" notation," *14th International Symposium on Computer Music Multidisciplinary Research (CMMR 2019)*, Marseille, 2019.

[14] J. Cage, *Music for___*, © 1984 by Henmar Press Inc., C.F. Peters Corporation, 1984.

PERFORMING THE PRACTICE OF COMPOSITION

Alison Isadora
Victoria University of Wellington,
Aotearoa/New Zealand
alisador@xs4all.nl

ABSTRACT

In this paper I discuss challenges towards our understanding of the role and ontology of the score, in relationship to the roles of composer, performer, audience and performance space. I consider the role of these actants within three situated aspects of the Western art music tradition as proposed by Coessens and colleagues: the ecological, epistemic and social [1].

What emerges from this deliberation, is the reification of the score as the 'work', the expectation of a 'genius' (male) composer, hierarchic and stultifying conditions for both musicians and audience members, and performance spaces that encourage these stratifications.

Modes of engagement are explored that might foster alternative roles for all actants and the notions of sympoiesis, and the anarchive are presented as potentially useful conceptual tools when imagining an alternative ontology of the score. Moreover, developing on Isabelle Stengers' ideas on an ecology and interdependence of practices I speculate on the ramifications of considering the score as having a 'challenging and fostering' role in relationship to the other parties [2, p.190].

The paper finishes with a discussion of *Together#5.1* in which methods for encouraging a social technology of belonging and shared compositional response-ability between all actants are explored. These methods include collective listening practices, audience scores, adaptive notations and context specific elements.

1. PERFORMING AN ECOLOGY OF A COMPOSITION PRACTICE.

(*This paper will be accompanied by a power point*).

My current research is concerned with **Performing an Ecology of a Composition Practice,** specifically within the confines of contemporary Western art music, or what Bhagwati would call eurological music. In this paper I will discuss challenges towards our understanding of the role and ontology of the score, in relationship to the roles of composer, performer, audience and performance space, before contextualising this through one of my recent works.

My use of the term 'ecology' is informed by Isabelle Stengers' proposals on the ecology and interdependence of practices, and resonates with my interests in the fostering of an inclusive compositional practice. Where natural ecologists approach a practice as it *is*, Stengers' interest lies in what it can *become.* This seems appropriate in a research concerned with the possible creation of new paradigms rather than the continuation of existing ones.

Developing on Brian Massumi's ideas of the 'social technology of belonging,' Stengers offers us a philosophical tool that can be useful in understanding the 'challenging and fostering' role that the score may have in relationship to the other parties [2, p.192]. Stengers identifies the social technology of belonging as being one that 'can and must address people from the point of view of what they may become able to do and think and feel because they belong.' [2, p.190]

What might the ramifications of an ecology of a composition practice be for the actants? Before we speculate on these possibilities, it is perhaps helpful to cast an eye on where we are now, our current situation, while recognising that the concept of a homogenous 'we' is potentially fraught.

2. SITUATEDNESS

Coessens and colleagues use Willem J. Clancey's definition of situatedness: "Where you are, when you do, what you do, matters" and make a distinction between three different aspects of situatedness: ecological, epistemic and social [1, p.47].[1] I would like to briefly consider the conditions of contemporary Western art music through this triple lens of situatedness.

Contemporary Western art music primarily emerges out of institutions that also teach the Western 'classical' music tradition.[2] Most musicians and composers involved in new music have studied and performed within this heritage. Furthermore, audiences for contemporary music concerts often consist of people who also attend historic Western art music concerts. It is not surprising therefore that codes of relationship in a new music context are closely con-nected to their historical relative. These codes determine not only how composers, performers and listeners position themselves and how they relate to each other but also their position in relationship to the score.

The score itself is not a neutral object. Bhagwati's insight into the inherent limitations of notational systems and his uncovering of 'notational bias', encourages a situated reading of scores and raises questions regarding the (often) unconscious perspective of notation in a compositional practice [3].

3. ONTOLOGICAL ISSUES

Historically in Western art music the score mediated the relationship between composer, performer and audience. The ontology of the 'work', the reification of the score, traditionally encouraged a clear delineation between composing and performing roles and resulted in the idea of a hierarchic flow of information from composer via the score to the performer and finally audience [4].

Musicologists such as Goehr and Durkin and the philosopher Benson argue, albeit from different vantage points, that the focus on 'works' rather than performances does not acknowledge the creative input from the performer in the artistic process. Furthermore, such a model is unlikely to allow for a potential creative function for the audience or a recognition of contextual situatedness. It is my suggestion that in order to explore the potential of a new music composition practice, it may be useful to 'transversally' explore connections between theories from Philosophy and Performance Studies and practices from modern theatre and dance that enable dynamic, creative roles for the composer, score, performer, audience and performance space.

Numerous musical ontologies have been developed over the years, mostly focusing on various levels of agency between the 'work', score and performance. I suggest that these ontologies are themselves often contextually related and that we may have to accept that not one ontology fits all. Furthermore, our interest in a speculative composition practice invites a reinterpretation of ontological roles.

So, rather than the reification of the score as the 'work', the expectation of the 'genius' (male) composer, hierarchic and stultifying conditions for musicians, physical separation of the audience and performers, and performance spaces that encourage these stratifications, I wonder about the role of:

- the entangled composer who co-ordinates, initiates, and acts as caretaker,
- the implicated musician who co-creates and performs,
- the agential audience who may be an active listener, participant, and co-creator,
- the situated performance space, and
- a recontextualized score which operates more as a script and can be interpreted and adapted.

Where does the role of the composer finish and the role of the performer begin? Is there overlap, or are we looking at a dynamic intertwining of roles? Can we address the audience through the technology of belonging and what role can the score inhabit in this entangled landscape?

4. THE ACT OF LISTENING

One of the things I am suggesting is that the act of listening is at the heart of a composition practice. Christopher Small coined the verb 'musicking' to describe the process of mu-sic making [5]. Listening, is also, he argues, part of musick-ing. Salomé Voegelin elaborates further when she suggests that:

> 'Listening has an exploratory capacity that does not seek to know *about* the world but approaches learning as a practice, as a physical and continuous effort to understand momentarily and al-ways again how to live in the bet ween-of-things.' [6]

When we learn an instrument, we start by learning through imitation, through what Denis Smalley would call transmodal perception, an interaction of different senses [7]. Sound, however, will always be the touchstone by which we compare our effort with the original. In the West, we often follow on by learning through the interface of written music – the score. However, in many traditions, and also in most beat-driven music, reading is not part of the equation.

Many years ago, I had the great fortune to learn rebab, the Indonesian bowed string instrument, from a master

[1] Quoting William J.Clancey. <https://openair.rgu.ac.uk> (10/06/2018).

[2] My use of the term classical is not in reference to the specific period of Classical music (ca.1750-1820) but refers to the more generic usage implying the entire tradition of Western art music.

musician. The only 'problem' was that we did not have a common spoken language.

I learned both how to play the rebab and also much repertoire, entirely by imitation. At first all I was trying to do was reproduce what he played, later on I started recognizing patterns of similarities, differences and consequences. The one note that was always performed microtonally sharp because the rebab played it at the same moment as the whole gamelan orchestra was also playing that note and it would not be heard as a distinct voice unless played slightly sharper than the rest, giving it a spectral advantage.

Where is the score in this process, where is the 'work'? As embodied and enculturated memory?

5. LISTENING IN TIME

Earlier I mentioned that I saw listening as being at the heart of a composition practice. I'd like to preface that by saying listening in time, to time, through time. And as we know from Einstein, space and time are interconnected. Which leads us to the importance of situation – space/time – in a composition practice. Content and context are intertwined.

An attention to situational specificity would seem valuable in an age of increased globalisation and I would like to think it might encourage a compositional practice that responds to context and stimulates diversity. If we recognise that the score is not 'objective' but situated, how do we respond to this, how might this influence our score-making process?

I would like to invite you to notice the situation you are in right now, the sound, light, temperature and to change your spatial situation during this talk if you feel like it.

Last September I spent time in Zealandia, a bird sanctuary the size of the city of Amsterdam, that can be found in a valley within the hills of Wellington city, surrounded by suburbs. Zealandia has a 500 year plan, the time it takes a rimu, an endemic tree, to mature. The rain forest in the valley will only fully be adult and sustainable after 500 years.

Within the sanctuary I hear both native birds I've never heard before, but also sparrows and blackbirds, what we in Aotearoa call exotics, introduced by English colonials. And there's a plane overhead, flying to Australia, and the ubiquitous sound of suburban NZ – the motor-mower. We're still very much in the city, but the audio balance has been changed. Our listening incorporates an acknowledgement of the human influence, while we hear a whisper, a tantalising breath of how this valley was and what it will hopefully one day again become. We listen in time.

We listen situationally, historically, geologically, mythologically, musically, bodily. Can we hold different ways of listening in our attention at the same time? And what might this mean for the type of score that we create? Furthermore, how do we create scores that are contextually responsive and that encourage a recognition of agency between all the actants?

6. SPECULATIVE ONTOLOGIES

Nicholas Cook proposes a shift to seeing scores as 'scripts in response to which social relationships are enacted' rather than 'texts within which social relationships are encoded' and I would like to extrapolate on this to also include the score as script wherein musical and extra-musical contributions from the musicians and audience are integrated and where different forms of listening may be en-acted [8, p.212]. In such an environment the score/ script has the ca-pacity to intra-act with the other actants, to create a com-plex web of connections. As Yolande Harris says, 'beyond theorizing the score in terms of notation, much can be learned from reconceptualizing the score as *relationship'*.

I would like to suggest an ontology of the score that em-braces an entangled *agential realism*.[3] Who and what is implicated in the different processes inherent in a score – the creation, the performance and the documentation pro-cess?

In this context, the notions of sympoiesis, and the anarchive might provide useful conceptual tools. Sympoiesis was first coined by environmentalist Beth Dempster and developed by Donna Haraway. Where Fischer-Lichte describes auto-poietic relationships between the performer and the audience, the 'feedback loops' present to some extent in all performance situations, Haraway suggests that nothing is completely self-organising, and that sympoiesis 'enfolds autopoiesis and generatively unfurls and extends it.'[10, p.58] This model of 'making with' seems useful when thinking about the ecology of a composition practice and the possible relationships between the various actants.

The notion of the anarchive can, I suggest, expand our understanding of the role of documentation. Massumi refers to an anarchive as a '*repertory of traces* of collaborative research-creation events. The traces are not inert, but are carriers of potential. They are reactivatable, and their reactivation helps trigger a new event which continues the creative process from which they came, but in a new iteration.' [11, p.6] I would suggest this could be a useful way of con-sidering the score after a performance – containing the possibility to include the embodied memory of the musi-cians as part of the anarchive and acknowledging the po-tential for situational adaptation in a score.

7. BELONGING

Let me now briefly focus on *Together#5.1*, one iteration in a series of works which explores potential creative relationships between the composer, musician, audience, performance situation and score. I will briefly outline how I hope this work encourages a social technology of belonging, new ways of listening and shared compositional responsibility. Karen Barad notes that 'Responsibility, … is a matter of the ability to respond. Listening for the response of the other and an obligation to be responsive to

[3] As developed by Karen Barad, see for instance interview in [9].

the other, who is not entirely separate from what we call self.'[9, p.70]

Nestled within the score of *Together#5.1* are in fact four distinct layers of script: for the composer, musicians, audience and 'lay' people.

7.1. The Composer

The composer has a number of tasks to fulfil both before and during the performance. Firstly, she is required to make a field recording from outside the specific performance space, played back over a localised speaker during the first part of the concert. Secondly, she gives the audience their score before they enter the performance space and instructs them how to proceed. Finally, after the performance, she invites the audience to take part in a reflection, which may be a writing process or a discussion in small groups, depending on the specific context of the performance. I consider this reflection process to be part of the work and the oral transmission by the composer to also be part of the score.

7.2 The Musicians

Although a score of *Together#5.1* exists, it is of little use to the musicians, as the vertical relations during the piece are defined by each person's own heartbeat. In this sense, the work is a series of simultaneously played parts, with moments of alignment. The musical material is not demanding but the detailed written text requires close reading in order to negotiate the work. Furthermore, the musicians are entrusted with creating situational texts relating to the performance space. These could be of an anecdotal, geological, historical or pre-colonial nature. The musicians are requested to reflect on the dialogue between the order and content of these texts and the musical material. *Together* encourages a constant interplay of the musicians' attention to their own pulse, to the audio around them and to the form, which they create together.

7.3 The Audience

The reading process for the audience begins before they enter the hall. The composer offers them a folded paper containing text and images which invites them to participate, firstly by exploring the sonic space of the performance hall, then by following their own heartbeat while attending to the music. During the piece, the audience receive musical, graphic and textual cues to proceed through their score, including an invitation to hum and later sing and finally to follow a musician in a gradual collective decrescendo. In *Together#5.1* the audience are given opportunities to create relationships between the embodied rhythm of their own heartbeat and those of the musicians around them, to listen and respond to an entangled co-creation. Could we call this diffractive listening? Listening to a performance through the pulse of one own's body? To paraphrase Barad, listening to patterns of difference that make a difference.

7.4. 'Lay' People

An extra layer of agency occurs in the iteration *Together#5.1.* A role intertwined between the audience and the musicians. These are 'lay' people (whom Cardew might call 'musical innocents') who spend an hour with me before the performance, learning their score. Music-reading skills are not required, their script is a sheet of text instructions giving cues to navigate the work. This group function as both extra sound sources in the work and as support for audience participation.

7.5. The Space

As mentioned above, the space is also addressed. How does the space we are in influence our perception of the work? Where do we situate ourselves in the space as a musician/audience member/participant?

8. CONCLUSION

In *Together#5* we consider the shared connection we have through our heartbeat and are encouraged to explore listening both to the other and ourselves. What if we decide we are all in this together? This search for a shared response-ability between all agents can I believe have both political and sonic consequences.

I'd like to end with a quote from Isabelle Stengers:

'The problem for each practice is how to foster its own force, make present what causes practitioners to think and feel and act....which may also produce experimental togetherness amongst practices, a dynamics of pragmatic learning of what works and how. This is the kind of active, fostering 'milieu' that practices need in order to answer challenges and experiment changes, that is, to unfold their own force.' [2, p.195].

9. REFERENCES

[1] K. Coessens, D. Crispin, and A. Douglas, *The Artistic Turn. A Manifesto.* Leuven University Press, 2009.

[2] I. Stengers, "Introductory Notes on an Ecology of Practices", *Cultural Studies Review*, vol 11, no.1, pp. 183-196, 2005.

[3] S. Bhagwati, "Notational Perspective and Comprovisation", in *Sound & Score: essays on sound, score and notation,* ed. P. de Assis, W. Brooks, and K. Coessens. Leuven University Press, 2013, pp. 165-177.

[4] A. Durkin, *Decomposition A Music Manifesto*, Pantheon Books, 2014.

[5] C. Small, *Musicking. The Meanings of Performing and Listening,* Wesleyan University Press, 1998.

[6] S. Voegelin, *The Political Possibility of Sound*, Bloomsbury Academic, 2019.

[7] D. Smalley, “Space-form and the acousmatic image”, *Organised Sound*, vol.12, no.1, pp. 35-38, 2007.

[8] *Study of Music. A Critical Introduction,* ed. M. Clayton, T. Herbert and R. Middleton. Routledge, pp. 204-214, 2003.

[9] K. Barad, interview in *New Materialism: Interviews & Cartographies,* R. Dolphijn and I. van der Tuin. Open Humanities Press, pp.48-70, 2012.

[10] D. J. Haraway, *Staying with the Trouble. Making Kin in the Chtulucene*. Duke University Press, 2016.

[11] B. Massumi, “Working Principles”, in *The Go-To How-To Book of Anarchiving,* ed. A. Murphie. The SenseLab, 2016.

NOTATIONAL STRATEGIES FOR INTEGRATING LIVE PERFORMERS WITH COMPLEX SOUNDS AND ENVIRONMENTS

Lindsay Vickery
Edith Cowan University
l.vickery@ecu.edu.au

ABSTRACT

This paper describes strategies for integrating live performers with complex "extra-musical" sounds and environments through extended traditional and proportional notations. The subjects of the works discussed include Animals (*wardang* [2019], *kurui* [2018]) and environments (*rising water* [2018], *willsons downfall* [2018], *njookenbooro* [2018]) . The techniques include spectrographic transcription, audio processing, extended forms of notation and spatial audio.

1. INTRODUCTION

The work discussed here draws from the rich history of recording and field recording in particular. The capability and aspiration to compose with complex "extra-musical" [1] sounds found in such recordings has closely tracked the steady advances in recording technology. From the first recorded animal recording by Ludwig Koch in 1889 [2], through the emergence of Musique Concrete in which "sound recordings were raised to the status of compositional raw material" [3] and the emergence of "field recording" practice advanced through the work of Westerkamp (1946-) and Truax (1947-) and the Vancouver World Soundscape Project (1972-) at Simon Fraser University [4]. The implications of this evolving practice have proliferated supporting the emergence of numerous associated compositional specialisations, from "virtual environments" [5] to "zoomusicology"[6].

This paper describes work in one of these specialisations: the development of strategies allow for instrumental performers to emulate and interact with complex sounds and environments through visualisation and musical notation. It aims to build on the work of composers such as Robert Erickson (1917-97), François-Bernard Mâche (1935), R. Murray Schafer (1933), Barry Truax (1947), Anne LeBaron (1953), David Dunn (1953), Michael Pisaro (1961), Matthew Burtner (1970) and Joanna Bailie (1973).

The subjects of the works discussed include Animals (*wardang* [2019], *kurui* [2018]) and environments (*rising water* [2018], *willsons downfall* [2018], *njookebooro* [2018]) . While visualization and notation attend to the moment to moment emergence of sound, this paper also discusses the interconnected issue of the shaping or absence of shaping structure in these works.

2. CONTEXT

This work attempts to take into account López' criticism of filed recording as " something perverse (…) as if the encoding of a semiotic referent in the form of an audio description of place could ever be something other than a human invention"[7], and to answer Bailie's question "if we are not to simply present the sounds of the world to an audience as a kind of musical fait accompli (...), what in fact are we to do with them?" [8]. It aims to do so by entering into a compositional and performative interaction with the recordings and in some cases the environments from which they arise.

In 2008 Dunn stated that "the dual heritage of Acoustic Ecology and experimental music—in foregrounding our aural perception of the Earth—seems more urgent than ever" [9]. At the time of writing fires are raging throughout the continent from which I write and is already estimated to have killed up to a billion animals. [10]

This devastation is the most likely the result of dissonance between the principally European and South East Asian population of Australia not just with the indigenous people but the environment as a whole, it's flora and fauna. Fires. The absolute decimation of the countryside in the south-west of Australia, where 93% of native vegetation has been "cleared" [11] and is now utterly incapable of supporting an autonomous ecosystem and reliant on human intervention to grow almost anything.

These works are intended in some part to create awareness of the natural environment including its fauna, which is often found in close in proximity to the city which for most inhabitants consists only of people, technology, buildings and roads.

The interaction with sound and the unfolding of these sounds over time in these works involves listening, analysis and compositional interaction and a similar process for the performers - learning to emulate the sounds.

In these works transcription of recordings is generally the principal basis for the score. Recordings are sometimes processed using Eric Lyon's *FFTease* objects *thresher~* or purpose built *Max* patches to emphasise principal frequencies. They are then analysed using Sonic Visualiser to create a spectrogram and sometimes *Audacity*'s "Plot Spectrum" function to identify pitch content.

Scores are generally assembled in *Adobe Illustrator*, usually in direct proportion to the spectrogram, using an image size equivalent to 8-10mm of score a second of sound. A horizontally proportional pitch grid is generally added on a separate layer (PStave) and later transposed to a traditional staff (TStave). In some cases a vertical grid is also added if a metrical tempo is required. In other cases both pitch and rhythmic components are undefined globally to allow pitch and rhythmic characteristics to be defined in the score locally. In some works a background image is assembled behind the score from "field photographs" related to the recording in others it is black allowing to maximally highlights the notation.

The score is performed in scrolling mode us-ing the Decibel Scoreplayer. In the scoreplayer formatted scores (.dsz. files) can be networked if they have the same file name and are of the same format (in this case "scrolling"), however the au-diofile embedded in any instance of the .dsz need not have the same name. Therefore the number of channels of audio is only limited to the number of iPads in the network. The ability to pair with Blue-tooth speakers provides an extremely portable op-tion for multichannel audio in site specific perfor-mances.

3. ANIMALS

The two works described here are the first of a project to compositionally interact with Australian fauna beginning with two of the country's most ubiquitous animals: the possum and the crow.

3.1. kurui [2018]

The title *kurui* was derived from the name for possum in the Gubi Gubi language (South-Eastern Queensland). The work was built around a scaffolding of 'faithful' transcriptions of 9 characteristic possum (probably Trichosurus vulpecula) sounds, bridged by "interpretive" digressions veering towards the performative (and in one case improvisational) style of the performers (Erik Griswold and Vanessa Tomlinson) – long term collaborators with the author.

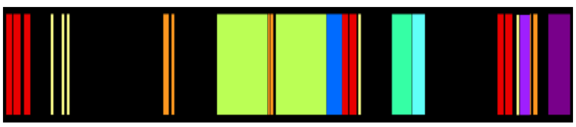


Figure 1. Proportions of the 9 transcriptions (colour coded) and digressions (in black) forming the struc-ture of *kurui*.

Figure 1 shows the proportions of the 9 transcriptions (colour coded) and digressions (in black) over the work's 8m and 32s duration.

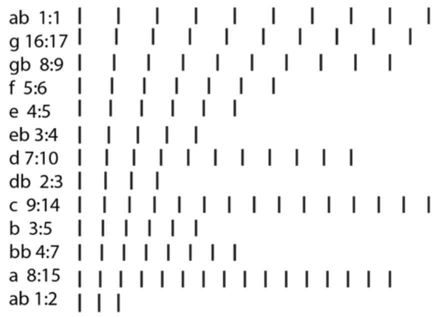


Figure 2. Grids representing proportional durations of pitches in an Ab tempo series.

The score was assembled in *Illustrator*, allowing spectrograms of the nine possum sounds to be displayed proportionally alongside the score as it was transcribed/composed. A grid layer was used to permit alternations between metrical and non-metrical material. Each vertical

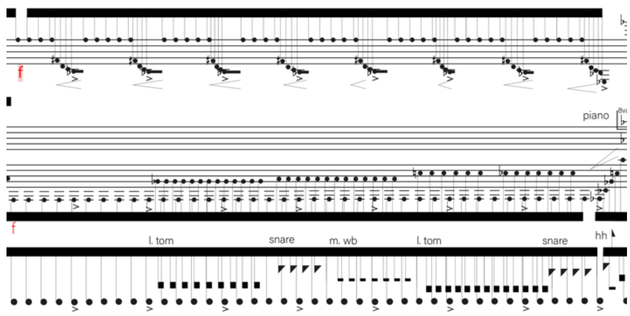


Figure 3. Polymetric digression from *kurui* [2018] for bass clari-net, melodica/piano and percussion.

grid line represented a tempo (approximately mm. 125) taken from the rhythmic grunts of the possum (shown in Figure 4). Secondary grids representing the proportional durations of pitches in a tempo series [12] (Figure 2.) were then used to create localized polyrhythms in the notation (Figure 3.).

Despite the "noisiness" of the possum sounds - significant pitch components are also evident. The contours of prominent pitched components were initially sketched directly onto the spectrogram and then "lifted-off" and adapted into a more traditionally spaced notational framework – Figure 4 is an example.

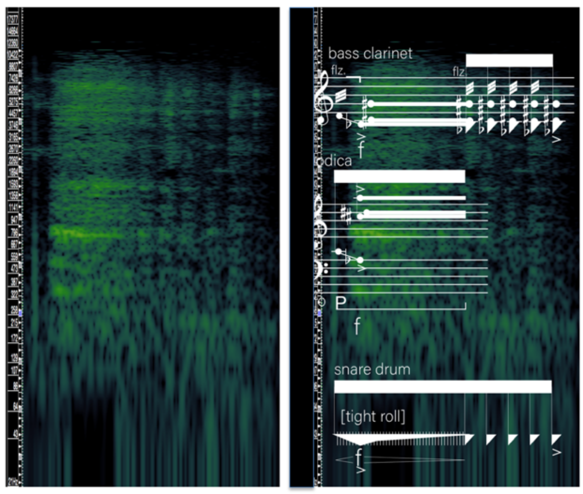


Figure 4. The opening motive of *kurui* [2018] spectrogram of Possum sound (left) and a superimposed transcription (right).

The characteristic screech emitted by the possum was transcribed for melodica and bass clarinet multi-phonic (Figure 5).

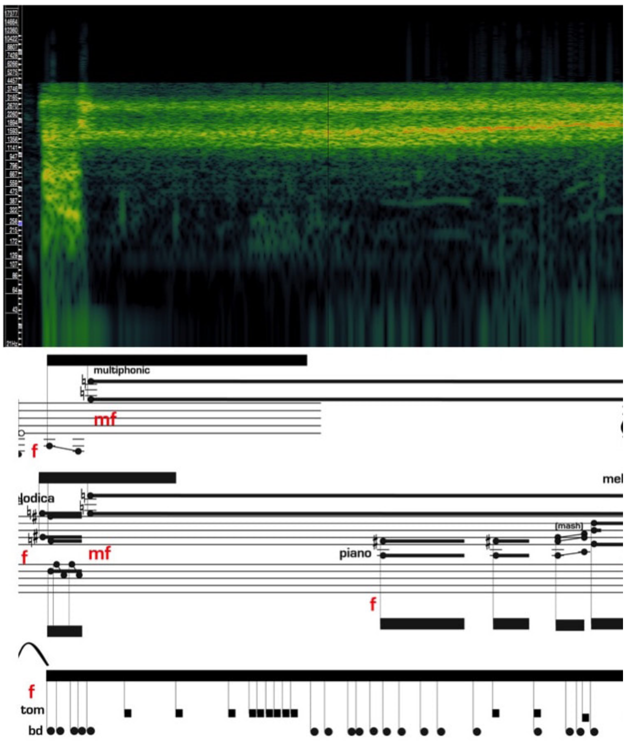


Figure 5. Possum Screech transcription from *kurui*.

In other transcriptions gestural [13], spectromorphological [14] notation and "cartoon convention" symbols [15] were used to sonically transcribe the sounds of possums moving in the trees (Figure 6.)

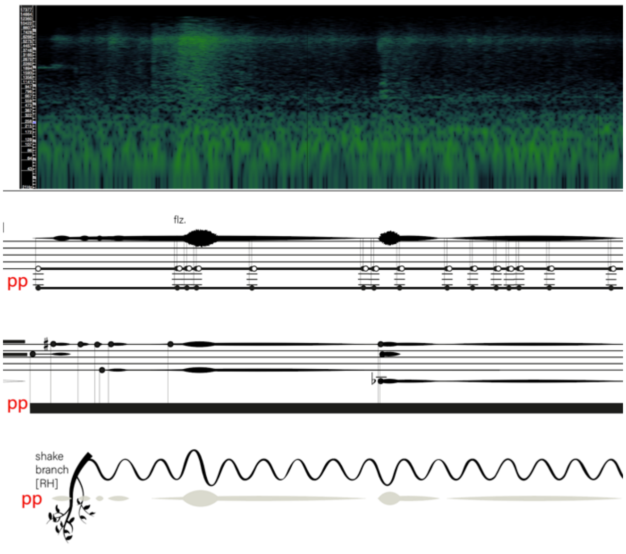


Figure 6. Gestural/spectromorphological transcription of Possum tree movements.

3.2. wardang [2019]

Wardang is the Wadjuk Noongar word for crow. The animal takes a central role in the Noongar familial lore, along with the Mannaitch (white cockatoo) defining rules of marriage and association. The recordings in this piece were made at Mooro Katta or Mount Eliza in Kings Park Perth, and include the three-part call typical of crows in this region and a peculiar almost "mewing" sound of a young crow. All of the electronic material was derived exclusively from this material and then assembled into a six-channel audio-collage.

Unlike *kurui*, which is entirely instrumental, crow recordings were used here "raw" and in combination with processed versions created using Eric Lyon's *FFTease* objects for *Max*: *thresher~* to sustain only the strongest sinusoids and *resent~* to alter the speed and phase of audio bins.

Instrumental parts were then created for flute, clarinet, electric guitar and harp, highlights and re-enforcing aspects of the collage (Figure 7).

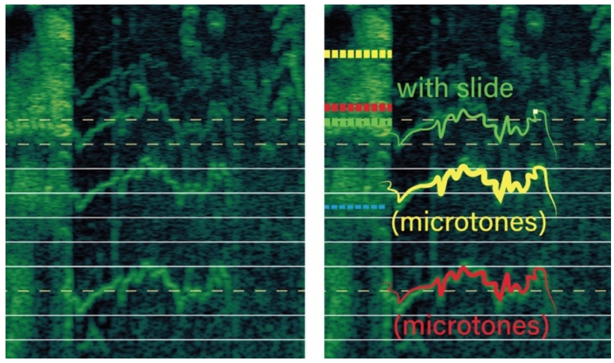


Figure 7. Transcription creating a spectral mixture in *wardang* showing greater weight on the centre har-monic (yellow = flute) and progressive less weight on the lower (red = bass clarinet) and then upper (green = electric guitar) harmonic.

Whereas *thresher~* has typically been used in these works as an audio expander to enhance spectrographic transcription, here the recordings were radically reduced to transform the recorded sounds into figures approximating instrumental music (Figure 8).

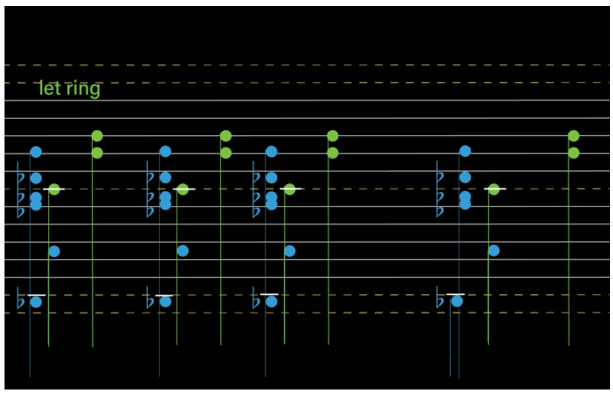


Figure 8. Instrumental figures for harp (blue) and electric guitar (green) drawn from crow recordings in *warding.*

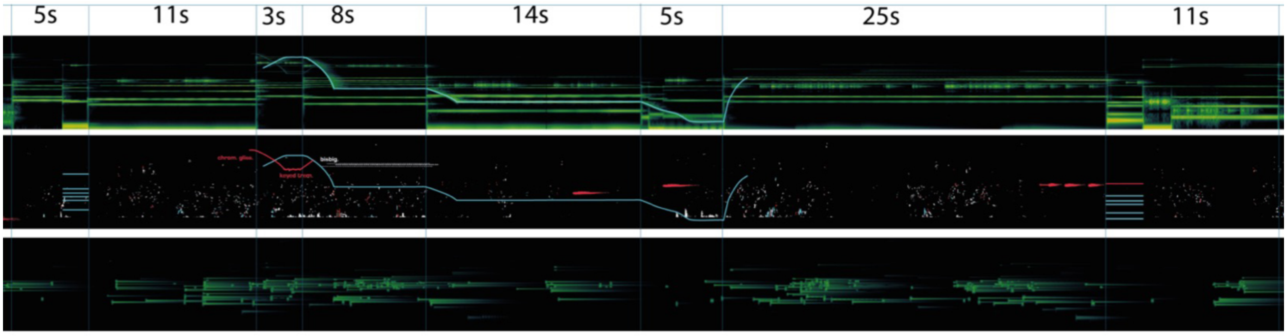


Figure 9. A segment of *rising water* showing its ecostructurally derived formal structure, created from major detected amplitude transitions causing changes in the “freeze” spectrogram (top), score (middle), original spectrogram (bottom).

In *wardang* the instrumental performance is embedded within a six audio channel audio-collage that is distributed amongst three iPads and may be output directly to bluetooth speakers, rather than output through a multichannel interface.

4. ENVIRONMENTS

4.1. rising water

For *rising water*, a field recording by Leah Barclay of flood waters in Queensland was resynthesized using three different processes: firstly by using frequency data to control amplitude and amplitude data to control frequency; secondly by "threshing" (as described above); and finally by "freezing" the frequency spectrum at points of prominent amplitude transitions and sometimes bending their pitchtowards the next amplitude transitions.

The author’s "Lyrebird" software [16] was used to cre-ate a "base" score representing strongest sinusoids in the frequency/amplitude exchanged recording and coloring them according to their timbral qualities. Illustrator was used to create notation representing the “threshed” and “frozen” recordings. The combination of these approaches permitted a rapid alternation between rapid, apparently random (but actually precisely placed) points of sound, periods of transition and complete stasis (Figure 10). This score was then separated into three colours (timbres) and distributed to the three instruments (bass clarinet, electric guitar and harp).

Then layers sometimes obscuring this score were built by transcribing shapes from the "thresh" and freeze" spectrograms. The variation in this material was used to define the sectional structure (which had been derived from the data in the original recording).

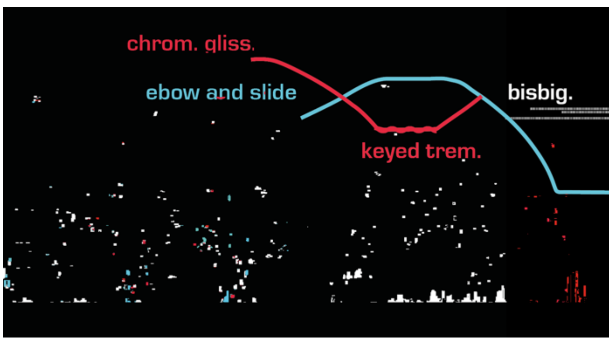


Figure 10. Excerpt of the score of *rising water* showing notation depicting rapid points of sound contrasted with frozen and gliding pitched material.

Significant changes in texture detected by spectral freeze process were then used to derive a sectional formal structure for the work (Figure 9). In this manner despite the transformation of the original field recording the work follows Ecostructural principals in which in which structural data derived from environmental sound sources “are used as the dominant material for creating the musical composition” (2006).

4.2. willsons downfall [2018]

The score for *willsons downfall* [2018] was written for a performance at Harrigan's Lane a property in the Great Dividing Range on the border of New South Wales and Queensland. LaBelle has stated that “composition becomes a form of research conveying cartographic routes in and through relations to place” (LaBelle, 2008: 198) and that is literally true of this work which relied upon satellite and terrain mapping of the Willsons Downfall locality around Harrigan's Lane.

In particular the score and structure of the work was assembled from topographical information related to three principal features: the Boonoo Boonoo River, Mount Lindesay Road and the tree line of the Bookookoora mountain ridge (Figure 11.).

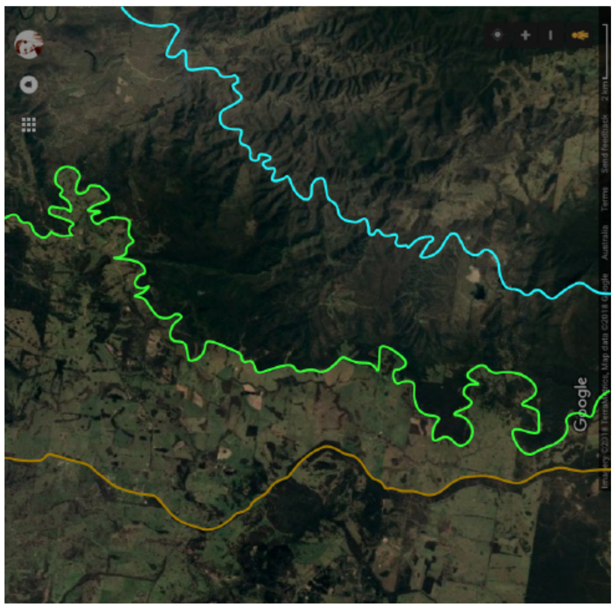


Figure 11. The Willsons Downfall locality around Harrigan's Lane with principal features: the Boonoo Boonoo River (blue), Mount Lindesay Road (brown) and the tree line of the Bookookoora mountain ridge (green).

Each feature is represented in the work by one of the performers. Other prominent topographical features – terrain, vegetation etc – were then extracted in *Illustrator* in a process roughly analogous to audio threshing. The three traced principal features were then used as "anchor-lines" to which the topographical features were attached according to their horizontal and vertical distance on the map (Figure 12). The lines (but not the topographical features) were then stretched out horizontally to represent the work's eight minute duration, spatially.

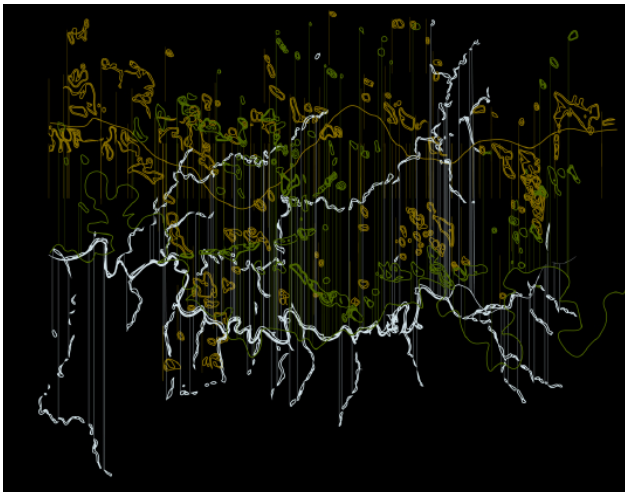


Figure 12. The three traced principal feature"anchor-lines" to which the topographical features were at-tached according to their horizontal and vertical dis-tance on the map

As a score, the traced principal feature lines were used to represent the volume for each performer and the topographical features were used to represent sonic complexes to be executed by the three performers. Several of these graphical complexes can be seen in Figure 13.

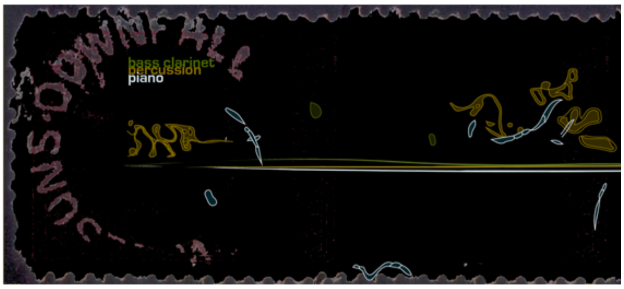


Figure 13. The opening of the score for *willsons downfall*, showing the beginning of the three "princi-pal feature"/ dynamic lines and several "topographical feature" glyphs.

Each of the three scrolling scores plays is embedded with one of the three field recordings from the Willsons Downfall region created by Jocelyn Wolfe. The recordings form the sonic connection to the terrain, forest and river topographies represented in each of the performer's scores. The "verticalised" contour of the feature was used to determine the volume and left/right spatial orientation for playback of each recording, by tracing it into a *function* object in *Max* (Figure 14). The recordings were intended to be projected through bluetooth speakers placed in the environment of throughout the audience in the case of a concert performance.

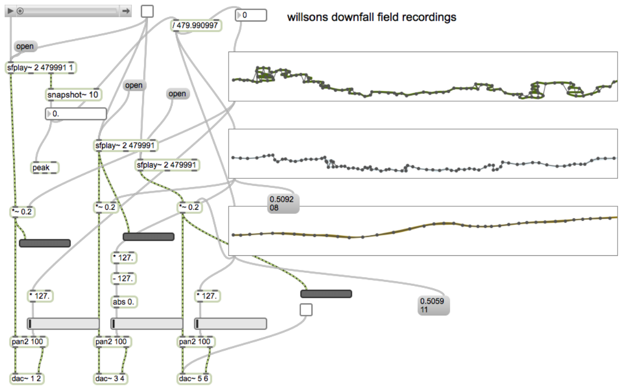


Figure 14. The "verticalised" contours of the three principal features were traced into *function* objects in *Max* and used to control the volume and left/right spa-tial orientation for playback of each recording.

4.3. njookenbooro [2018]

njookenbooro was created for the site-specific *series Limited Hangout: in the field*, instigated by the author. Works in the series were intended to explore approaches to the creation of site-specific music and means of establishing connections and interaction between sound and site. Composers were encouraged to use field recordings, topographical information, transcriptions, resonant frequencies, spectral niches [18] and other characteristics.

The site chosen for this performance was a walkway jutting into Herdsman Lake (Njookenbooro in the local indigenous language Noongar) in Perth.

The first section of the 27 minute score was created through the transcription of prominent environmental frequencies which were verified in *Audacity*'s "Plot Spectrum" function and then mapped directly onto a spectrogram of a field recording made at the location several days earlier at the same hour of the evening (photographs were also taken). Pitches were indicated via small staves before the shape denoting the sound (Figure 16).

In the central section the spectrographic renderings of the contours of bird and frog sounds were transcribed to create the notation (Figure 17).

The final section again focused on the prominent frequencies from the field recording, but verticalising the pitches into chords using a "spectral freeze" (Figure 18).

A panorama of the "field photographs" taken at the performance location was assembled in Photoshop and was used as the background for the scrolling score. Instruments were colour coded (using colours taken from photographs of the site). Partly because of the limited colour pallet, the performer's notation was also displayed as "parts" in which sits "above" the greyed-out notation of the other performers, in the Decibel Scoreplayer [19] (Figure 15).

Figure 15. Instrumental colour coding (using colours taken from photographs of the site).

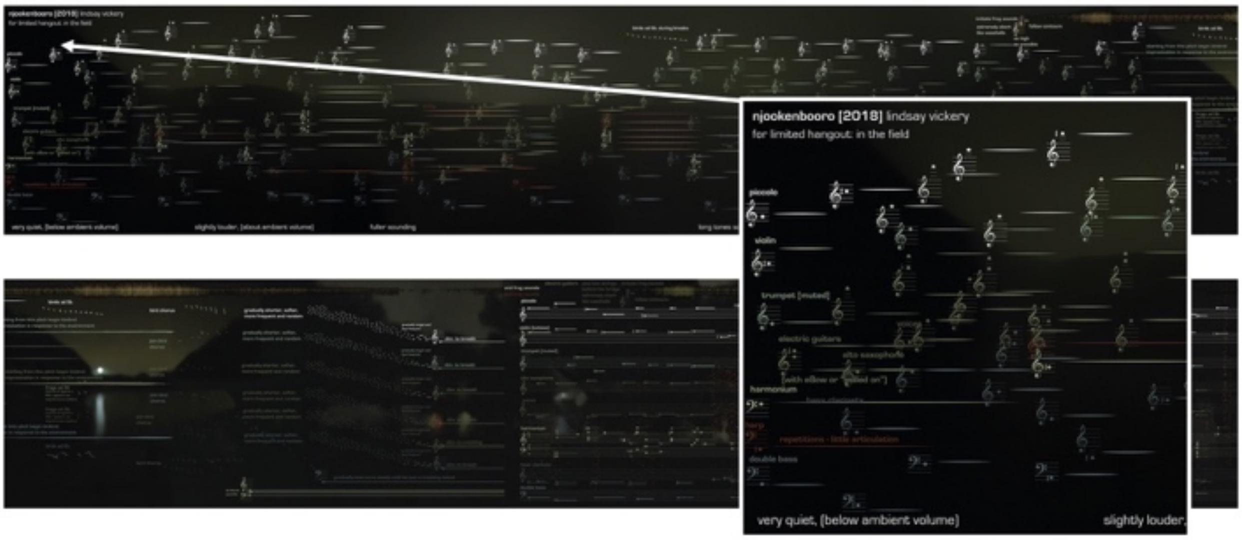


Figure 16. The first section of the score for *njookenbooro.*

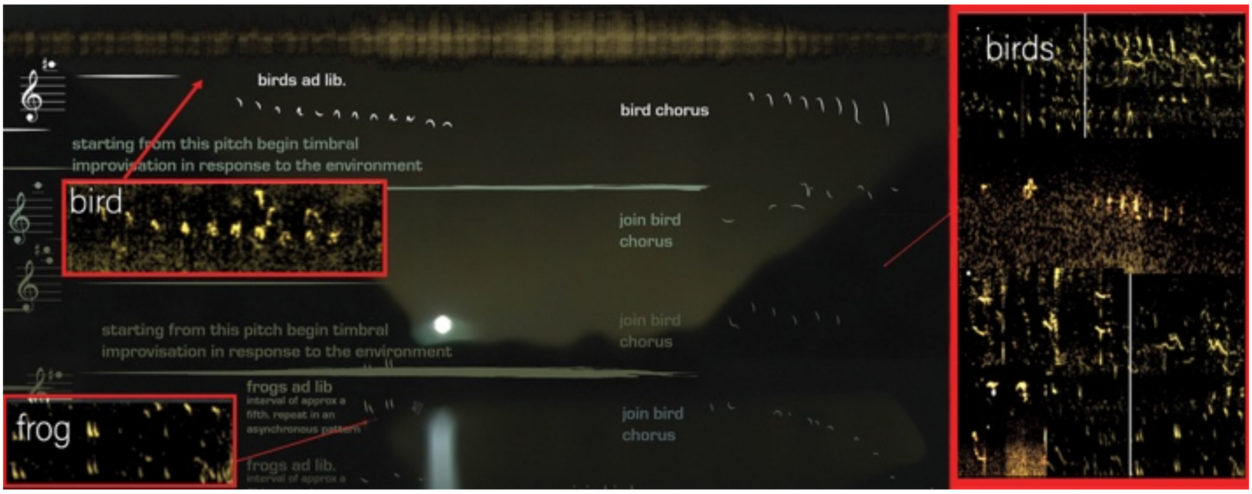


Figure 17. The central section of the score for *njookenbooro.*

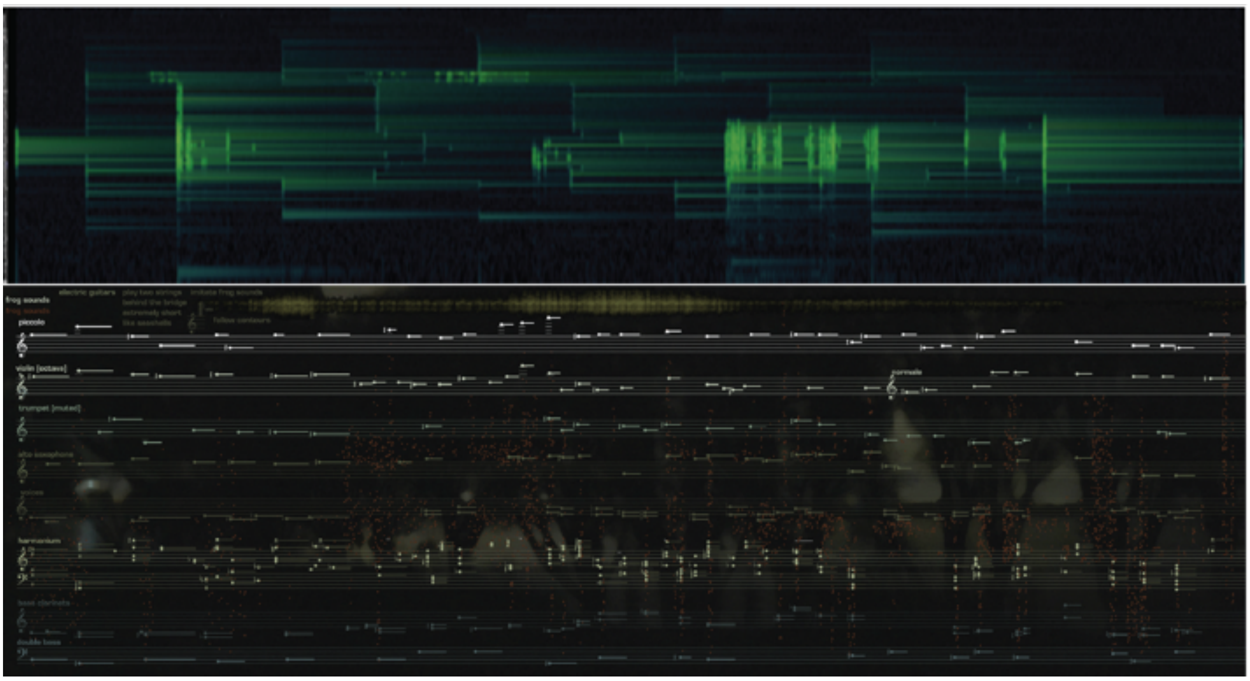


Figure 18. The final section of the score for *njookenbooro.*

Battery power allowed the performers to play from net-worked iPads, each connected via bluetooth to a speaker. An ad hoc WiFi network that was broadcast from a laptop. De-spite the extremely low ambi-ent light, the scores could read by the screenlight from the iPads. They were synchronised across an ad hoc WiFi network that was broadcast from a lap-top. Solar lights were used to mark the pathway to the per-formance for the audience.

The performers were arranged around a roughly circular area at the terminus of the walkway. Their Bluetooth speakers were arranged in a second pattern around the space. This permitted independent spatia-lisation of 10 unique stereo channels of audio and spatialisation of the live performers during the perfor-mance (Figure 19.)

Figure 19. Location map (above) and performer (blue circles) and speaker (red rectangles) placement in *njookenbooro*.

The field recordings used as a basis for the score comprised raw recordings mixed with threshed versions of the same audio highlighting different spectral niches.

5. CONCLUSION

The scores described here form the beginnings of a body of works seeking to interact with sounds of the Australian environment using a range of electronic tools: spectrographic transcription, audio processing, extended forms of notation and spatial audio. Other components of this pro-ject focusing on man-made sites and machines were not included in this discussion.

They aim to aid in the awareness of natural environ-ments where they still exist, but also to the sonic environ-ment generally, through bringing sounds into the focus of the concert hall as well as bringing audiences (and composers and performers) into the environment itself.

I mean the acoustic world in general, including that which is man-made. It's 'natural' in the sense that it's a 'given'? whether it be birdsongs, the noise of the ocean, or those of machines. One receives them and creates acoustic objects through the relation between what is given and consciousness. (François-Bernard Mâche) [20].

Acknowledgments

Thanks to Leah Barclay and Jocelyn Wolfe for permission to use their field recordings.

6. REFERENCES

[1] J. O'Callaghan, "Soundscape Elements in the Music of Denis Smalley: negotiating the abstract and the mi-metic". *Organised Sound*, vol. 16, no. 1, 2011: pp. 54-62.

[2] R. Ranft, "The Wildlife Section of The British Library National Sound Archive (NSA)". *Bioacous-tics: The International Journal of Animal Sound and its Recording.* vol. 7, no. 4, 1997: pp. 315-319.

[3] C. Palombini, "Machine Songs V: Pierre Schaeffer: From research into noises to experimental music". *Computer Music Journal,* vol. 17, no. 3, 1993: 14-19.

[4] H. Westerkamp, "Linking soundscape composition and acoustic ecology". *Organised Sound*, vol. 7, no. 1, 2002: pp. 51-56.

[5] S. R. Ellis, "What are virtual environments?" *IEEE Computer Graphics and Applications*, vol. 14, no. 1, 1994: 17-22.

[6] F. B. Mâche, *Music, myth, and nature, or, The Dolphins of Arion.* Taylor & Francis, 1992.

[7] F. López, "Environmental sound matter." *La Selva (Sound environments from a Neotropical rain for-est)* (CD). Holanda: V2_Archief. Cobertura, 1998.

[8] J. Bailie, "Squeezing out the music from real sound". *Interpretations*. Q-O2 Winter, 2014. http://www.q-o2.be/uploads/interpretations.pdf.

[9] D. Dunn, *Acoustic ecology and the experimental music tradition*. American Music Center, New Music Box, 2008.

[10] M. Zaveri and E. S. Rueb. "How Many Animals Have Died in Australia's Wildfires?" *New York Times*, Jan 11, 2019. https://www.ny-times.com/2020/01/11/world/ australia/fires-animals.html

[11] C. J. A. Bradshaw, "Little left to lose: deforestation and forest degradation in Australia since European colonization", *Journal of Plant Ecology*, vol. 5, no. 1, 2012: pp. 109–120.

[12] J. Kohl, "The Evolution of Macro-and Micro-Time Relations in Stockhausen's Recent Music." *Perspec-tives of New Music*, vol. 22, no.1, 1983: 147.

[13] J. Kojs, "Notating Action-Based Music". *Leonardo Music Journal*, vol. 21, 2011: pp. 65-72.

[14] M. Blackburn, "The Visual Sound-Shapes of Spectromorphology: an illustrative guide to composition". *Organised Sound*, vol. 16, no. 1, 2011, pp. 5-13.

[15] R. Patel, K. Schooley, and J. Wilner, "Visual Features that Convey Meaning in Graphic Symbols: A Comparison of PCS and Artists' Depictions", *Assistive Technology Outcomes and Benefits,* vol. 41, 2007: pp. 62–80.

[16] L. Vickery, "Visualising the Sonic Environment", *Proceedings of EVAA 2016, Electronic Visualisation and the Arts*, Canberra, Australia, 2016.

[17] T. Opie, A. Brown, "An Introduction to Eco-Structuralism". *Proceedings of the International Computer Music Conference*, New Orleans, USA, 2006.

[18] B. Krause, "The niche hypothesis: A hidden symphony of animal sounds, the origins of musical expression and the health of habitats." *Explorers Journal*, vol. 71, no. 4, 1993: 156–160.

[19] C. Hope, A. Wyatt, and L. Vickery, "The Decibel Scoreplayer: Enriched Scores For The iPad." *Proceedings of the International Computer Music Conference*, Denton, Texas 2015: 314-317.

[20] A. Infanti, "Without Fooling the Bird: Structural Anthropology and the "Naturalist" Music of F.-B. Mâche." *Perspectives of New Music*, 2004: 4-30.

NETWORKED COMPROVISATION STRATEGIES WITH ZSCORE

Slavko Zagorac
Goldsmiths College, University Of London
slavko@zagorac.com

Michael Zbyszynski
Goldsmiths College, University Of London
m.zbyszynski@gold.ac.uk

ABSTRACT

ZScore is a networked notation system for mixed ensemble composition and performance. This paper describes recent system developments towards a platform for comprovised music making. The long-term project objective is to provide an inclusive, democratised music-making environment by utilising technology that enables distributed decision making, dynamic notation processing and visualisation. It is proposed that all music is the outcome of a decision-making process that can be represented on the spectrum between immutable static and real-time dynamic decision making. Networking technology can act as an enabler for moving the dial on this decision-making spectrum in a required direction. Furthermore, a definition of a networked notational perspective is outlined, covering the dynamic aspects of distributed notation for heterogeneous clients. Several strategies for dealing with dynamic notation visualisation and control are presented, such as the dynamic performance parameter processing and embedded scripting. Finally, this paper presents the results from recent user trials and plans for future developments.

1. INTRODUCTION

Composition and improvisation are often regarded as mutually exclusive music-making categories. In practice, however, it is not possible to define a clear boundary between composed and improvised. Bhagwati [1] argues that no score can totally determine all aspects of a musical performance and that some elements of music-making will always be contingent. Likewise, a performer's free improvisation is built on years of practice and performance, stemming from a particular tradition and aesthetic context. A free improviser also adheres to a set of rules and regulations that might be imperceptible to a performer. It follows that any music performance lies somewhere on the spectrum between composed and improvised. The portmanteau word 'comprovisation' is often used to describe this mix. This paper considers music-making strategies that blur traditional boundaries and intentionally occupy the middle range of the comprovisation spectrum.

Music notations developed in different musical traditions tend to optimise the amount of written information to what is regarded as essential, and omit performance elements that are ubiquitous, verbose or too difficult to notate [2]. Traditions rely on the performer's understanding of the particular style, playing techniques and aesthetics to complete required information gaps missing in the score. In the contemporary context, this notational bias, described as 'notational perspective' [2], has a more granular scope that does not only define a tradition or a style of music, but helps identify individual comprovisation practices. Networked notation technology further extends this concept by allowing for the creation of dynamic notational perspectives that can be modified in real-time. The dynamic notation, in this case, becomes a context-dependent contingent element in a performance. This notation contingency can be achieved through generative symbolic notation algorithms or dynamic performance parameter control described further in this paper.

Existing networked comprovisation software solutions typically consist of a front end capable of rendering dynamic notation such as: InScore [3], MaxScore [4], Bach [5], internet browser etc.; and a server engine in charge of notation scheduling and distribution, usually hosted in software containers such as MaxMSP, Processing, Node.js or programming language environments like SuperCollider and ChucK. Comprovisador [6] is an example of how a complex, higher level comprovisation software tool can be built on top of existing software components. ZScore [7] is a networked notation system providing real-time notation distribution and performance flow control. It utilises InScore for notation visualisation and a custom-built Java engine for algorithmic notation processing and distribution. This paper outlines ZScore comprovisation strategies, focusing on a dynamic notational perspective and real-time decision-making controls.

2. COMPROVISATION DECISION-MAKING SPECTRUM

A piece of music, whether composed or improvised, is conceived through a decision-making process defining what sound or action is to be performed and at what time. In a composed piece, most music material decisions are made pre-performance in isolation by a composer who preserves these decisions as notation realised in a static score. Even in the most meticulously notated scores, however, many performance decisions are left to the performers who interpret the given notation based on their experience and knowledge of the particular music tradition. Rodrigo Constanzo [8] developed a formal methodology for analysis of the decision-making process in improvised works. His segmentation of the music-making decisions into material, formal, interface and interaction, illustrates the improviser's real-time dynamic decision-making approach in an interactive group performance environment. It could be postulated that all music-making exists within the comprovisation decision-making spectrum between predetermined statically notated decisions and dynamically made decisions in real-time (Figure 1). An important consideration

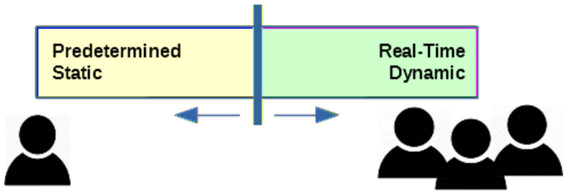


Figure 1. Music performance decision-making spectrum.

when discussing the comprovisation decision-making spectrum is to observe who the decision makers are, the type of the decisions made and their impact on the aesthetic evaluation of the piece of music. Traditionally, composers' decisions are immutable. If a performer intentionally or accidentally modifies a statically notated piece of music during a performance, it is generally interpreted as the aesthetically unsatisfactory realisation of a piece. In a free improvisation performance, performers are allowed to make individual decisions at any time. Although there are no right or wrong decisions in this case, the aesthetic value of a performer's decision is continually assessed within the context of a particular performance by the audience, the performer who made the decision, and by all other performers who react in a chain of decisions resulting in a self-regulating musical output. Comprovisation pieces that sit in the middle range of the decision-making spectrum contain both static immutable elements and a dynamic decision loop created by multiple decision makers in real-time. Concepts such as static data containers, networking, interactivity and real-time dynamic event processing are ubiquitous in modern computer technology, therefore, technology is well placed for modelling and enhancements of comprovised music-making data and decision processes. The intention behind ZScore is to provide a platform for the democratisation of music-making processes and unconstrained control of the comprovisation decision-making dial (Figure 1) as required by the context of a performed piece of music.

3. NETWORKED COMPROVISATION NOTATIONAL PERSPECTIVE

Conventionally, a music score consists of notation acting as a set of instructions or stimuli required for the realisation of a piece of music as imagined by the composer. As discussed in section 1 (Introduction), all scores contain a certain bias regarding the type and quantity of information defined as a 'notational perspective'. The quantity of information in a score should reflect the composer's intention as to what the piece of music is within the assumed notational perspective. 'New Complexity' composers [9] create dense, detailed, deterministic scores that intentionally challenge boundaries of performability. Jazz music notation often consists of a melody (a tune) and related chord progression charts, leaving out the instrumentation, dynamics and even tempo. Many composers utilise elements of indeterminacy in their compositions (John Cage, Earle Brown, Witold Lutosławski etc.) and their scores are constructed to reflect intended contingency, thereby frequently containing graphic or unusual symbolic notation layouts. In his influential composition *Pression* [10], Helmut Lachenmann focuses on the modelling of the multi-dimensional physical sound production gestures in a two-dimensional notational space by combining graphic, symbolic and textual notation elements. Taking this concept further, Aaron Cassidy in his *Second String Quartet* [11] formalised decoupling of the left and right hand playing techniques, further introducing the use of colour in gesture notation. In all cases described above, the score performance evaluation should examine whether the musical outcome matches the composer's intention.

Networked notation systems allow heterogeneous clients to connect and exchange data and events over a computer network. In such systems, a music score can be modelled as a collection of data and algorithms driven by a set of scheduled or triggered events. All participants in a networked music performance (musicians, composer, conductor, audience, digital audio and video engines, etc.) can communicate with each other through an interactive system in real-time. This multidirectional flow of information blurs the boundaries between traditional music-making roles, as any participant in a performance can be assigned a decision-making or a sound production agency. In a way, networked systems provide an environment reminiscent of ancient communal music-making (also described as 'musicking' [12]). Audience members, as networked system participants, can actively engage in a performance through personal mobile devices. The composer and musicians can interact in real-time during a performance to modify the composition flow, a conductor can change generative algorithm parameters to alter notation sent to the performers, and a computer running an algorithmic engine can trigger score visualisation on audience mobile devices etc.

This heterogeneous client environment requires multiple score representations as each network client might require a different input type. For example, a score on musicians' devices might be rendered as a symbolic notation while the same score on the audience's devices might be visualised as animated graphics. In Simon Katan's *Conditional Love* [1], the audience's mobile devices are used both for score visualisation and as audio sources. The audience's score representation can also contain a form of instructional notation, prompting the audience to perform certain actions at specific points in time, e.g. vocalise certain words. Furthermore, the audience's score representation on mobile devices can contain interactive elements, thus allowing the audience to send real-time events to other participants via an algorithmic notation engine. A part of the same score might be sent to a digital video engine translated to a device-dependent data protocol, as in Slavko

[1] S. Katan. *Conditional Love*, last accessed 12 Jan 2020. `http://www.simonkatan.co.uk/projects/conditionallove.html`

Zagorac's *Vexilla* (Figure 7). Likewise, the score data can be used to drive audio engines implemented in MaxMSP or SuperCollider.

Composers have to consider several new performance aspects in this interactive networked environment. The score becomes more than just a static notation collection. It requires a definition of dynamic performance elements and a strategy on how to present and deal with real-time events. The dynamic actions display should not interfere with the rest of the notation on musicians' screens. Performers need time to perceive any actionable elements and consequently take any required actions. Therefore, reaction time needs to be built into the score in a way that does not impact the performance flow. All participants need some kind of confirmation that their actions have been processed. If it is not immediately audible what the outcome of their actions is during a performance, then some kind of a visual acknowledgement is required. Likely issues with the network technology during a performance need to be taken into account. Some kind of default outcome should be defined if real-time events cannot be processed due to technical issues. The aesthetics of networked comprovisation will be evaluated through the participants' ability to perceive the score realisation and whether the impact of their decisions matches their desired musical output.

4. DYNAMIC NOTATION STRATEGIES IN ZSCORE

Unlike static notation, networked dynamic notation views for instrumental parts require a carefully thought-out refresh strategy that does not interfere with any currently played notation, whilst leaving enough look-ahead time for the performers to prepare for the upcoming material. The alternating pane notation strategy in ZScore [7] aims to resolve dynamic notation update issues by providing familiar left-to-right and top-to-bottom reading directions and pre-defined time windows when upcoming notation can be updated, thus leaving ample look-ahead time for performers (Figure 2). At any point of time during a performance, there is always one active and one preparatory pane. The preparatory pane is updated with upcoming notation within the pre-defined time window while the active pane is played [7]. Currently, ZScore notation is a combination of static files generated in Adobe Illustrator and dynamic SVG (scalable vector graphics) content rendered by InScore. This approach allows for syntax-independent and constraint-free positioning of notation elements suitable for various dynamic notation styles. Following the usability first principle, however, the current design preference is to provide a consistent layout familiar to classically trained musicians. Early versions of the notation layout used a mixed symbolic/graphic notation space, offering optimal and flexible screen real estate utilisation (Figure 7). However, after the feedback received through research questionnaires and verbal conversation, it became clear that the majority of musicians preferred distinct and consistent positioning of different performance parameters rather than the mixed space approach. Furthermore, some notation element positioning orders felt more 'natural' to

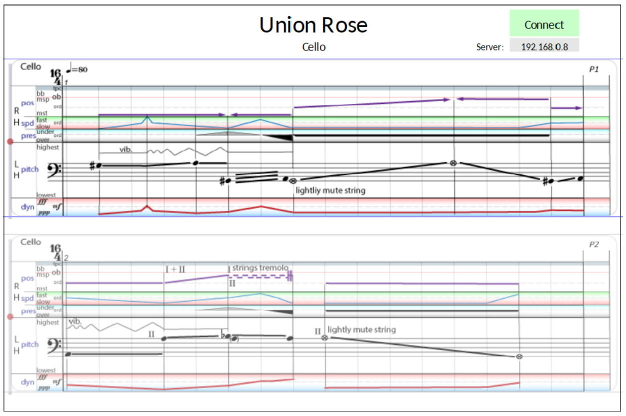


Figure 2. Cello part from *Union Rose* by Slavko Zagorac.

performers than others, such as having the dynamic markings always displayed below the pitch information. The above considerations and the multiple user trials have led to the current layout design illustrated in Figure 2.

4.1 Dynamic Performance Parameter Notation

Performers intuitively apply complex playing techniques when reading a notated score, based on years of learning, practice and performance. Classically trained string instrument players, for example, automatically translate music dynamic markings into multiple bowing techniques, such as the control of the bow speed and pressure. When interpreting symbolic notation, string players strive to produce a sound quality that satisfies aesthetic requirements of the particular tradition or style. This anticipation of the required 'ideal' sound then translates into learned application of the bowing techniques. The artistic aim in the latest pieces written for ZScore is to create a particular sound quality through notation that intentionally decouples learned mapping between the notation, playing techniques and sound. Additionally, the objective is to have a system suitable for both static and dynamic notation that allows for a flexible control of different playing techniques in real-time.

User trials and various concerns described above lead to the notation layout where the key performance parameters are separated into distinct two-dimensional Cartesian coordinate spaces. Figure 3 illustrates a vertical stave layout for string instruments taken from the score shown in Figure 2. The stave is vertically split into three main sections dedicated to the right hand, left hand and music dynamic notation. The right hand notation section is split into three subsections: position, speed and pressure notation. These subsections refer mostly to the bowing technique but can contain other actions such as pizzicato or percussive sound gestures. Left hand notation contains pitch or a playing technique information in either symbolic or graphic notation. The dynamics section is always displayed below the pitch information based on the feedback from the musicians during user trials. The vertical axis in each section (Figure 3) represents a named performance parameter value. The vertical range is parameter specific. For ex-

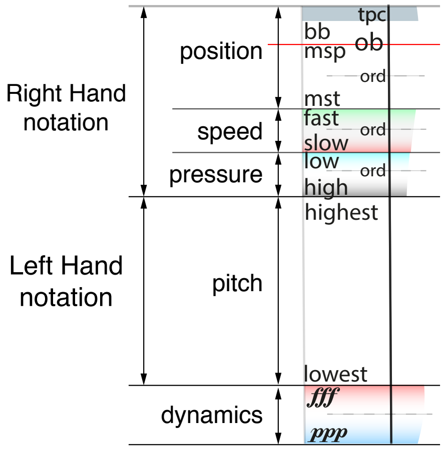


Figure 3. String instruments vertical notation layout.

ample, bow position range for all string instruments starts with molto sul tasto (mst) then continues with markers for ordinary playing position (relative to the given dynamics), molto sul pont (msp), on bridge (ob), behind bridge (bb) and ends with the tailpiece (tp) marker. The horizontal axis represents time and is identical for all performance parameters. The position cursor rendered as the green line across the stave shows the current position on each stave at any point of time (Figure 4). Additionally, the dynamic beat cursor in the shape of the red bouncing ball indicates the onset of each beat, similar to a conducting gesture. The

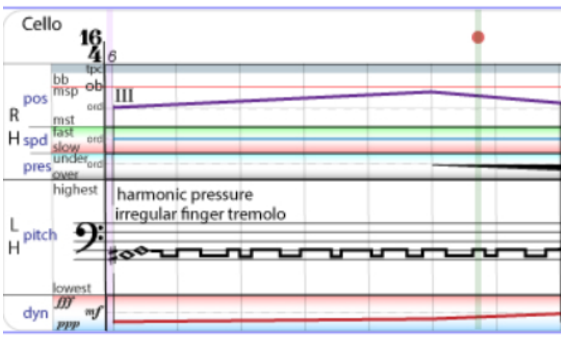


Figure 4. Cello part notation excerpt from *Union Rose*.

cello part excerpt (Figure 4) taken from *Union Rose* score (Figure 2) illustrates a pre-composed static notation layout. The current value for several performance parameters (position, speed and dynamic) is visualised as a continuous line. Pressure notation uses coloured geometrical shapes indicating the amount of bow pressure that needs to be applied relative to the current dynamics. Left hand notation, in this case, is a combination of textual, symbolic and graphic notation indicating finger pressure, position and timing. The finger tremolo timing information is an approximation that illustrates an idea of an irregular finger placement. It is left to a performer to decide the exact timing by using their aesthetic judgement at the time of the execution. This decision might be different every time the section is played as the overall context and the surrounding sound output can change with each pass as described below.

Real-time dynamic decision making and notation rendering require a user interface design that allows for dynamic elements to update at any point of time without any detrimental impact on the displayed static notation elements and musicians' look-ahead preparation time. Dynamic notation updates need to be clear, easily understandable and suitable for real-time cognitive processing. Therefore, complex notation updates should preferably be scheduled for display in predetermined time window slots as described above. As a general guideline, the performance continuity should not suffer at any point on dynamic notation update, unless it is an intentional side effect. One of the reasons that lead to the performance parameter notation separation was a need to provide dynamic parameter value overrides. In order to achieve dynamic overrides, each parameter is assigned a graphic overlay [2]. Overlays sit on top of the

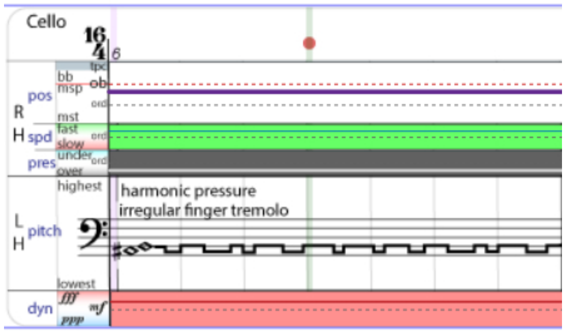


Figure 5. Dynamic notation overlays for position, speed, pressure and dynamics.

static notation covering the entire Cartesian space assigned to the corresponding performance parameter. The current dynamic parameter value is rendered either as a horizontal line on top of the overlay and/or as an overlay background colour value. Figure 5 illustrates the position, speed, pressure and music dynamic overlays for the same notation excerpt shown in Figure 4. In this case, only the pitch information remained the same as the pre-composed static notation. As can be observed in Figure 5, each parameter's current value is represented by the coloured line (red for dynamics, purple for position, blue for speed, grey for pressure) and the background colour covering the entire two-dimensional Cartesian space assigned to the parameter. In this instance the player is asked to play forte sul pont with fast bow and strong overpressure. Bow speed and position markings are always relative to the indicated music dynamics. Parameter values are controlled from the ZScore control GUI described below and have a preset range (min = 0, max = 100). The parameter's line position is obtained by mapping the current parameter value to the vertical space assigned to the parameter (minY, maxY). Similarly, the background colour is interpolated from the range representing minimum, middle and maximum value. For example, the music dynamics colour is interpolated from the

[2] S. Zagorac, "ZScore Dynamic Notation" video, last accessed: 30 Dec 2019. `https://youtu.be/Yh6wUqLZwkU`

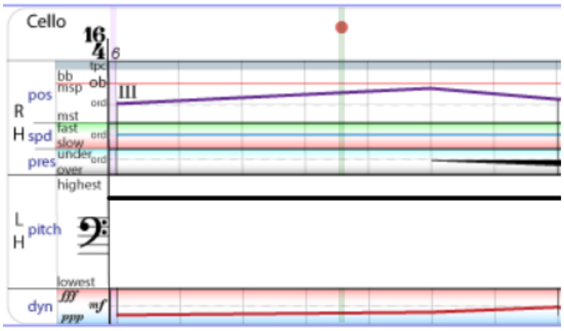


Figure 6. Dynamic notation overlay for pitch only.

range min: blue[R=0, G=0, B=255], mid: white[R=255, G=255, B=255], max: red[R=255, G=0, B=0]. Overlays and parameter values can be set independently at any point of time and are immediately displayed on the musicians' screens. Figure 6 illustrates a different overlay configuration for the same score excerpt as in Figures 4 and 5. Here, only the pitch parameter has been overridden with the overlay, indicating an approximate pitch to be played within the instruments range (min = lowest, max = highest possible pitch). All other performance parameters in this example are pre-composed. In a similar fashion, any overlay can be switched on or off during a performance.

4.2 Embedded Scripting

In ZScore, actionable commands can be embedded directly into the score, as illustrated in in the excerpt from *Vexilla* by Slavko Zagorac (Figure 7). Here, the AV (audio/video) part contains commands in the textual form. These commands are loaded together with the score and sent to the server for scheduling and distribution to the networked video or audio engines. In this instance, JavaScript and SVG commands are routed to an InScore client which renders a video signal displayed to the audience via a video projector. The AV command scheduling resolution in this example is one bar, so the commands are attached to a bar container and scheduled to be executed on the onset of the starting beat of the bar. The commands are executed immediately on the client side. Some commands have a duration, such as a transition between two colours, and are executed over a period of time based on command settings.

5. ZSCORE PERFORMANCE CONTROLS

Dynamic performance parameters are controlled from the ZScore's administration GUI, written in Java FX (Figure 8). Currently, this GUI also allows for import of static score data, communication with performance participants and their connection monitoring, various algorithm control selections, as well as the score position and start/stop controls. Performance parameter overlays described in the previous chapter can be switched on or off at any point during a performance from the administration GUI. Here, the performance parameter ranges are displayed in musical terms. For example, the dynamics range is displayed as 'ppp - p - mf - f - fff' (Figure 8). The administration GUI

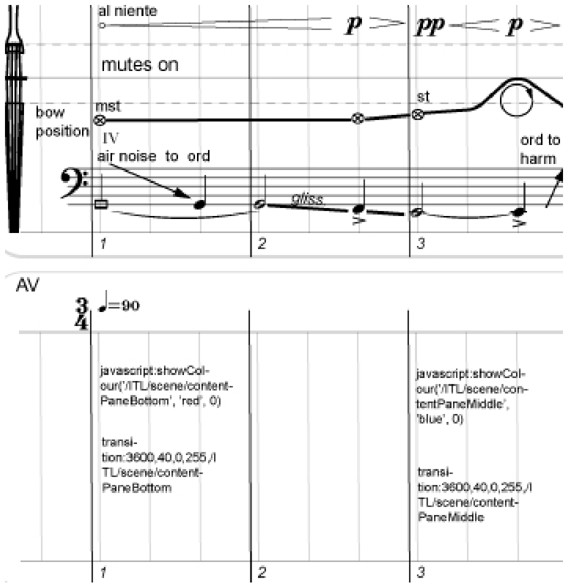


Figure 7. Embedded scripts in AV part from *Vexilla*.

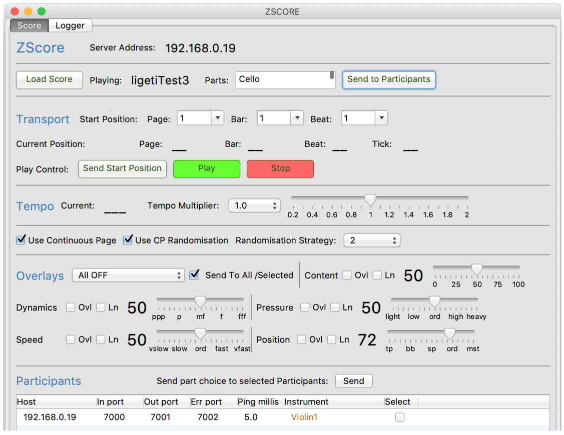


Figure 8. ZScore performance control front end.

communicates with other performance participants via ZScore's server, which provides notation scheduling and distribution service. The server also receives events sent by other performance participants, passes them through an algorithm which analyses incoming events and validates decision logic (Figure 9). The outcome of the decision logic is then passed to the score processing and scheduling engine which identifies required notation and distributes it back to the performance participants.

One of the administration GUI's features is the randomisation strategy configuration selection. The randomisation algorithm decides what notation and instrumentation should be used in the next time window as calculated by the scheduling engine. Figure 10 shows the available randomisation strategy configurations from *Union Rose* written for a string quartet. The configuration determines the number of distinct score pages and instruments that should be used. The term page here is equivalent to the notation displayed in a single pane. For example, the configuration value '2' means that two randomly selected instruments should play the same randomly selected page; the configu-

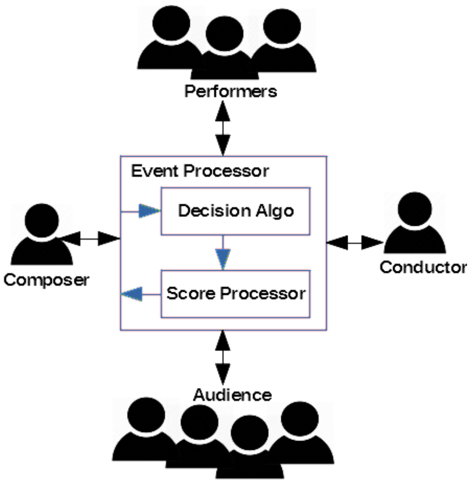


Figure 9. Decision event flow.

ration value '2,1' means that two instruments should play the same page and one instrument should play another randomly selected page; '1,1,1,1' means that all four string quartet instruments should play a randomly selected page, and so on. *Union Rose* pages have been constructed to

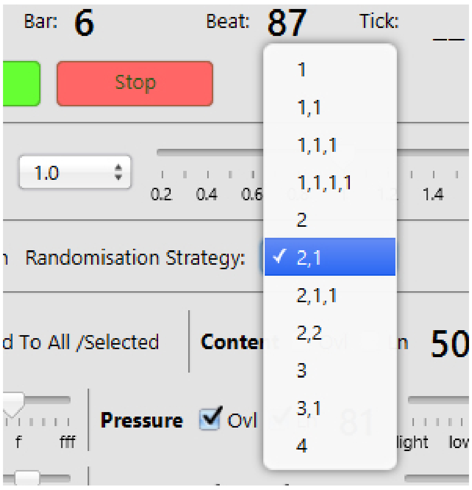


Figure 10. Randomisation strategy configuration control.

function as an independent unit, reusable in various vertical structure combinations. The aesthetic impact of sounding page combinations is evaluated in real-time during a performance. Any undesired sound elements can be modified with dynamic performance parameter controls to rebalance the page combination musical outcome.

6. COMPROVISATION USER TRIALS

ZScore has been used successfully in several live performances and workshops featuring networked score distribution, embedded scripting and synchronized multimedia. The latest version with the notation layout modifications and real-time comprovisation features described above was tested in a November 2019 workshop with the Ligeti Quartet [3], where an excerpt from *Union Rose* by Slavko Zagorac was performed [4]. The aim of the workshop was to validate the notation layout design changes and test technical stability and interactive aspects of the dynamic performance parameters.

The feedback from the musicians was mostly positive, citing: significant notation layout improvements compared to the previous versions; a 'natural' feel of the notation flow and the real-time performance parameter visualisation; a very good system stability; and accurate performance synchronization. The comprovisation features attracted a great deal of interest, including comments on how the repeated, familiar music material sounded completely different depending on the randomisation strategy configuration and dynamic performance parameter value selection. It was interesting to observe when and how musicians perceived dynamic notation changes. Initially, musicians focused almost entirely on the pitch notation section. Gradually, as they were getting more familiar with the composition material, they widened the scope first to dynamics notation and then eventually to the left hand performance parameter changes. Bowing pressure dynamic changes were only audible once musicians went several times through the same material and were comfortable with the pitch and dynamics notation. For future performances, the composition material will be distributed to musicians beforehand to give them some time to familiarise themselves with the piece. In addition, the performance plan will be modified to introduce dynamic performance changes gradually.

The musician's criticisms were mostly aimed at the quantity of visual markers, such as helper lines for mid-level performance parameter values, which increased the complexity of the user front end. The stave layout design is a compromise between notation precision and readability. In order to improve score readability, new features, such as current performance parameter value indicators (described in section 7: Future Work And Conclusions) will be implemented. Furthermore, the notation layout configuration features could be made available to musicians, thus allowing individual musicians to customise the notation layout as required.

7. FUTURE WORK AND CONCLUSIONS

At the moment all dynamic performance controls in ZScore are only available in the administration GUI. One of the ZScore project aims is to democratise the decision-making process and distribute performance controls to other participants. The short-term aim is to replace the randomisation strategy described above with separate controls for instrumentation and page selection. The musicians' front ends will be expanded with controls to allow for the selection of instrumentation, so in *Union Rose* for example, the quartet members themselves will be able to select instrument groupings and override any randomisation strategy

[3] November 2019 workshop with the Ligeti Quartet was funded by the Goldsmiths College Music Department.

[4] S. Zagorac, "Comprovisation with ZScore" video, last accessed: 10 Jan 2020. `https://youtu.be/2pBqepq3Khc`

instrumentation decisions (Figure 11).

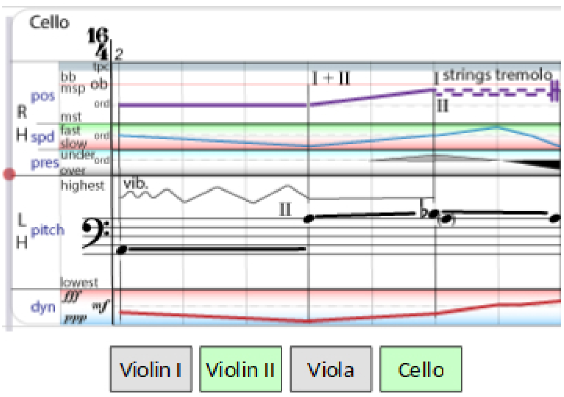


Figure 11. Musician's instrumentation selection controls.

Furthermore, the page selection and sequencing decision will be given to the audience. An example of the audience score visualisation for *Union Rose* by Slavko Zagorac is shown in Figure 12. The inspiration for this piece is the large stained glass rose window in Union Chapel, London, where the piece is scheduled to be performed. The audience score visualisation is a digital representation of the rose window split into 64 tiles. Each tile will be mapped to a score page which will allow the audience to select the path through the piece by clicking the next tile to be played on their mobile devices. The algorithm on the server side will process user input and schedule notation distribution as required. In order to improve the precision of dynamic

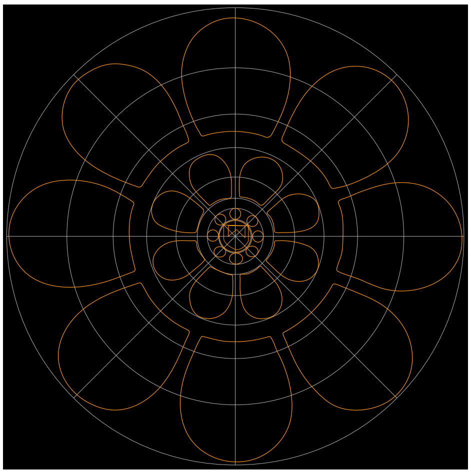


Figure 12. Audience score visualisation for *Union Rose*.

performance parameter display, an indicator of the current value will be attached to the current position tracker. An example of the music dynamics current value indicator is shown in Figure 13 ('mp' in the square box attached to the current position line). The indicator value will be set in real-time to reflect the dynamic parameter value. This should improve readability and decrease the notation layout complexity, as some markers, like the parameter mid-value lines, can be removed from the stave layout. The next version of ZScore will also provide dynamic SVG notation positioning both for symbolic and graphic notation styles.

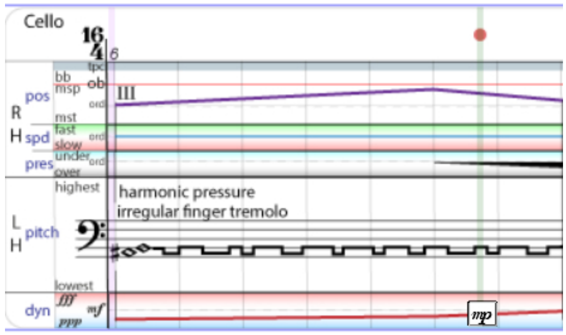


Figure 13. Current dynamics value indicator.

The score encoded in YAML format will allow for unconstrained named symbol positioning. Figure 14 shows two examples of symbolic notation, one where a named pitch is placed on the staff (G1) and one where a notehead (diamond) is explicitly placed in the x/y Cartesian space of the part's pitch notation area. Once the YAML score is loaded

```
notation:
  - { p: G1/4, stem: up, staff: 1}
  - { notehead: diamond, dx: 0.04, dy: 0.03 }
```

Figure 14. Symbol notation in YAML format.

into the ZScore sever engine, it could be passed into a generative algorithm capable of creating real-time notation for all participants. Symbol names (e.g., notehead=diamond) are mapped to SVG file names on the client side. SVG files representing all required notation symbols for a performed piece will be loaded on all notation clients prior to the performance so once the score symbol name is mapped to a file name it can be rendered immediately on the client side.

So far, several ZScore project goals have been achieved, such as the reliable real-time notation distribution over a network and dynamic performance parameter control, enabling networked comprovisation. The distribution of dynamic controls to different performance participants and multiple score visualisation representations are the next steps in the journey towards an inclusive comprovised music-making platform. This will require carefully thought-out strategies for real-time event management and score processing. The score composition process will have to evolve to take into account the impact of real-time events and their consequences. Contributions to these decision-making processes from both performers and audience members will inevitably bring additional risks and responsibilities to a comprovised music performance. The aesthetic evaluation of music-making in this shared comprovisation environment will, therefore, have to take into account the analysis of how well the musical and visual outcomes match the participants intentions. The hope is that this interactive human-computer system will lead to a new kind of musical and visual aesthetics.

8. REFERENCES

[1] S. Bhagwati, “Comprovisation – Concepts and techniques,” *Re) Thinking Improvisation: Artistic explorations and conceptual writing*, pp. 99–103, 2013.

[2] ——, “Notational perspective and comprovisation,” *Sound & Score. Essays on Sound, Score and Notation*, pp. 165–177, 2013.

[3] D. Fober, Y. Orlarey, and S. Letz, “Programming Interactive Music Scores with INScore,” in *Proc. SMC Sound and Music Computing conference*, 2013, pp. 185–190.

[4] G. Hajdu and N. Didkovsky, “MaxScore - Current State of the Art,” in *International Computer Music Conference*, 2012, pp. 156–162.

[5] A. Agostini and D. Ghisi, “Bach: An environment for computer-aided composition in Max,” in *International Computer Music Conference*, 2012, pp. 373–378.

[6] P. Louzeiro, “Improving sight-reading skills through dynamic notation–the case of comprovisador,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation–TENOR*, vol. 18, 2018, pp. 55–61.

[7] S. Zagorac and P. Alessandrini, “ZScore: A Distributed System For Integrated Mixed Music Composition and Performance,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR’18*, Montreal, Canada, 2018, pp. 62–70.

[8] R. Constanzo, “Making decisions in time,” last accessed: 27 Dec 2019. [Online]. Available: http://www.rodrigoconstanzo.com/2015/04/making-decisions-in-time/

[9] S. P. Duncan, “Re-Complexifying the Function(s) of Notation in the Music of Brian Ferneyhough and the ”New Complexity”,” *Perspectives of New Music*, vol. 48, no. 1, pp. 136–172, 2010.

[10] C. Utz, “Time-Space Experience in Works for Solo Cello by Lachenmann, Xenakis and Ferneyhough: a Performance-Sensitive Approach to Morphosyntactic Musical Analysis,” *Music Analysis*, vol. 36, no. 2, pp. 216–256, 2017.

[11] 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.

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

INTERACTIVE TIMESHAPING OF MUSICAL SCORES WITH BACH AND CAGE

Daniele Ghisi
University of California, Berkeley
CNMAT
danieleghisi@berkeley.edu

Andrea Agostini
Conservatorio di Torino
andreaagostini@conservatoriotorino.eu

Eric Maestri
Conservatorio di Genova
eric.maestri@conspaganini.it

ABSTRACT

In this article we aim to organize the collection of practices that amount to modifying the temporality of a musical score in order to create a new one, by representing them as a combination of three basic families of operations: dilations, distorsions and repetitions. The initial score can be borrowed or designed on purpose, and can be of any complexity (an entire score, a single note, or anything in between). We call the complex of these practices 'timeshaping'. We describe the analytical space derived from the proposed organization and show how many of its processes can be implemented in an interactive computer-aided composition environment using the *bach* and *cage* libraries for Max.

1. INTRODUCTION

In this paper we focus on a set of practices that we call 'timeshaping'. These practices are defined by the transformation of a pre-existing musical representation—such as a score—through the reorganization in time of its elements.

More specifically, timeshaping can be thought of as a 'symbolic processing technique' and, as such, it can be applied to both specifically conceived musical materials and pre-existing ones. In a way, similarly to the signal processing techniques, these two possibilities mirror the way an audio effect, such as a filter, can be employed both as a part of a synthesizer, thus being instrumental to the production from scratch of a new sound object, and as a step of the processing chain applied to a prerecorded sound sample. If pushed to the extreme, these processing techniques, in both the audio and the symbolic domains, can become essential tools for working within the broad conceptual area of musical borrowing [1, 2], which defines an operative approach laying between the traditional compositional strategies of reinterpretation of pre-existing material and the manipulation of a corpus of data that can be performed in real time. One may refer to these two compositional attitudes, which define two conceptual poles with a vast area of overlapping between them rather than a clear, black-and-white opposition, respectively *'tabula rasa'* and *'tabula plena'* approaches [3].

Timeshaping, potentially as an instance of an ampler, yet-to-be-defined family of symbolic processing techniques, shares some other traits with sound processing techniques: for example, it can be both fluid and written; it can conceptually be performed in both real and deferred time; its parameters can be defined both through experimentation and formalisation; and it is an abstract operation (or, rather, a family thereof) which is cognitively significant, and substantially independent from the details of its practical implementation. Each of the techniques described in the following sections can indeed be expressed through a variety of tools, insofar as these tools provide ways of representing and manipulating symbolic musical events: these include all the major software systems for computer-aided and algorithmic composition, such as OpenMusic [1] [5], PWGL [6] and Abjad [7]. However, we shall discuss one specific implementation of timeshaping practices, the only comprehensive one we are aware of, that is, a subset of the *cage* package for Max [8], developed by two of the authors as a complement and an expansion of the *bach* package.

1.1 *bach* and *cage*

bach [9] is a set of modules for Max (both external objects written in C/C++ and abstractions developed in Max itself) establishing a few new data types aimed at representing symbolic musical data, a large set of tools for operating upon those data types, and some objects for displaying, editing graphically and playing back the musical scores they represent. *cage* [10], on the other hand, is a Max package built upon *bach* and taking advantage of its features. It implements a set of musically meaningful operations ranging from middle-level utilities (such as the parsing of SDIF files) to high-level musical processes (such as generation of scales and arpeggios, or symbolic ring modulation).

One of the main, overarching principles of the *bach* ecosystem is real-time reactiveness: generally speaking, *bach* complies with the real-time data-flow computational paradigm of Max, which, although quite complex from a theoretical point of view, fosters a novel kind of relation with computer-aided composition, one in which the computer becomes a real musical instrument [11, 12].

[1] Our approach is fundamentally different from the one proposed in [4], where the author, under the term 'timesculpt', describes a series of analytical and combinatorial techniques applied to rhythms. In particular, our approach does not separate the rhythmic component from the other musical parameters, but rather treats each score as a whole.

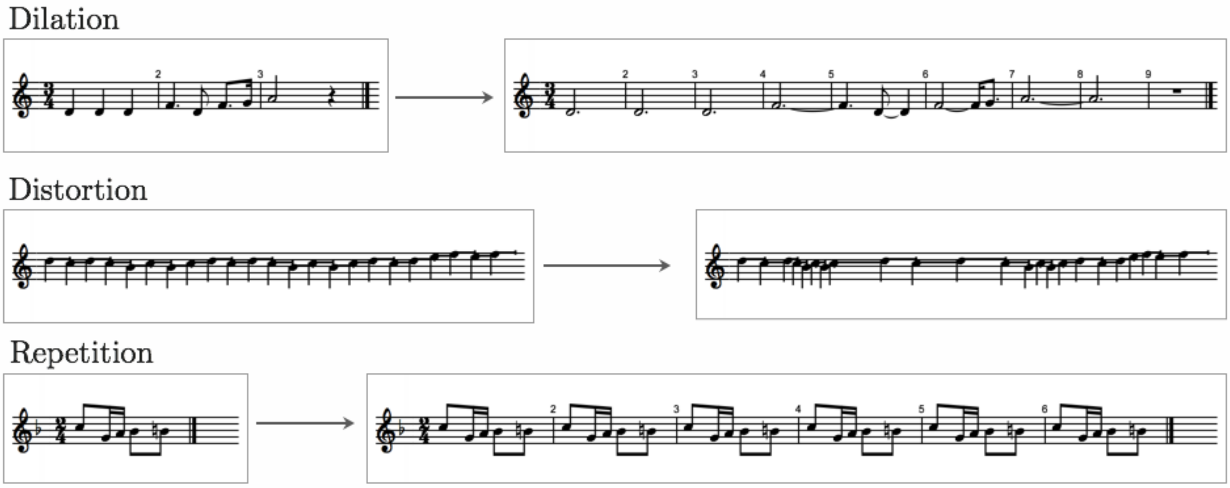


Figure 1. A simple example for each of the main timeshaping modifications. From top to bottom: augmentation from a section of Josquin's *Missa L'homme armé (Agnus Dei)*; temporal distorsion of the beginning of Bach's Cantata BWV 249a; repetition of a figure in Beethoven's 6th Symphony (1st movement, measure 17, first violins).

1.2 Relationship with compositional time

As Kramer suggests, time has a dual nature [13, p. 18]. On the one hand, it is linear, when it concerns becoming; on the other hand, it is non-linear, since it is thought in its spatial dimension. Linear time defines the flow that goes from the past to the future passing through the present; nonlinear time is visualised, represented by musical notation. Composers usually deal with both types of time: non-linear time (the composed one) is a model for linear time (the perceived one).

Libraries such as *bach* and *cage* allow a specific type of interaction between these compositional times. The computer becomes a real musical instrument for sketching and experimenting with direct feedback in the hands of the composer. As a consequence, the musical representation status changes and can be considered as a performed musical object. Conceiving notation as a 'temporal instrument' [14], *bach* and *cage* allow to reduce the distance between linear and non-linear time, thanks to 'symbolic-sonorous objects' that can potentially be manipulated performatively. These libraries offer an interactive 'graphematic space' [15, p. 215] which is, at the same time, a representational space.

2. TIMESHAPING

We call 'timeshaping' a collection of compositional practices that modify the temporal shape of a musical score (borrowed or designed on purpose), or a part of it—an entire score, a single note, or anything in between—in order to create a new one. These timeshaping practices are common when applied to audio buffers, but less so when applied to symbolic musical representations. We can distinguish at least three orthogonal axes, representing different kinds of modifications (Fig. 1):

1. **Dilation**: uniformly stretch or shrink the content of the source score by a given factor. Sometimes the factor is inferred in order to match a given target duration. A classical example of usage of pure dilations are augmentation or diminution canons. If the factor is negative, the dilation is also combined with a retrogradation (reversal).

2. **Temporal distortion**: warp the content of the source score internally, without changing its overall duration. The operation is non-linear and may happen in a variety of different ways. It is accomplished by reading the source score at a variable rate, under the condition that the overall elapsed time stays unchanged. Rhythmic interpolations between different musical figures often fall in this category. As a side note, we can consider rhythmic quantization as a very specific type of distortion, approximating a proportionally notated score with a traditionally notated one.

3. **Repetition**: replicate the content of the source score at regular intervals in time.

One may attempt to use the space determined by the three orthogonal axis as a map on which to organize a number of timeshaping processes: such an attempt yields, in our view, Fig. 2. Such a diagram does not aim to any mathematical consistency, rather it is meant to be an informal analytical tool. For one thing, the dilation axis represents both the amount of 'stretching' and the amount of 'shrinking'. Secondly, temporal distortions can of course be so complex that representing them on a single axis makes for a drastic approximation; it should also be remarked that the constraint by which a distortion may not cause a modification of the overall duration of the original score can be seen as an arbitrary one, but it is useful to clearly distinguish it from dilation; indeed, most typical agogical processes, such as accelerandos and rallentandos, involve both categories. Finally, when repetitions are mixed with other modifications, there may be ambiguity on whether the modification is to be applied to all instances equally or can be applied with different parameters to different instances—we shall abide by the latter interpretation.

It should be clear to this point that the axes of this space do not constitute a representation of the actual parameters of one specific process, but they will hopefully provide a conceptual framework for discussing such processes in terms of few elementary operations.

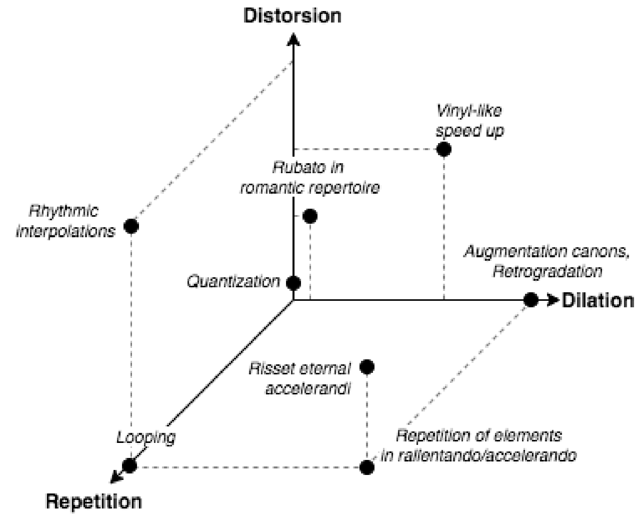


Figure 2. A diagram showing some timeshaping practices organized by amount of dilation, repetition and distortion.

3. TIMESHAPING PRACTICES IN CAGE

A portion of the *cage* library has been designed to allow users to manipulate the temporality of scores. We will describe their general underlying ideas and how they relate to the representation proposed in the previous section; their scope of application with respect to the three main axes is represented in Fig. 6. For all that follows, let t_L be the time of the source score (sometimes called 'logical time'), and let t_E be the elapsed time in the output score.

3.1 Dilation

The simplest module allowing the modification of the temporality of a source score is *cage.timestretch*: it stretches the incoming score either by a fixed factor or in order to match a given target duration (Fig. 3), i.e., $t_E = k \cdot t_L$. The module works both on proportionally notated scores, as represented by the object *bach.roll*, and on traditionally notated ones, as represented by *bach.score*. In the latter case, the dilation factor has to be a rational number [2], and the user has the ability to choose how meters in the output score should be handled.

3.2 Temporal warping

The *cage.timewarp* module performs more refined local modifications of the temporality of a source score; in its general form, it is able to reassign notes and markers to some other position in time, usually defined through the usage of a 'transfer function' f, in a lambda loop configuration [16]. [3] In short, the object module from its rightmost 'lambda' outlet the original position in time t_L of every item (such as a note or a marker) and expects to receive in its rightmost 'lambda' inlet either its desired resulting temporal position or the rate of the remapping. More specifically, the module operates in two in two different ways according to the value of the 'order' attribute:

- With 'order 0', it performs a time-to-time mapping: for each original temporal position, given in milliseconds, the transfer function returns a new temporal position at which the corresponding musical items must be remapped, as in $t_E = f(t_L)$. For instance, if the function f is such that $f(1000) = 2000$, then a chord with onset at 1 second will have its onset at 2 seconds in the resulting score. If $f(t_L) = kt_L$ is a linear function of slope k, then the module coincides with *cage.timestretch*.
- With 'order 1', it maps every source time (on the x axis) to the *rate* of the remapping, i.e., to the derivative $r(t_L) = dt_L/dt_E$. For instance a function $f \equiv 1$ will simply yield the same source score; a function $f \equiv k$ will result in a timestretch of a factor $1/k$; linear functions yield uniform accelerandi/rallentandi, and so on.

It is interesting to remark that *cage.timewarp* with 'order 0' provides a quite direct implementation of the dilation and distortion axes of the timeshaping conceptual space described above, in that distortions correspond to non-linear functions of time against time, whereas dilations generally correspond to linear functions. Assuming $f(0) = 0$, no

Figure 3. A simple example of dilation of a musical score obtained via *cage.timestretch*.

[2] Rational numbers are not implemented in Max per se, but they are one of the data types added by the *bach* package.

[3] A lambda loop is an idiomatic patching configuration typical of several *bach* and *cage* objects and abstractions: one or more dedicated outlets of a module output data iteratively to a section of the patch that must calculate a result (such as a modification of the original data or a boolean return value) and return it to a dedicated inlet of the starting object [16]. Lambda loops are an attempt at providing a way to implement custom specialization of generic processes, in a similar way to what happens with anonymous function in several programming languages, in the strictly non-functional paradigm of Max. For example, sorting a list of MIDI pitch numbers by pitch class can be achieved through the *bach.sort* object with a lambda loop containing a modulo 12 operation.

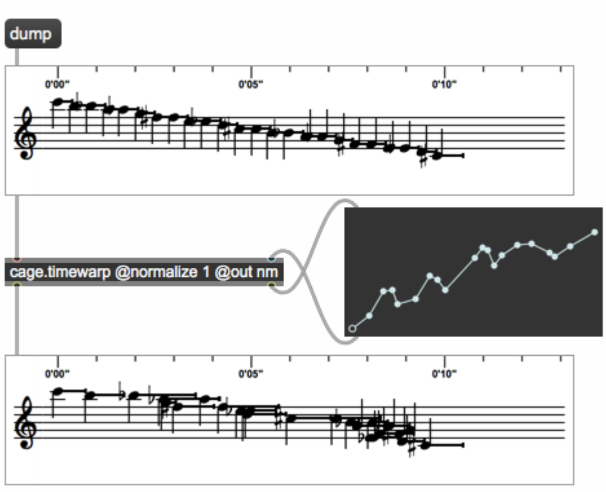


Figure 4. An example of temporal distortion of a descending chromatic scale obtained via *cage.timewarp*. Notice how the order of notes can also become shuffled.

dilation takes place if and only if $f(D_L) = D_L$ with D_L being the duration of the source score (if the object is set to normalized mode, this means that $f(1) = 1$, thus including for instance all the functions $f(x) = x^\alpha$). The linear function $f(x) = x$ means that no temporal modification has been performed, and corresponds to the origin of the timeshaping space.

The module can be also used in a 'normalized' mode, so that the source time t_L is received in the normalized range $[0., 1.]$, 0 corresponding to the beginning of the score and 1 to its end. This normalized usage makes it easy to perform pure temporal distortion, without modification of the overall score duration (see Fig. 4).

In a typical usage scenario, the lambda loop of *cage.timewarp* contains a graphical breakpoint function, rather than an algebraically defined one. Especially in this case, the 'order 1' case can be seen as more intuitively musical, as higher values on the y axis correspond to faster tempi, rather than the somehow more abstract notion of more advanced positions in the score.

With this in mind, this module can be seen as a flexible tool for representing various shapes of accelerando or rallentando, the latter of which can be pushed to the extreme case of producing a 'negative' tempi, that is, retrograde readings of portions of a score. Consistently with the overall design of *bach* and *cage*, which both put a strong focus on the reactivity of operations, an interesting aspect of *cage.timewarp* lies in that it allows to perform a real-time experimentation with the temporality of the original score, one in which the composer can interactively modify the transfer function, either graphically or through some parameter, and immediately receive from the machine a feedback showing the result. It is our position that this kind of interactivity represents in itself a novel paradigm in the field of computer-aided composition.

3.3 Varied repetition

The *cage.repeat* module produces identical repetitions of a source score (either in proportional or in standard notation). The *cage.looper* module goes further, providing the possibility to only loop a portion of score (and not the entire one), and to allow for each iteration to be individually modified, according to a lambda loop mechanism: each loop iteration (in *bach.roll* format) as well as its duration are output through the lambda outlets of the module: the user can provide a custom mechanism to alter the loop, re-injecting the next iteration (in *bach.roll* format), along with its possibly modified duration, in the module's lambda inlets. Due to the fact that such modifications are designed to operate continuously on the musical parameters, *cage.looper* only works with proportionally notated scores (*bach.roll*s). It should be remarked that, albeit originally based upon a time-shaping paradigm, the operation of *cage.looper* transcends the scope of this article, as it allows not only to modify the timing of the original score, but also its pitch content and, more generally, all its parameters in an iterative way.

3.4 Accelerandi and rallentandi

The *cage.agogics* abstraction allows to express rallentandi and accelerandi through a set of high-level musical parameters. The module combines the repetition functionalities of *cage.repeat* with the time distorsion and dilation approach provided by *cage.timewarp*, and indeed is based internally upon those modules.

The idea behind *cage.agogics* is to only expose control of three musically important parameters:

- N, i.e., number of repetitions of the source figure;
- D_E, i.e., the total duration of the accelerando/rallentando (i.e. the duration of the output score);
- r_{end}, i.e., the ending rate, determining how faster/slower the last repetition is, compared to the original (the output figure always starts with the same rate as the source score, i.e. $r_{\text{start}} = 1$).

The *cage.agogics* module assumes constant acceleration, which makes the three parameters not independent. In particular, let D_L be the total duration of the source score and let $r(t_L) = dt_L/dt_E$ be the remapping rate (and $r_{\text{end}} = r(ND_L)$), then the acceleration $a = \dfrac{r_{\text{end}} - 1}{ND_L}$ is assumed to be constant. Hence $a = \dfrac{r(t_L) - 1}{t_L}$ and by integration

$$t_E = \frac{\log(a \cdot t_L + 1)}{a}$$

and substituting $t_L = ND_L$ (as $t_E = D_E$) as well as the equation for a, one obtains the fundamental relation between the three musical parameters of *cage.agogics*:

$$\boxed{(r_{\text{end}} - 1)D_E = ND_L \log(r_{\text{end}})}$$

Only two of the three parameters are hence needed as input from the user—the third one is inferred.

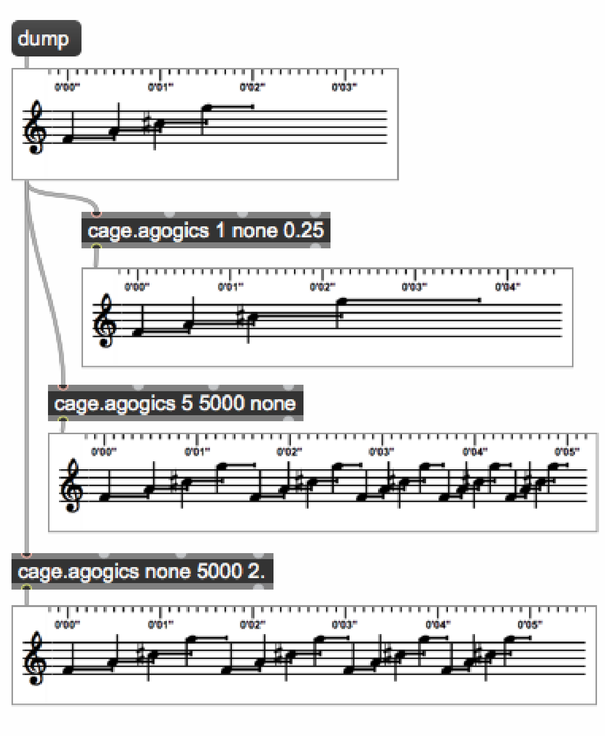


Figure 5. Three examples of usage of *cage.agogics*: the three arguments of the module are, respectively, the number of repetitions N, the total duration D_E and the ending rate r_{end}. The original cell is the uppermost score; then, from top to bottom: 1 repetition of the original cell ending 4 times slower; 5 repetitions of the cell lasting 5 seconds in total; a certain number of repetitions of the cell lasting 5 seconds and ending at about twice the original speed.

- If N and D_E are given, then

$$r_{\text{end}} = -\frac{ND_L}{D_E} W\left(-\frac{D_E}{ND_L} e^{-\frac{D_E}{ND_L}}\right),$$

  where W is the Lambert function (product logarithm);

- If N and r_{end} are given, then

$$D_E = ND_L \frac{\log(r_{\text{end}})}{r_{\text{end}} - 1};$$

- If D_E and r_{end} are given considered that N has to be integer, we take

$$N = \text{round}\left[\frac{(r_{\text{end}} - 1)D_E}{D_L \log(r_{\text{end}})}\right],$$

  and then proceed to recalculate r_{end} starting from D_E and the newly found N.

It could be worth remarking that the score subject to repetition does not necessarily have to be the actual material intended as the final musical result of the process. The repetition paradigm is a simple one to manage, hence its choice, but the process can be as well applied to an elementary cell, such as a single note, whose role is that of constituting a

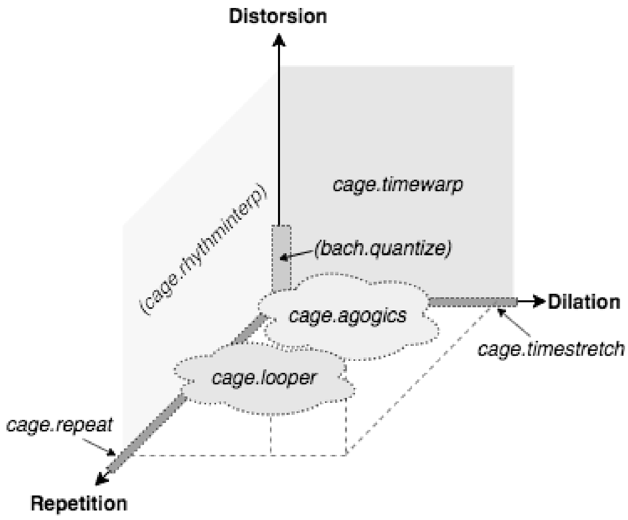


Figure 6. The organization of timeshaping modules of *cage* inside the space of Fig. 2.

rhythmic grid which will serve as a lattice for a further compositional process. This of course is true for all the other operations described in this article, but seems particularly relevant with respect to this specific process, whose very formulation might otherwise appear arbitrarily restrictive.

3.5 Rhythmic interpolation

Rhythmic interpolation is achieved, in *cage*, by using the *cage.rhythminterp* abstraction. Differently from the previously described modules, this abstraction does not operate directly on scores, but rather only takes the rhythmical parameters into account (onsets and durations). This may however be interpreted as a way to achieve a certain kind of temporal distortion. The general idea is that, given a number n of rhythm-only scores, each containing m notes, it is possible to create a new one whose features lie somehow "in between" the original ones. Let us assume that o^i_j and d^i_j are respectively the onset and the duration of the j-th note of the i-th score ($i = 1, \ldots, n$, $j = 1, \ldots, m$); then we can build an interpolated score in a straightforward manner, having onsets $\bar{o}_j = \sum_i w_i o^i_j$ and durations $\bar{d}_j = \sum_i w_i d^i_j$, for $j = 1, \ldots, m$. The weights w_i determine the importance of each score in the interpolation and are normalized such that $\sum_i w_i = 1$.

The case where the original scores do not contain the same number of notes is more complex. The way we chose for tackling it actually treats all the scores as if they contained the same amount of notes, with some notes being duplicated so as to provide a fixed number of interpolation paths. For example, let us suppose that the first score contains three notes A_1, A_2 and A_3, and the second score contains two notes, B_1 and B_2. The idea behind the interpolation model adopted is that it is possible to build four lines upon which the interpolation happens: A_1 to B_1, A_3 to B_2, A_2 to B_1 and A_2 to B_2 (see Fig. 7). In principle the interpolation actually produces four notes, but when two (or more) of them have both their starting and ending times within a given threshold ϵ they are collapsed into a single one: in

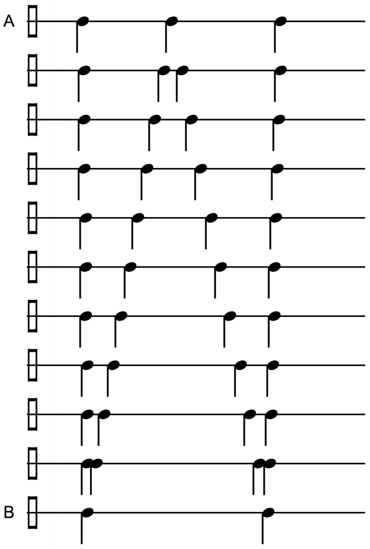


Figure 7. Rhythmic interpolation of two scores A and B containing a different number of notes (in 9 internal steps).

this way, at least the initial and final interpolation points will coincide with the original rhythmic structures.

4. QUANTIZATION

As hinted at before, we can consider rhythmic quantization as a very specific process of temporal distortion. Whether it should be included in this discussion is debatable, as quantization is generally considered a purely technical process, rather than a compositional tool for the musical elaboration of symbolic materials, since its aim is most often to convert a non-measured score into a measured one, or to simplify the rhythmic spelling of a pre-existing score. Nonetheless, in the practice of composers such as Emmanuel Nunes, quantization processes are arguably used with a somewhat direct aesthetic goal [17]: we shall therefore briefly discuss some aspects of quantization here; a more complete treatment will be perhaps the subject of a future article.

The term 'quantization' refers to the process of converting a score from proportional notation, where onsets and durations are real numbers representing some absolute timing, to standard notation, where onsets and durations are represented as rational subdivisions of a 'whole'. Quantization systems have a long history (see, in particular, [18, 19, 20]); in *bach*, quantization is achieved via the *bach.quantize* module, a rather complex one, with many different options and working mechanisms meant to address various specific cases which we shall not describe in detail here.

A simple quantization process could amount to 'snapping' the starting and ending positions of each note in a score to the nearest sixteenth subdivision. This operation would preserve the overall duration of a score, and alter its internal articulation through a consistent, deterministic rounding function of time against time: in this sense, it would precisely fit the definition of the distortion axis given above. Moreover, this operation would closely relate to a very idiomatic signal processing technique, bitcrushing, which is actually a kind of distortion performed through

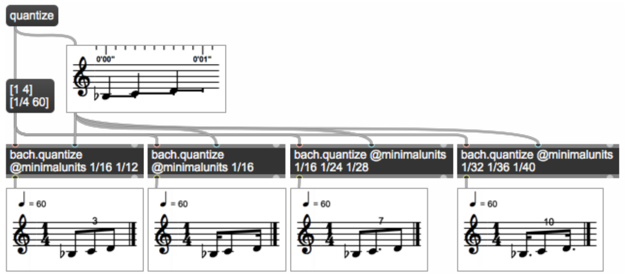


Figure 8. Temporal distortions given by quantizations of a proportionally notated fragment with the same meter/tempo and different 'snapping grids'.

applying a rounding function of instantaneous amplitude against instantaneous amplitude to an audio signal.

More complex kinds of quantization, also possible through *bach.quantize*, typically choose different rounding functions according to the specific contents and context of a temporal window of the score, thus performing an adaptive approximation rather than a fixed one. In the case of *bach.quantize*, the specific function adopted for each temporal window is not defined analytically, but constructed by a backtracking algorithm.

5. EXAMPLES

5.1 Risset rhythms

Following [21], we can implement in *bach* and *cage* a process producing eternal accelerandi or rallentandi build upon a given 'tenor'; the patch is shown in Fig. 9 (the 'tenor' is represented in the upper *bach.roll*, the resulting pattern is obtained in the lower *bach.roll*). The main building blocks are indeed *cage.repeat* and *cage.timewarp*, controlled via a very precise warping function, along with a volume windowing (via *cage.volume*) and a possible transposition of the pitches—the transposition stage can be bypassed, decoupling the treatment of agogics from the pitch shifting. Assuming that for a layer of index $v \geq 0$ the source score has been looped 2^v times, the equation connecting the source time and the elapsed time is

$$t_E = D_L \left(\log_2 \left(\frac{t_L}{D_L} + 2^v \right) - v \right),$$

where D_L is the duration of the source score. For more information on how this formula is obtained and on the mathematics involved, see [21]. [4] The patch heavily relies on the programming capabilities provided by the *bach.eval* module [22].

5.2 Vinyl-like speed up of scores

We can model the effect of a vinyl or tape recorder speeding up or slowing down (i.e. combining time warping and pitch shifting), and applying it to a proportionally notated score.

[4] In our implementation, the T and τ variables mentioned in the paper are set to have the same value $T = \tau = D_L$.

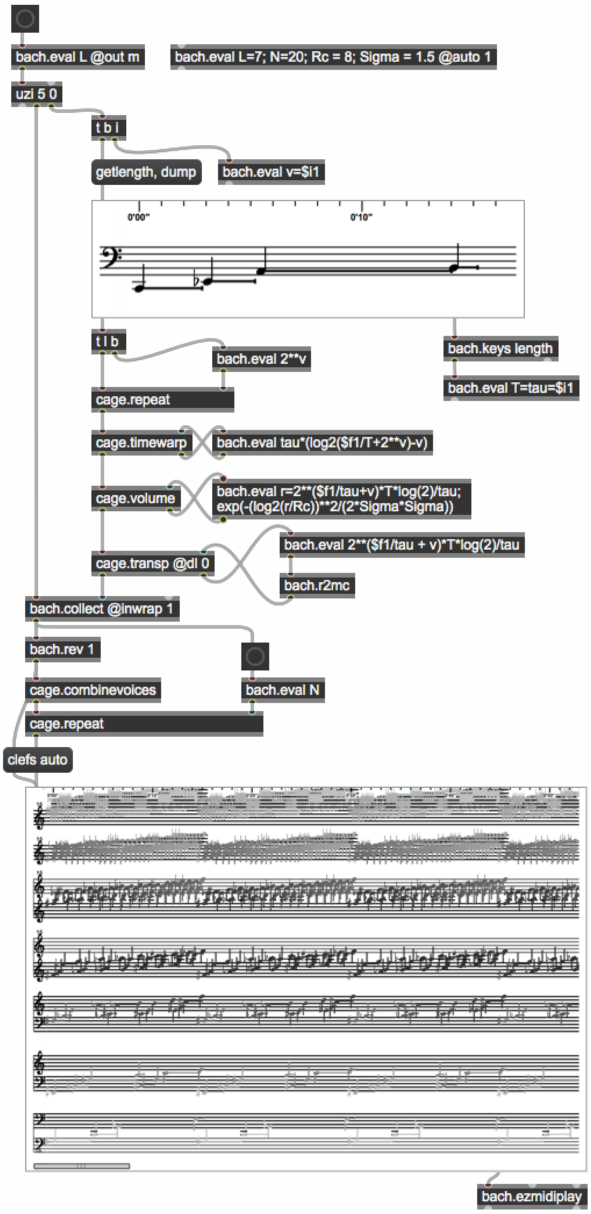


Figure 9. An example of patch producing Risset rhythms starting from a musical cell, implemented following [21] and used in Daniele Ghisi's 'Jean-Claude' (from the Rockenhausen Almanach)

Suppose you need to speed up the beginning of a score, until a certain note at onset $t_L = \overline{T}_S$. We can model the speeding up rate as $r(t_L) = (t_L/\overline{T}_S)^\alpha$ if $t_L < \overline{T}_S$, with $\alpha > 0$ being a reshaping exponent. Hence, for $t_L < \overline{T}_S$ one gets

$$t_E = \int_0^{t_L} \frac{1}{r(u)} du = \overline{T}_S^\alpha \int_0^{t_L} \frac{1}{u^\alpha} du = \frac{\overline{T}_S^\alpha}{1-\alpha} t_L^{1-\alpha}$$

with $0 < \alpha < 1$ needed for the integral to converge. This also yields a total speeding up time of $\overline{T}_S/(1-\alpha)$.

We remark that, conveniently, the slowing down portion can be seen as the retrogradation of the speeding up of the retrogradation.

Intuitively, this process yields a longer score than the original — unless one crops out only the final portion of the speeding up, and the initial portion of the slowing down in order to retain the absolute duration of each portion.

The patch implementing the process, including the possible cropping, is shown in Fig. 10.

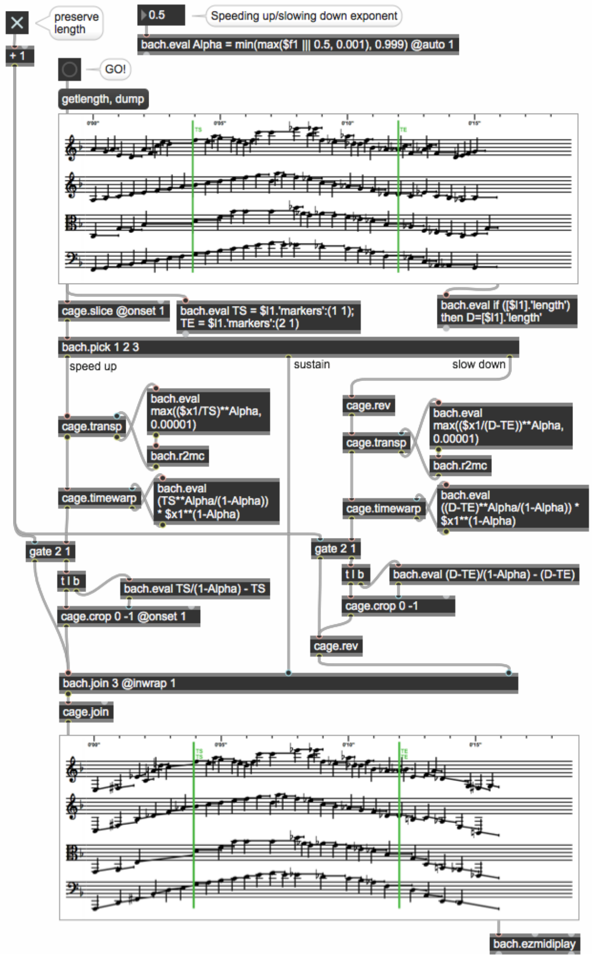


Figure 10. A patch producing a sped up (in the beginning) and slowed down (in the end) version of a source score (in this case, the beginning of Ravel's quartet). The shape of the agogics can be controlled via an exponent; the length of the original score can be preserved (via trimming). This process is used in Daniele Ghisi's music for 'La Chute'.

6. CONCLUSIONS AND FUTURE WORK

We have organized the collection of 'timeshaping' practices, which amount to modifying the temporality of a score, into three different axes (dilations, distorsions and repetitions), and we have shown how some common musical practices lie within this categorization. We maintain that interactive computer-aided composition is an important, innovative approach to bridging the dual nature of time (linear, when it concerns becoming, and non-linear as source time). In particular, we have described how the *cage* timeshaping modules fit into the geometry of the aforementioned axes and we have provided some examples of applications.

There are still many open areas of interest. From a music theory perspective, the concept of interactive timeshaping needs to be explored further: we have hinted at some of the implications in section 1.2, which may be, in the future, the starting point for a more detailed research on compositional practices. From a mathematical perspective, one may attempt to characterize the space defined in section 2 more formally; as an example, we think it would be interesting to investigate a general framework for temporal distorsions. These models may then have implications on the development of new computer aided-composition tools, such as a module for temporal distorsions, controlled via high-level musical parameters. This module might possibly stem from an extension of *cage.agogics* to account for non-uniform acceleration and variable starting rate. This would also in part overlap with the possibility of adding an ‘order 2’ attribute to *cage.timewarp*, allowing users to define acceleration in the lambda loop. Furthermore, quantization has a series of problem of its own, ranging from the capability to adapt the behavior to the musical processes, to the search for semi-automatic, dynamical meter and tempo inference tools.

7. REFERENCES

[1] J. P. Burkholder, “The uses of existing music: Musical borrowing as a field,” *Notes*, vol. 50, no. 3, pp. 851–870, 1994. [Online]. Available: http://www.jstor.org/stable/898531

[2] J. Boyle and J. Jenkins, *Theft! A History of Music*. Duke Center for the Study of the Public Domain, 2017.

[3] D. Ghisi, “Music Across Music: Towards a Corpus-Based, Interactive Computer-Aided Composition,” Ph.D. dissertation, Université Pierre et Marie Curie, Université Paris-Sorbonne, Sorbonne Université, IRCAM (UMR STMS 9912), 2017.

[4] K. Haddad, “Timesculpt in openmusic,” in *Agon, C., Assayag, G. et Bresson, J., éditeurs: The OM Composer’s Book*. Ircam/Delatour, 2006, vol. 1.

[5] C. Agon, “OpenMusic : Un langage visuel pour la composition musicale assistée par ordinateur.” Ph.D. dissertation, University of Paris 6, 1998.

[6] M. Laurson and M. Kuuskankare, “PWGL: A Novel Visual Language based on Common Lisp, CLOS and OpenGL,” in *Proceedings of International Computer Music Conference*, Gothenburg, Sweden, 2002, pp. 142–145.

[7] T. Baca, J. W. Oberholtzer, J. Trevino, 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*, 2015.

[8] M. Puckette, “Max at seventeen,” *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002.

[9] A. Agostini and D. Ghisi, “Real-time computer-aided composition with *bach*,” *Contemporary Music Review*, no. 32 (1), pp. 41–48, 2013.

[10] A. Agostini, E. Daubresse, and D. Ghisi, “*cage*: a High-Level Library for Real-Time Computer-Aided Composition,” in *Proceedings of the International Computer Music Conference*, Athens, Greece, 2014.

[11] P. Desain *et al.*, “Putting Max in Perspective,” *Computer Music Journal*, vol. 17, no. 2, pp. 3–11, 1992.

[12] A. Agostini, D. Ghisi, and J.-L. Giavitto, “Programming in style with bach,” in *Computer Music Multidisciplinary Research*, 2019.

[13] J. D. Kramer, *The Time of Music: New Meanings, New Temporalities, New Listening Strategies*. G. Schirmer, 1988.

[14] E. Maestri, “Notation as temporal instrument,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR’16*, R. Hoadley, C. Nash, and D. Fober, Eds. Cambridge, UK: Anglia Ruskin University, 2016, pp. 226–229.

[15] H.-J. Rheinberger, “Transpositions: From Traces through Data to Models and Simulations,” in *Transpositions: Aesthetico-Epistemic Operators in Artistic Research*, M. Schwab, Ed. Leuven University Press, 2018.

[16] 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, 2015/10/03 2015. [Online]. Available: http://dx.doi.org/10.1162/COMJ_a_00296

[17] K. Haddad, “Fragments de recherche et d’expérimentation : Eléments de réflexions autour de l’écriture rythmique d’emmanuel nunes,” https://medias.ircam.fr/x8f9800_karim-haddad-fragments-de-recherche-et-d, accessed: 2020-03-26.

[18] C. Agon, G. Assayag, J. Fineberg, and C. Rueda, “Kant: a critique of pure quantification.” in *ICMC*, 1994.

[19] K. Sprotte, M. Laurson, and M. Kuuskankare, “Ksquant-complex score manipulation in pwgl through quantization,” in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2008, pp. 253–261.

[20] A. Ycart, F. Jacquemard, J. Bresson, and S. Staworko, “A supervised approach for rhythm transcription based on tree series enumeration,” in *Proceedings of the TENOR Conference*, Cambridge, UK, 2016.

[21] D. Stowell, “Scheduling and composing with risset eternal accelerando rhythms,” in *Proceedings of the International Computer Music Conference*, 2011.

[22] A. Agostini and J. Giavitto, “*bell*, a textual language for the bach library,” in *Proceedings of the International Computer Music Conference (to appear)*, New York, USA, 2019.

THE MINIMUM CUT PITCH SPELLING ALGORITHM: SIMPLIFICATIONS AND DEVELOPMENTS

Benjamin Wetherfield
University of Cambridge
bsw28@cam.ac.uk

ABSTRACT

This paper describes and refines the Minimum Cut Pitch Spelling Algorithm, designed for flexible use in modern software and contemporary music settings. In the process of composing notated music, a decision must be made by the composer as to which enharmonic spelling should be assigned to each represented pitch. Spelling assignments in close proximity on the page are interrelated, with each choice exerting a pull on the surrounding choices. Hence, the complexity of the problem can proliferate, especially where tonal centers are contextually ambiguous or even non-existent. The minimum cut approach herein presents a model for spelling pitches efficiently based on their intervallic relationships. Building on the previous presentation of the model (in the author's bachelor's thesis), simplifications and extensions of both the workings of the algorithm and its exposition are given. Among the simplified components of the presentation are the system of encoding applied to pitch spellings, the approach taken to avoid double-accidentals, and a decoupling of the full complexity of the model from its simplest pitch relationships. A new practical inverting (or 'learning') process for generating algorithm parameters from collections of spelled pitches (based on the Edmonds-Karp Maximum Flow algorithm) is also introduced.

1. INTRODUCTION

In the process of composing music, pitch spellings make up a conduit for sound, an intermediate representation. Without due care, however, spellings can lead to a score that is difficult to read, rehearse or tune. Secondary as it is to musical sound, the assignment of "correct" spellings – or at least a good "default" – is an obvious candidate for automation in modern software. Various pitch spelling procedures have been proposed, with a number successfully reproducing the spellings of corpora of canonical works to a high degree of accuracy [1, 2, 3], but there are further measures of utility, and indeed further use cases, that can be applied to pitch spelling algorithms.

In [3], Honigh proposes four criteria for identifying a useful pitch spelling algorithm: *accuracy* in reproducing the spelling choices present in well-spelled scores, *parsimony* – a term introduced by Cambouropoulos to signify the avoidance of accidentals, and particularly doubled accidentals [4], *generalizability* – the value of the model in illuminating other features of music, and *cognitive plausibility* – the extent to which the pitch spelling approach mirrors the mental process of spelling done by a musician. To these criteria we can add several more. *Applicability* of the algorithm to different musical styles and contexts, *flexibility* of the algorithm to factor in user-preference and interaction, and *extensibility* of the algorithm to perform spellings with microtones. The latter three criteria are emphasized in the construction of the minimum cut pitch spelling algorithm, along with a reliance on the efficiency of algorithms studied in graph theory and operations research. A deep look at the construction of microtonal pitch spelling from the basic algorithm proves to be beyond the scope of this paper, but further details can be found in [5].

Despite seeking efficiency, this algorithm does not seek to be "real-time" in the sense that it could be used to perform real-time transcription or other stream-input/output operations. It should be noted that streaming pitch spelling algorithms are achieved elsewhere [2, 6]. Rather, the algorithm seeks to make strong use of a static score, taking advantage of the potential for pitches to exert a tug on each other forwards *and backwards* across the page. Interestingly, the static view allows us to abandon "windowing", a common feature in pitch spelling algorithms whereby a moving frame of reference restricts the pitches that can interact with each other as we move through the score from left to right. Instead, we make use of an extended network of pitches, the strength of whose connections is dampened over larger distances in the score.

In this paper, I will describe simplifications to previous modeling steps, which in turn clarify the exposition of the algorithm as a whole (see [5] for a comparison). In particular, I decouple musical context and proximity information from the pitch relationships governing spelling choices, leading to a further level of flexibility in allowing a user fine-grained control over the weight and adjacency structure of the pitch network. I examine inverting procedures, by way of which algorithm parameters can be extracted from spelled input data, and analyze a new inverse approach, which brings several practical benefits.

2. MINIMUM CUT

A **weighted directed graph** is a set of *nodes*, together with a set of *arcs* that denote one-way (directed) connections be-

$\downarrow$	$\uparrow$	**Encoded spelling**
0	0	Flattest spelling
1	1	Sharpest spelling
0	1	Natural spelling

Table 1. Encoding system for all pitch classes except 8, for which there is no 'natural' spelling (only G♯ and A♭), and hence, for which the $(0, 1)$ entry is removed (see Table 2).

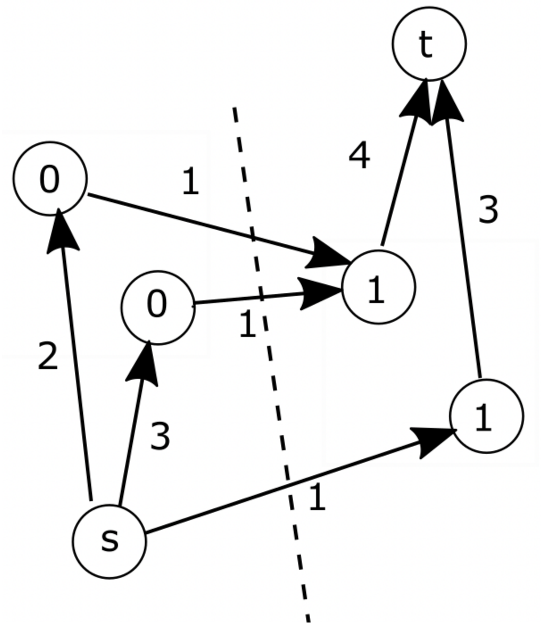


Figure 1. A flow network, with source s and sink t. The minimum cut is indicated with a dashed line. Numbers beside arcs indicate their weights. Numbers inside of nodes indicate on which side of the cut they fall: nodes with a 0 lie on the source side, while those with a 1 lie on the sink side.

tween nodes; each arc is given a numerical *weight*. A **flow network** denotes a weighted directed graph with two special nodes, a *source* and a *sink*. By convention, no arc starts at the sink and none ends at the source. Solving for the **minimum cut** of a flow network partitions the set of nodes into nodes strongly connected to the source, nodes strongly connected to the sink, and (possibly) nodes strongly connected to neither. The first class of nodes can all be assigned the value 0, the second class the value 1, and the third class arbitrarily 0 or 1 (assigned to the class in its entirety). The set of arcs that go from a node of value 0 to a node of value 1 is referred to as a cut. The cut is a *minimum* cut, when the total weight of its arcs is the least of all possible cuts in the network. Fast algorithms for solving for the minimum cut, either directly, or via the 'dual' maximum flow property, are well-known [7, 8, 9].

3. PITCH SPELLING MODEL

The strategy for constructing the pitch spelling model proceeds by encoding spelled pitches using a pair of binary

Pitch Class	Flattest Spelling	Sharpest Spelling	Natural Spelling
0	D𝄫	B♯	C♮
1	D♭	B𝄪	C♯
2	E𝄫	C𝄪	D♮
3	F𝄫	D♯	E♭
4	F♭	D𝄪	E♮
5	G𝄫	E♯	F♮
6	G♭	E𝄪	F♯
7	A𝄫	F𝄪	G♮
8	A♭	G♯	NONE
9	B𝄫	G𝄪	A♮
10	C𝄫	A♯	B♭
11	C♭	A𝄪	B♮

Table 2. Mapping from the 'Flattest', 'Sharpest' and 'Natural' signifiers in Table 1 to explicit spellings of pitch classes 0-11.

variables (or bits). The assignment of 0's and 1's to these bits arises from the solution to a carefully constructed minimum cut problem instance.

3.1 Binary Encoding of Spelled Pitches

Slightly adjusting the approach taken in [5], possible spellings for a given pitch can be encoded by a pair of binary variables. The system of encoding summarized in Table 1 provides a significantly simplified approach, reducing the rule-set needed from five tables of cases to a single table of the same size [5]. I will, moreover, diverge from [5] by referring to the two bits used to encode a spelled pitch respectively as $\downarrow$ ("down") and $\uparrow$ ("up").

3.2 Constructing a Flow Network

We start with a collection of pitches that we wish to spell. We split each pitch into a $\downarrow$ and $\uparrow$ bit, and for each bit, we introduce a corresponding node into the flow network. Ending at each $\downarrow$ node is an arc from the source, and starting at each $\uparrow$ node is an arc to the sink. Further arcs are drawn between internal (non-sink and non-source) nodes with weights based on rules addressed in Section 3.2.3. Weights are assigned to the arcs based on empirically derived relationships between pitch classes, explored heuristically below and in greater depth in Section 4. Solving for the minimum cut gives us a 0 or 1 assignment for each $\uparrow$ and $\downarrow$. Reading off the encoding system via the mapping in Table 1 gives us a spelling for each pitch.

3.2.1 Arc-Weight Heuristics

To enhance the intuitive intelligibility of the model, we can give a heuristic derivation of relationships required between arc weights. To start, for pitches that permit a 𝄫 spelling and no 𝄪 spelling (pitch classes 0, 3, 5, 10) we ensure that the weight of the arc from $\uparrow$ to the sink is heavier than the weight of the arc from the source to $\downarrow$ (see Fig. 2). This way, the 𝄫 spelling is the most costly option to include in a cut. Analogously, for pitches that permit a 𝄪 spelling

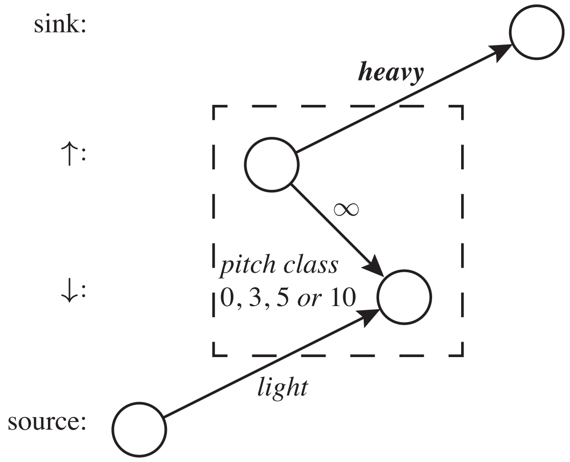


Figure 2. Visual representation of the relative arc weights needed in connecting the ↑ and ↓ nodes corresponding to pitches of class 0, 3, 5 or 10. The *heavy* and *light* arcs are discussed in Section 3.2.1, while the arc with weight ∞ is discussed in Section 3.2.2.

and no 𝄫 spelling (pitch classes 1, 4, 6, 11), we ensure that the weight of the arc from the source to ↓ is heavier than the weight of the arc from ↑ to the sink. For pitches that permit both a 𝄫 spelling and a 𝄪 spelling (pitch classes 2, 7, 9), we merely ensure that the weights from the source to ↓ and from ↑ to the sink are both relatively heavy so that the natural (indeed, ♮) spelling is almost always favored.

By penalizing double accidentals, we enforce the parsimony condition sought extensively in the literature. [1]

3.2.2 *Avoiding Invalid Encodings*

As Table 1 dictates, the assignment (↓: 1, ↑: 0) does not correspond to a valid spelling for any pitch. We add a further condition to our flow network construction to ensure that no pitch arrives at this state.

Between the ↑ and ↓ nodes of a *single pitch* in the flow network, we add an arc from ↑ to ↓ of weight 'infinity'. [2] Now if, for a given pitch, the ↑ node has state 0 and the ↓ has state 1, the total weight across the cut blows up (since the infinite weight gets included), and hence our cut is not minimal.

For pitches of pitch class 8 we can introduce an infinite weight arc in *both* directions, enforcing that the ↑ and ↓ nodes associated with the pitch class 8 pitch are always equal in value.

3.2.3 *Connecting Internal Nodes*

For each pair of pitches, we add arcs according to one of two cases.

1. We add arcs in both directions that connect the ↑ node of the first pitch to the ↑ node of the second pitch and likewise for the ↓ nodes.

[1] Penalizing accidentals directly in this way removes the need for a special 'parsimony pivot' as described in the original exposition [5].

[2] In our number system we include a value *infinity* that is greater than all numbers in the system but itself.

Pitch Class Pair	Lowest Cost Spellings
(0, 1)	(C♮, D♭) (B♯, C♯)
(0, 4)	(C♮, E♮)
(1, 3)	(C♯, D♯) (D♭, E♭)
(1, 5)	(C♯, E♯) (D♭, F♮)
(1, 10)	(C♯, B♮) (D♭, C♭)
(3, 4)	(D♯, E♮)
(3, 6)	(D♯, F♯) (E♭, G♭)
(3, 11)	(D♯, B♮) (E♭, C♭)
(4, 5)	(E♮, F♮)
(5, 6)	(E♯, F♯) (F♮, G♭)
(5, 11)	(F♮, B♮) (E♯, B♮) (F♮, C♭)
(6, 10)	(F♯, A♯) (G♭, B♭)
(10, 11)	(A♯, B♮) (B♭, C♭)

Table 3. Pitch class pairs for which ↑ node is connected to ↓ and vice versa.

2. We add arcs in both directions that connect the ↑ node of the first pitch to the ↓ node of the second pitch and vice versa.

Case 1 makes sense for all pairs of pitches where the respective natural spellings of the two pitches form a well-spelled interval and likewise for the respective sharpest spellings and flattest spellings (per Table 1). With appropriate weights added to the arcs, the ↓ nodes will tend to be coupled, and end up in the same state (unless there is a strong pull from another intervallic relationship) and likewise for the ↑ nodes.

Case 2 makes sense when we need to keep the pair of pitches away from problematic "natural" spellings per Table 2 (e.g. (C♮, C♯) as opposed to (C♮, D♭)). With the effects of sufficiently heavy source and sink incident arc weights, as discussed in Section 3.2.1, double spellings are avoided, and hence Table 3 gives the set of lowest cost spellings for pairs of pitch classes that should have ↑ nodes and ↓ nodes connected.

For pairs of pitch classes not listed in Table 3, case 1 applies (as can be checked exhaustively), and hence we connect the respective ↑ nodes to each other, and likewise for the respective ↓ nodes.

3.2.4 *Phantom Pitches*

To pull preferred spellings in one or another direction (more sharps, more flats, or more balance), we can add phantom pitches to the graph that do not come from the score in question, but nevertheless affect its spelling. The source and sink arcs of phantom nodes can be set to infinity so that these pitches are always spelled in their 'natural' form. Hence adding 0 as a phantom pitch would function the same as having an already-spelled C♮ among the unspelled pitches.

3.3 Injecting Musical Context

So far, we have only considered collections of pitches to spell in a musical vacuum, without temporal or cross-part information. This boils the problem of pitch spelling down

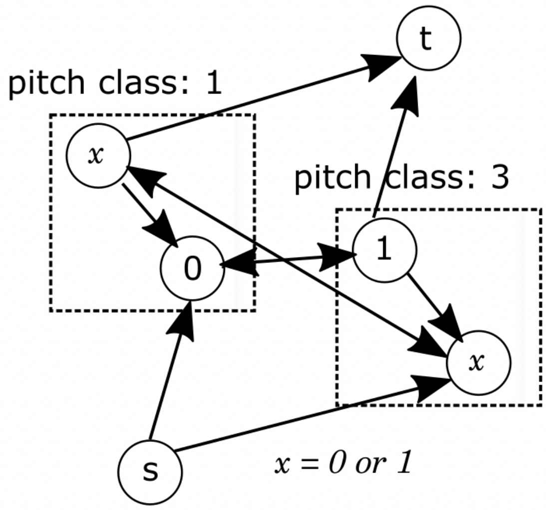


Figure 3. A fully connected network for spelling a pitch class 1 note against a pitch class 3 note. Arc weights need to be assigned such that the nodes fall on the sides of the cut indicated (as per Figure 1). The problem of finding the right arc weights is addressed in Section 4. When $x = 0$, the spelling (C♯, D♯) results. When $x = 1$, the spelling (D♭, E♭) results (cross-reference with Tables 1 and 2).

to its bare essentials of pitch relationships. But the model can also be easily be modified to factor in more contextual information, as would be appropriate in the spelling of a real score. By scaling arcs, we can increase or decrease the importance in certain interval relationships in determining the final spelling. In particular, scaling up the arcs between two pitches will increase the likelihood that those pitches are spelled harmoniously as an interval in a given context of other pitches.

3.3.1 Proximity Information

Pitches, as they occur in a score, can be independent from each other or exert a distant pull on each other. For pitches that should not be interrelated in terms of their spelling, [3] the arcs between their nodes can be removed from the flow network. For pitches that exert a distant pull, the arcs between their nodes can be scaled down appropriately, and indeed, adjacency in the score can be treated on a sliding scale whereby closer pitches have a larger scale factor less than 1 and more distant pitches have a smaller scale factor. For example, in [5], the adjacency scale factor – call it t_{pq} – for pitches in the same part, was implemented as an exponential factor for some base $\alpha > 1$:

$$t_{pq} = \alpha^{-(p-q)} \tag{1}$$

with $p \geq q \geq 0$ proportional to the sequential position of the two pitches in the score. A simple scale factor $0 < \beta < 1$ can also be added similarly, for pitches p and q that are simultaneous but in separate parts:

$$t'_{pq} = \beta \tag{2}$$

[3] Notes that should not be interrelated could include those either sids of a double barline, or simultaneous but in different sub-tonalities of a bitonal chord etc..

In a practical software implementation, adjacency information, or relevance of different pitches to each other across parts can be edited by a user, to provide an interactive, computer-*aided* spelling of a complex harmonic landscape. However, rules for removing arcs for pitches that should not exert an influence on each other are reasonable to generate. In [5], methods are proposed, such as treating double barlines and long rests as barriers between pitches, and removing any arcs for pitches on either side of these barriers.

4. INVERSE PROBLEM

While the existence of arcs based on pitch relationships can be determined heuristically, specific pitch-based weights can be determined through a solution to the *inverse problem*. The inverse problem poses the challenge of extracting arc-weight information from a collection of spelled pitches, rather than using the arc weights in the context of the flow network to derive assignments and hence spellings.

4.1 Full Complexity Solution

The original inverse problem solution is conceived by leveraging the primal-dual relationship of the maximum flow problem and the minimum cut problem. In linear optimization, a known result states that if a solution to the primal problem is equal in value to the value of a solution to the dual problem, then those solutions are optimal [10]. Both the maximum flow problem and the minimum cut problem can be stated as linear optimization problems, namely, with linear objective functions and linear constraints.

4.1.1 Defining Linear Constraints

For the purpose of defining the **maximum flow** problem, we assign to each arc a non-negative variable called the *flow* f_a of the arc a, bounded above by the weight w_a of a. The flow constraint for each node states that the sum of the flow variables for arcs going into the node should be equal to the sum of the flow variables for arcs going out from the node. The value of the total flow through the network as a whole is the sum of the flow variables for arcs going out from the source (or equivalently, going into the sink).

By satisfying flow constraints (in the maximum flow problem) and putting an upper bound B on the size of each pitch-based arc weight w_a, we can maximize the arc-weights to obtain a valid flow with non-zero arc-weights. By adding a duality constraint that sets the flow network's cut value equal to its total flow value, we ensure that the arc-weights found result in an optimal solution with pitches as they are found in the source score or corpus of scores.

From (3)-(8) we give a simplified statement of the constrained linear optimization labeled EXACTINVERSE in [5]. Contextual information, which can be injected into the inverse problem analogously to the approach discussed in Section 3.3, is omitted in this statement, for clarity and brevity. We define an *arc type* by the pitch classes and ↑/↓ types of its start and end node. To satisfy the model, we require a constraint, (6), that specifies that arcs with the same type have the same weight. Meanwhile, the objective in (3) is to maximize the sum of weights w_a over all combinations of pitch classes and ↑/↓ start and end nodes, indexed by a. (4), (5), (7) and (8) express the flow, duality and bounding constraints straightforwardly.

$$\max_{w_a, f_a} \sum_{\text{arcs types } \alpha} w_\alpha \tag{3}$$

subject to

flow constraints:

$$\sum_{\text{ingoing arcs } a} f_a = \sum_{\text{outgoing arcs } b} f_b \quad \text{for each node} \tag{4}$$

$$0 \leq f_a \leq w_a \quad \text{for each arc } a \tag{5}$$

model constraints:

$$w_a = w_b \quad \text{up to context-based scaling}^4$$

$$\text{for } a, b \text{ of the same arc type} \tag{6}$$

duality constraint:

$$\sum_{\text{source arcs: } s} f_s = \sum_{\text{cut arcs: } c} w_c \tag{7}$$

upper bound:

$$0 \leq w_a \leq B \quad \text{for each arc } a \tag{8}$$

[5] provides a mathematical proof that a solution of this optimization functions as an inverse to the pitch spelling problem; that is, once weights are extracted using the optimization described in (3)-(8), solving for the minimum cut *using those weights* will once again return the same set of spellings.

The simplex algorithm is, in practical computations, extremely efficient, and can be used to solve linear optimization problems, or linear programs, of the form of (3)-(8) [10]. Hence, in cases where the intervallic spelling rules of a score or corpus are consistent, we can compute a set of weights for constructing pitch spelling flow networks efficiently.

4.1.2 Robustness Through Approximation

[5] also provides an *approximate* solution to the inverse problem, as the existence of a set of weights consistent with the spelling of a given corpus of music is not guaranteed. By loosening the duality constraint and adjusting a constant λ to set how strict an inverse we seek, we can guarantee an output even for a corpus with inconsistent intervallic spelling.

$$\max_{w_a, f_a, \Delta} \sum_{\text{arcs types } \alpha} w_\alpha - \lambda\Delta \tag{9}$$

subject to

flow constraints:

$$\sum_{\text{ingoing arcs } a} f_a = \sum_{\text{outgoing arcs } b} f_b \quad \text{for each node} \tag{10}$$

$$0 \leq f_a \leq w_a \quad \text{for each arc } a \tag{11}$$

model constraints:

$$w_a = w_b \quad \text{up to context-based scaling}^5$$

$$\text{for } a, b \text{ of the same arc type} \tag{12}$$

approx. duality:

$$\sum_{\text{source arcs: } s} f_s \geq \sum_{\text{cut arcs: } c} w_c - \Delta \tag{13}$$

upper bound:

$$0 \leq w_a \leq B \quad \text{for each arc } a \tag{14}$$

Indeed, the Δ value provides an interesting measure on *how consistent* in terms of the pitch spelling model the spelling of a corpus of music is. By adding a further constraint,

$$\Delta \leq pB \tag{15}$$

for some constant $0 < p < 1$, we can test whether the corpus of music can produce a duality of gap of less than or equal to pB, at which point, according to the terminology of [5] we could say that the corpus of music is 'p-consistent' in spelling. B is the upper bound on arc weights, as given in (15).

Constants such as λ, α and β in (1), (2) and (9) can be characterized empirically through repeated trials, using a systematic 'grid-search' method or random hyper parameter search method (see [11, 12]). Work is underway to characterize these constants empirically. [6]

4.2 Practical Inverse

In the production of practical software, a reliance on the simplex algorithm for inverse solutions can be limiting. The fastest simplex implementations tend to be commercial or, at the opposite extreme, use 'copyleft' licensing, with each paradigm placing restrictions on how the pitch spelling algorithm can be distributed and deployed downstream [13]. The optimization of a simplex algorithm, moreover, is highly technical, making it especially difficult to effectively implement without the required background in specialized operations research. In learning from a corpus of music, moreover, it is hard to guarantee that there is enough saturation of all the intervals to generate a representative set of arc weights. As a result, this paper proposes a more contained implementation for practically generating a set of arc weights with lower software production and data wrangling overheads.

[4] per Section 3.3

[5] per Section 3.3

[6] A work-in-progress Python implementation with these aims can be found here: https://github.com/bwetherfield/pitchspell

The principle of this inverse problem approach is to generate arc weights not from a corpus of scores, but rather from a collection of spelled dyads, triads and/or larger groups of pitches. Mirroring the context-free approach to pitch spelling in the 'forward' direction, we generate inverses only from pitch relationships.

We define an **unweighted network** to be a flow network where the arcs do not have weights. Using a collection of spelled groups of pitches, we can construct an unweighted network with the use of the pitch spelling model outlined in Section 3. Nodes are given values 0 or 1 according to the encoding rules in Tables 1 and 2. We stipulate the following condition.

Condition 1. Flattest spellings and sharpest spellings should never be adjacent in inputs to the practical inverse.

For example, we cannot feed in a chord containing both D♭ and B♯.

4.2.1 Adapting the Edmonds-Karp Algorithm

The Edmonds-Karp Algorithm solves for the maximum flow of a network. As we have noted, the maximum flow problem is the dual of the minimum cut, and, as such, the minimum cut solution can be obtained easily from a solved maximum flow. I will here give a brief description of the Edmonds-Karp algorithm and then explain how it can be modified for the purposes of performing the first part of an inverse solving approach.

In the Edmonds-Karp algorithm, we begin by finding the shortest path from source to sink such that all arcs in the path have *residual capacity*; the path must trace nodes connected in the network, either forward or backward across arcs. A backward arc has residual capacity if its current flow value is nonzero. A forward arc has residual capacity if its flow value is strictly less than its weight. Having found a path, we *push flow* through it. To push flow, we increase the flow of all forward arcs and decrease the flow of all backward arcs in the path by the same amount. In the algorithm, we push flow equal to the *minimum residual capacity* in the path, namely the minimum difference between the upper bound of a forward arc or the lower bound of a backward arc and its respective flow. Figure 4 shows a possible setup. When the flow of an arc equals its weight, we say the arc is *saturated*. When an arc is saturated, we remove it and insert a reversed arc of the same weight in its place. In the course of the algorithm, another shortest path with flow capacity is now found and we iterate until no shortest path with flow capacity remains. It can be shown that the process terminates after at most as many iterations as there are nodes [9].

FILLSTORAGE (Fig 5) is a modified Edmonds-Karp algorithm that operates on the unweighted network constructed in Section 4.2. By analogy with the Edmonds-Karp algorithm, we iterate on finding the shortest available path from source to sink. For each *arc type* (characterized by the pitch classes and ↑/↓ of its endpoints), we store a set of other arc types. Each arc type's weight will have to exceed or equal the weights of the arc types in its storage. We find the arc in the current path for which the start node has value 0 and the end node has value 1. As we will prove below

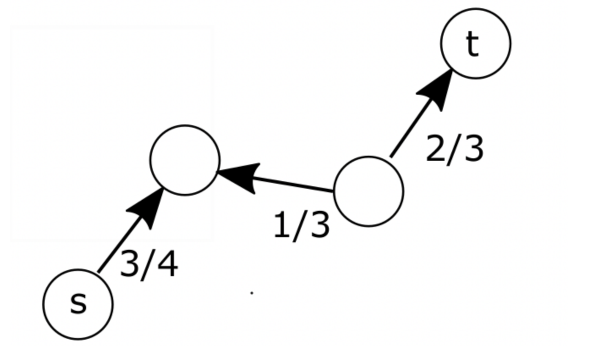


Figure 4. A path in a residual network. 3/4 represents a flow of 3 and a capacity of 4 for the given arc. The above path can have 1 unit of flow pushed through it, as each of the arcs has a residual capacity of 1. The first and last arcs will be saturated and replaced by reversed arcs with 0 flow, and respectively 4 and 3 as capacity.

(Claim 2), by the construction of the network, there will only be one such arc for each path found in the algorithm. For each path, we add this $0 \to 1$ arc's type to the STORAGE corresponding to all the other arc types present in the path. As in the Edmonds-Karp algorithm, we remove the 'saturated' arc and insert a reversed copy.

Claim 1. *The following arcs cannot be taken in paths found by the modified Edmonds-Karp algorithm:*

1. *↓ with value 1 to ↓ with value 0*
2. *↑ with value 1 to ↑ with value 0*
3. *↑ with value 1 to ↓ with value 0*
4. *↓ with value 1 to ↑ with value 0*

Proof.

1. There is a shorter path directly through the node with value 0.
2. There is a shorter path through the first node directly to the sink.
3. There is a shorter path through the first node directly to the sink.
4. The first node must belong to a (↑: 1, ↓: 1) encoding, and the second to a (↑: 0, ↓: 0), hence a sharpest spelling and a flattest spelling are connected, violating Condition 1.

□

Claim 2. *There is exactly one arc from a value 0 node to a value 1 node in each path found by the modified Edmonds-Karp algorithm described.*

Proof. No arc with values $1 \to 0$ can occur between internal nodes by Claim 1. Nor can a source or sink arc have weights $1 \to 0$ as source and sink always have value 0 and 1 respectively. Since there is no $1 \to 0$ arc in the path, and since a $0 \to 1$ arc must appear between source and sink, there is exactly one $0 \to 1$ arc in each path found by the algorithm. □

```
function FILLSTORAGE
    STORAGE(a) ← an empty set for each arc type a
    while p ← AUGMENTINGPATH do
        for a in arcs of p do
            if a is a "0 → 1" arc then
                remove a from the network
                insert REVERSED(a) into the network
                for b ≠ a in p do
                    append type of a to STORAGE(b)
                end for
            end if
        end for
    end while
    return STORAGE
end function

function AUGMENTINGPATH
    return shortest path from source to sink
end function
```

Figure 5. Modified Edmonds-Karp algorithm used to populate the storage assigned to each arc type present in the network.

4.2.2 *Concrete Weight Generation Steps*

For each arc we have stored the set of arcs on which its weight must depend. This forms a linked directed graph structure (where the arcs are playing the role of nodes in the new directed graph structure). In the MAIN function loop (Fig. 6), we can use standard graph algorithms to detect cycles of dependencies (for instance, a modified depth first search), and consolidate arcs in the storage into groups of connected components as needed (using Tarjan's algorithm) [7]. Now the storage is free of cyclic dependencies, we can recursively ensure each arc type is greater than the sum of all arc types in its storage, by adding 1 at each level of the recursion. If Tarjan's algorithm was called, we need to undo the grouping of arcs with UNDAGIFY, giving each arc in the same group the arc weight that was computed for that group. Since all arcs in the same group have the same arc weight, they are all greater than *or equal to* each other in weight, as needed.

4.2.3 *Analysis of Correctness*

We now imagine constructing a flow network from the arc weights derived in the inverse procedure. The flow network represents the same pitches in the same relationships, so it has the same adjacency structure as the unweighted network used to generate weights. We wish to show that it can generate a spelling consistent with the input spelled pitches.

To show the consistency of the flow network with the input pitches, note that we can run the Edmonds-Karp algorithm and follow the same sequence of paths found in the inverse procedure, which we will refer to as the *route*. If there are no cyclic dependencies in the storage structure, then the order of cut arcs being saturated and reversed follows the order of the inverse process, by construction. Hence the desired minimum cut is found.

```
function MAIN
    STORAGE ← FILLSTORAGE
    if DETECTCYCLE then
        DAGIFY
        GENERATEWEIGHTS
        UNDAGIFY
    end if
    GENERATEWEIGHTS
    return STORAGE
end function

procedure GENERATEWEIGHTS
    map over STORAGE calling WEIGHT on arcs, or
    arc groups if DAGIFY has been called
    function WEIGHT(a)
        if WEIGHT(a) has not already been called then
            w ← 0
            for b in STORAGE(a) do
                add WEIGHT(b) to w
            end for
            STORAGE(a) ← w + 1
        end if
    end function
end procedure

function DETECTCYCLE
    perform a cycle detection on STORAGE
end function

procedure DAGIFY
    group arcs in STORAGE that have cyclic dependen-
    cies using Tarjan's algorithm for finding strongly con-
    nected components
end procedure

procedure UNDAGIFY
    unwrap arc groups so that STORAGE maps from
    arcs to weights instead of arc groups to weights.
end procedure
```

Figure 6. The entire practical inverse procedure, including a call to the modified Edmonds-Karp algorithm.

When there is a cyclic dependency, we can show, by a contradiction argument that the forward solver will find at least one consistent minimum cut. If not, there would be a path in the faulty forward solution that contained an unseen $0 \to 1$ arc, which contradicts the construction of the inverse.

Informal experiments have shown that all three of the following methods can increase the robustness of an inverse solution, such that *all* minimum cuts that were fed in can be recovered with a forward solver.

1. Supplying more input spelled collections to the inverter, including, for example, all correctly spelled dyads.

2. "Warm starting" the STORAGE data structure with preset arc type dependencies before running FILLSTORAGE, using the arc heuristics in Section 3.2.1

to enforce the equality of certain pairs of "heavy" and "light" weights.

3. Supplying a suitable phantom pitch set to the forward problem instance (see Section 3.2.4).

4.2.4 Time Complexity

For the first part of the algorithm, the time complexity can be reduced to the time complexity of the Edmonds-Karp algorithm, which is $O(VE^2)$ – as it is described in this paper – where V denotes the number of nodes in the network, and E the number of arcs [9]. For the second part of the algorithm, note that the number of arc types is fixed, and so the number of steps needed is bounded (albeit by a large bound). Though at practical scales, the impact of the second half of the algorithm may be felt, its contribution in asymptotic terms is constant. Hence, we can fully characterize the complexity of the algorithm by that of Edmonds-Karp.

4.2.5 Simple Set of Results

Running the Practical Inverse procedure against an exhaustive list of well-spelled dyads, we obtain the arc weights laid out in tables 4, 5, 6 and 7. The exhaustive set of "good" input spellings contains all plausible spellings of semitones ([C, D♭], [C♯, D], and so on), tones ([C, D], [C♯, D♯] *and* [D♭, E♭], and so on), minor thirds, major thirds and perfect fourths *that do not contain any double accidentals.*[7]

4.2.6 Insufficiency of Results

It is worthy of note that Table 4, unlike Table 5, only features 0's and 1's. Where $\downarrow$ nodes are connected only to $\downarrow$ nodes and $\uparrow$ nodes are connected only to $\uparrow$ in the construction of a spelling network, the minimum cut set is *empty* as the source and sink are already strongly disconnected! Hence, the algorithm runs only trivially, adding 1 to internal connected arcs only once. Moreover, the (0, 0)-(11, 11) diagonal contains only 0's. The set of dyads used lacks unisons ([C, C], [C♯, C♯] and so on), to fill in this diagonal, in part to cope with limitations in the implementation. At best, with unisons included, however, we would have seen a diagonal full of 1's, which does not give proper weight to these pitched spelling relationships.

Intuitively, the size of the entries in Table 4 should reflect how important it is for pitch classes to be spelled in the same direction when side by side. Hence, it encodes the pull of pitch class 10 to B♭ when near a C♮, or conversely, to A♯ when near a B♯.

Analysis of these results demonstrate, therefore, the insufficiency of dyads alone for characterizing all arc weights in the inverse problem. The dyads prove adequate, however, for inducing differentiation for *half* of the internal arc weights, along with the outer weights to and from the source and sink nodes.

[7] These results were obtained using an open source Swift implementation hosted on github: https://github.com/bwetherfield/PitchSpellingModel

pc	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	1	1	0	1	0	1	0	1	1	1
1	0	0	1	0	1	0	1	0	1	1	0	1
2	1	1	0	1	1	1	1	1	0	1	1	1
3	1	0	1	0	0	1	0	1	0	0	1	0
4	0	1	1	0	0	0	1	1	1	1	0	1
5	1	0	1	1	0	0	0	1	0	1	1	0
6	0	1	1	0	1	0	0	1	1	1	0	1
7	1	0	1	1	1	1	1	0	0	1	1	1
8	0	1	0	0	1	0	1	0	0	1	0	1
9	1	1	1	0	1	1	1	1	1	0	1	1
10	1	0	1	1	0	1	0	1	0	1	0	0
11	1	1	1	0	1	0	1	1	1	1	0	0

Table 4. Empirical arc weights derived for $(\downarrow, \downarrow)$ arc types. The empirical results for $(\uparrow, \uparrow)$ arc types are identical, thanks to the symmetry of the input set of spellings.

pc	0	1	2	3	4	5	6	7	8	9	10	11
0	1	2	0	0	1	0	0	0	2	0	0	0
1	1	1	0	1	0	1	0	0	0	0	1	0
2	0	0	1	0	0	0	0	0	0	0	0	0
3	0	3	0	1	2	0	3	0	2	0	0	2
4	1	0	0	1	1	1	0	0	0	0	0	0
5	0	2	0	0	1	1	2	0	2	0	0	0
6	0	0	0	1	0	1	1	0	0	0	1	0
7	0	0	0	0	0	0	0	1	2	0	0	0
8	2	0	0	2	0	2	0	2	1	0	3	0
9	0	0	0	0	0	0	0	0	0	1	0	0
10	0	3	0	0	0	0	3	0	3	0	1	2
11	0	0	0	1	0	0	0	0	0	0	1	1

Table 5. Empirical arc weights derived for $(\downarrow, \uparrow)$ arc types. The empirical results for $(\uparrow, \downarrow)$ arc types are the exact matrix transposition (i.e. reflection along the (0,0)-(11,11) diagonal) thanks to the symmetry of the input set of spellings.

pc	0	1	2	3	4	5	6	7	8	9	10	11
	13	26	3	1	13	13	26	3	0	3	1	13

Table 6. Empirical arc weights derived for (source, $\downarrow$) arc types.

pc	0	1	2	3	4	5	6	7	8	9	10	11
	13	1	3	26	13	13	1	3	0	3	26	13

Table 7. Empirical arc weights derived for ($\uparrow$, sink) arc types.

5. CONCLUSIONS AND FURTHER WORK

This paper has summarized the structure and composition of the minimum cut pitch spelling algorithm, while presenting some theoretical simplifications and extensions to the original exposition given in [5]. Clarifications to the system of encoding for pitch spellings, along with the decoupling of context information from the simple pitch relationships allow us to reason more directly about the core functionality of the algorithm. The new inverse problem approach, moreover, reduces the potentially unwieldy dependency on the simplex algorithm in generating sensible arc weights for the model.

There is still plenty of work to do before the algorithm presented is practically useful in production software. More heuristic methods, or larger spelling sets, are needed to populate the full set of pitch-based arc weights (in response to the limitations described in Section 4.2.6). Moreover, an empirical study on a large corpus of scores is still needed to tune hyper-parameters (as mentioned in Section 4.1.2), check the validity of heuristic measures (such as those described in Section 4.2.3) and compare the accuracy of this and other algorithms on corpora of canonical scores.

6. REFERENCES

[1] D. Meredith, "Comparing pitch spelling algorithms on a large corpus of tonal music," *Computer Music Modeling And Retrieval*, vol. 3310, pp. 173–192, 2005.

[2] E. Chew and Y.-C. Chen, "Real-Time Pitch Spelling Using the Spiral Array," *Computer Music Journal*, vol. 29, no. 2, pp. 61–76, June 2005.

[3] A. K. Honingh, "Compactness in the Euler-Lattice: A Parsimonious Pitch Spelling Model," *Musicae Scientiae*, vol. 13, no. 1, pp. 117–138, March 2009.

[4] E. Cambouropoulos, "A general pitch interval representation: Theory and applications," *Journal of New Music Research*, vol. 25, no. 3, pp. 231–251, September 1996.

[5] B. Wetherfield, "A graphical theory of musical pitch spelling," Bachelor's Thesis, Harvard University, 2017. [Online]. Available: https://dash.harvard.edu/handle/1/38779539

[6] D. Meredith, "Optimizing Chew and Chen's Pitch-Spelling Algorithm," *Computer Music Journal*, vol. 31, no. 2, pp. 54–72, June 2007.

[7] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972. [Online]. Available: https://doi.org/10.1137/0201010

[8] D. S. Hochbaum, "The pseudoflow algorithm: A new algorithm for the maximum-flow problem," *Operations Research*, vol. 56, no. 4, pp. 992–1009, August 2008.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[10] D. Bertsimas, *Introduction to linear optimization*, ser. Athena Scientific series in optimization and neural computation. Belmont, Mass.: Athena Scientific, 1997.

[11] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," *CoRR*, vol. abs/1309.0238, 2013. [Online]. Available: http://arxiv.org/abs/1309.0238

[12] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/jmlr/jmlr13.html#BergstraB12

[13] J. L. Gearhart, K. L. Adair, R. J. Detry, J. D. Durfee, K. A. Jones, and N. Martin, "Comparison of open-source linear programming solvers."

SUPER COLLIDERS: A GAMIFIED SCREEN-SCORE

Takuto Fukuda
McGill University, CIRMMT
takuto.fukuda@mail.mcgill.ca

ABSTRACT

This paper introduces *Super Colliders* (2018), a piece written for three pitched instruments and a computer. This piece applies gamification to the screen-score as a compositional approach to achieve playful human-computer interactions. The piece features a game design that encompasses various game mechanics and elements. The paper describes the technical details of the game's design, the role and effects of the featured game elements from the perspective of motivational affordances. Finally, through the analysis of a performance of the piece, the paper reveals how motivational affordances in the screen-score supported to generate the musical structure through the playful performer-computer interactions.

1. INTRODUCTION

In recent years, there has been increasing interest in gamification in human-computer interaction studies. The term 'gamification' first appeared in a blog post written by Brett Trill [1] in 2008 and is defined as "the use of game design elements in non-game contexts" [2] to evoke users' playfulness and address specific challenges. This approach has been applied to, for example, educational software design to motivate learners. Although this approach primarily aims to increase user engagement with difficult challenges, gamification drew the author's attention as a design method for human-computer interaction (HCI) in interactive computer music composition.

Interactive computer music refers to "a music composition or improvisation where software interprets a live performance to affect music generated or modified by computers" [3]. In this field of music, HCI has been used to incorporate the inherent variability in human performance into various projects [4] while posing a question about what the design of interactive systems that can listen, interpret, compose, and respond to a human performer in a way to make sense could be [3] [1]. Numerous precursors have addressed this question, such as George Lewis's *Voyager*, which is a computer program designed for improvising in response to what a computer hears during a performance [6]. Joel Chadabe developed *interactive composing*, a compositional method during which he performs with his self-built 'intelligent' instruments that give quasi-unpredictable responses to a human performer [7]. These precursors have compelled "a paradigm shift from interactive composing [...] to composing interactions" [8]. Designing interactive systems has become a central issue in the compositional process.

This paradigm shift raises a question about what the optimal notation systems are: while interactive systems need to be responsive to performers' actions, most traditional musical scores are prescriptive and represent a predetermined course of music. The screen-score concept was developed, in part, as a solution to this problem of fixity. Thus, screen-scores are often designed to not only project a score on a video screen but also generate musical symbols, graphics, or performance instructions in real-time, as in *KOMA* (G.E. Winkler, 1996) [9] and *music for 2* (D. Kim-Boyle, 2010) [10].

The precursors revealed the problem of an extreme sight-reading. Performers seldom have the opportunity to study a screen-score prior to the performance as the system often generates a unique version of the score in real-time during every performance, and most screen-scores show musical symbols or graphics for only a few seconds on a video display during the performance. These limitations situate the performers against the risk of misinterpretation [11], which conflicts with the concept of perfection prominent in an age where the proliferation of music records has heightened audience engagement with the reproduction of recorded performances in concerts [12].

This cultural condition highlights the game aspect inherent in music performance: the player's challenge to achieve perfection, that is, a game of either success or failure. *Super Colliders* explores an alternative approach to turn the risk of failure into an engaging performance, embracing the game aspect of the music performance by combining gamification with a screen-score in interactive system design. Gamification can create a performance ecosystem in which mistakes play a meaningful role in engendering playful HCI, while a screen-score can mediate performer-computer communication through musical symbols or graphics in a way that is perceptible to the audience [2].

[1] A comprehensive overview of the discussions on interactive systems can be found in [5].

[2] Although the concept of gamification emerged only recently, the merger of games and music was attempted in 'Game Piece,' a musical composition without a predetermined course, like most pieces of classical music, but determined in real-time according to rules, chance operations, and competitive engagement between performing opponents toward a goal, similar to sports and video games, as in *Cobra* (J. Zorn, 1984) [13] [14] and *Duel* (I. Xenakis, 1959) [15]. In recent years, the term 'gamification' has been applied to various pieces, such as *Contraction Point* (K. Giannoutakis, 2015) [16] and *Game Over* (C. Ressi, 2017)

This author's vision of integrating gamification with a screen-score raise two additional questions: which design elements engender playful performer-computer interactions in a piece using a gamified screen-score, and what musical structures and components emerge from these interactions? In this paper, the author reveals the design elements and their musical effects through the analysis of the game elements and a performance. The paper introduces the piece in the following order: (1) game design, (2) the identification of game elements from the perspective of motivational affordances, and (3) the analysis of a performance.

Additionally, the piece is a reflection of the author's artistic point of view, which is to integrate the enthusiasm of game-play with historically inherited compositional devices (e.g., fugato in counterpoint, modulations in tonal music, cyclic form predominantly in romantic music, etc.) into a single composition. To do this, the author introduced two approaches, (1) the use of a midi sequence data of Invention No. 1, BWV 772 (J.S. Bach) to determine initial contents on the screen-score, and (2) the use of a second-order Markov chain algorithm to transform the initial contents according to the performers' interventions. The first approach aimed to introduce the contrapuntal compositional devices in the Bach's music into the initial contents on the screen-score, thereby enriching the musical texture. The second approach was used to transform the initial contents on the screen-score according to the performers' enthusiastic game-play. Applying the two approaches, the piece was envisioned to unfold a course of music, in which, performer's interventions gradually overwrite the Bach Invention.

In Section 2, the game design is explained, including the game mechanisms (i.e., goals and rules) and actions of gaming components (i.e., performers, the gamified screen-score, the interactive computer system, and the second-order Markov chain algorithm). Section 3 identifies elements that can afford players' motivational needs during a performance in light of a taxonomy of motivational affordances as an analytical framework. Section 4 focuses on the observation of musical structures and materials emerging from the interactions between performers and the identified elements. Section 5 presents an evaluation of the piece in light of the author's artistic objective, and future works to improve the game design for enriching the musical structure. In the conclusions section, the author emphasizes the importance of optimizing the difficulty level of challenges (i.e., blob behaviors) during a performance and the need to tune the Markov chain algorithm more effectively so that more engaging playful performer-computer interactions can be achieved.

2. GAME DESIGN IN *SUPER COLLIDERS*

Super Colliders is a game piece for three pitched instrumentalists or vocals with an audiovisual projection setup. The author aimed to design the piece in such a way that the player's desire to win the game results in an enriching musical performance. Therefore, while some music video games, such as *Guitar Hero* [18] and *Dance Dance Revolution*, are played by the haptic manipulation of a game controller, in this composition, the game is played by sound. A key approach was gamifying the 'scrolling score and fixed playhead' model in the taxonomy of animated scores [19]. This approach enabled the author to design a performance ecosystem in which the interactive computer system poses challenges as musical symbols, which the players must interact with to perform (Figure 1).

Indeterminacy is introduced to several levels as the instrumentation, duration, and detailed figure of contours that musicians play. There is no prescribed score or fixed musical sequence. Although performance instruction is provided, it does not specify when and what the performers must play during the game. It informs them of the general objective of the game and the attitude needed to meet challenges imposed by the interactive computer system. Performers are free to choose several musical parameters such as the desired playing technique, contour type, and dynamics for their performance.

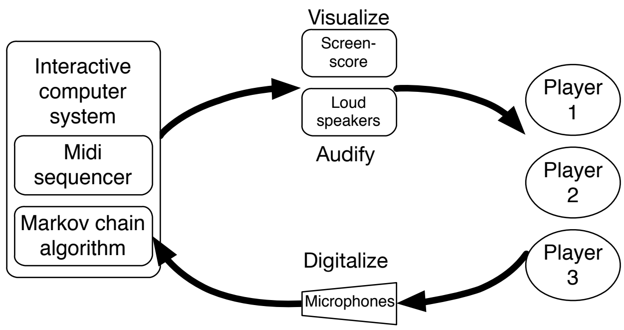


Figure 1. Performance ecosystem in which the interactive computer system and players communicate using a screen-score

2.1 Goals and rules

The players compete with each other in four rounds of the game. One grand winner is identified at the end of the game. To be the grand winner, a player must win the most rounds.

Players must earn 1,080 life points faster than other players to win each of the four rounds. They clash their avatars with blobs moving continually from left to right on the screen-score to earn life points. The players begin the game with 540 life points.

Players earn one life point per clash. However, all three players lose a point if all of them miss a clash. Therefore, there are two possible consequences of the game. One scenario is that one musician earns the most points, so he or she wins the round. Another option is an 'all dead' scenario, meaning all the musicians lose and the computer wins. Notably, the third round is designed such that a human performer always wins.

[17] in the field of interactive computer music. These attempts use a computer as an opponent against the human player in a game.

2.2 Gamified screen-score

The term screen-score refers to the graphical or symbolic representation of music projected visually on a screen. This model typically shows musical notes (or symbols that represent musical information) moving from right to left and a fixed playhead. The notes are played when they overlap on the playhead.

In this piece, the 'scrolling score and fixed playhead' model [19] is gamified as mentioned previously. The musical notes on the scrolling score are replaced by moving blobs (orange rectangles and grey dots) that are representations of midi notes and 'targets,' which players must clash with their avatars, replacing the fixed playhead.

The gamified screen-score shows three types of elements: three musicians' avatars, three life-point indicators, and moving blobs (Figure 2). The three avatars are represented by three dots in different colors. Each avatar is labeled with a musician's name and positioned on the right side on the screen. The three life-point indicators represent the players' life points, and each indicator elongates to the top of the screen as the performer earns life points and drops toward the bottom of the screen as the player loses life points. The moving blobs represent musical notes in a midi file, as well as 'targets,' with which the avatars must collide to earn life points in the game.

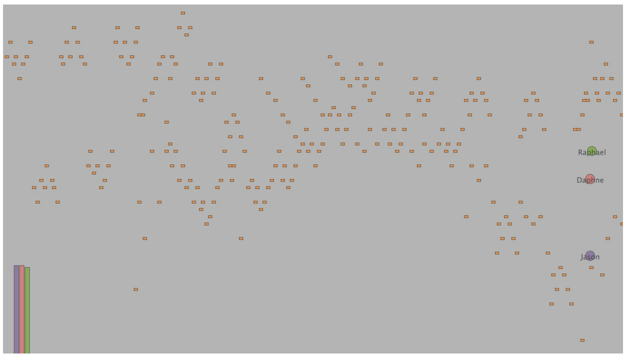


Figure 2. Screen graphics showing three avatars on the right, three life-point indicators on the left, and target blobs moving from left to right.

The vertical position of each blob is mapped to the pitch parameter. The blobs appear in the order of musical notes in a midi sequence in every round, and the blobs move continuously from left to right except during the third round. During the third round, the blobs bounce up, down, left, and right. Although the blobs are still mapped to musical notes to determine the initial vertical position by the computer, the bouncing behavior makes it more difficult to associate the blobs with particular pitches.

2.3 How performers play the game

The piece allows two alternative setups. Either the musicians face the main gamified screen-score with their backs to the audience, or the players face the audience and look at monitor displays (Figure 3).

The performers control their avatars by performing ascending or descending glissandi quietly or loudly. The pitch change is mapped to the vertical position of the performers' avatars. Avatars move to the upper edge of the screen when the musicians play ascending pitch changes and to the bottom of the screen when they play descending pitch changes. The performers need to play the optimal length of the glissandi in the desired direction (up or down) at appropriate timings to adjust the avatar's position to the location of their target blobs.

Loudness is mapped to the horizontal position and the size of the avatars. The avatars are aligned on the right side of the screen when the performer plays silent or very quietly. The speed at which avatars move from right to left depends on how loudly the musicians play. The performers need to play loudly to clash against a blob with their avatar before other players hit it. The size of avatars enlarges as the performers play louder. The enlarged surface allows the avatar to clash with more blobs. This feature is intended to motivate players to perform louder so they can earn more life points.

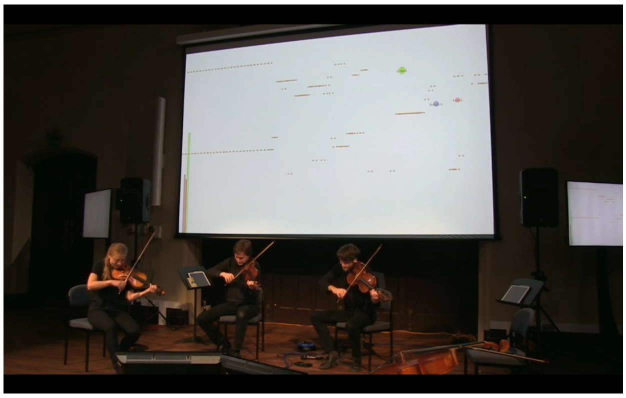


Figure 3. Three string players facing a small monitor display with their backs to the main screen-score.

2.4 Interactive computer system design

The interactive computer system is effective when optimizing the challenges' difficulty to the commensurate level with the performer's skills. The computer system consists of two components: a midi sequencer and a second-order Markov chain algorithm. The midi sequencer sows the initial seed challenges, and the algorithm monitors the performer's ability and adjusts the difficulty level of subsequent challenges in real-time. The midi sequencer outputs midi notes from a preprogrammed midi sequence of the Bach's intention No. 1 as elements of the challenges. The system visualizes the midi notes as blobs on the screen-score.

The second-order Markov chain algorithm plays a key role in optimizing blob behaviors. These behaviors have a significant impact on the level of challenges posed to the performers. The algorithm continues tailoring the difficulty level according to the performers' history of successes since the beginning of the performance.

Another important role of the Markov chain algorithm is a contribution to achieving the author's artistic aim, performers' interventions that gradually override the Bach

Invention, mentioned in the end of Section 1. As the algorithm continues tailoring the difficulty level of the challenge, so does the algorithm overwrites the midi sequence of the Bach Invention with a more dissimilar sequence from the original.

The Markov chain algorithm's behavior changes in every round. During round one, the algorithm learns the moving blobs representing the musical notes in the Bach's intention No. 1. During the performance of the first round, the algorithm creates a state transition matrix (STM) that stores a weight of the probability of every pitch progression between three subsequently clashed midi notes, represented by blobs on the screen-score.

In the following rounds, the algorithm performs two tasks simultaneously: generating a blob sequence according to weighted random choices of the pitch progressions entered in the STM and renewing the weight of the probability based on the newly detected pitch progressions, represented by blob collisions. Notably, the STM is not flushed after every round but maintained for further renewal in subsequent rounds.

Progressions are detected in three different ways. During rounds one and three, they are detected when the moving blobs on the screen-score collide with avatars. During round two, they are detected when the moving blobs are missed by all avatars. During round four, they are detected when the blobs are intercepted and missed by avatars.

2.5 Implementation

A game system was implemented in Cycling '74's Max for audio signal processing and the playback of prerecorded sound files, as well as in the Processing programming environment for real-time visual processing. The system captures acoustic sounds from microphones and then streams the sounds to Max as separate audio signals in real-time. In a Max program, a sigmund˜ object tracks the pitch change, and a peakamp˜ object detects amplitude. The detected pitch changes and amplitude are mapped to the vertical and horizontal position of avatars, respectively, in a processing program.

The prerecorded sound files are classified into two different sound types: a drone sound and fragmentary sounds. While the drone sound is played in the background throughout all the rounds, the fragmentary sounds are triggered in response to each collision between the avatars and blobs. The fragmentary sounds are further subcategorized into two different types of sounds: piano-like sounds and synthetic attacks. The piano-like sounds are used to play a single pitch that each collided blob represents, thereby possibly reproducing the Bach Invention if all the blobs are collided in the order they appear in the first round. The synthetic attack is triggered each time a collision occurs, so that the synthetic attack sounds give the players and audience audible feedback of the collisions.

The piano-like attack sound was created by EXS24, a built-in synthesizer on the Logic digital work station. The synthetic attack sounds and drone sound were created by TAL-NoiseMaker, a VST plugin synthesizer used on Logic.

3. IDENTIFICATION OF GAME ELEMENTS FROM THE PERSPECTIVE OF MOTIVATIONAL AFFORDANCES

Zhang's article [20] proposes effective design principles for enjoyable human-computer interactions. According to the article, enjoyable human-computer interactions emerge when players' *motivational needs* are fulfilled by the *motivational affordances* of featured elements (e.g., life points and avatars) in a game. The term *motivational affordances* refers to "the properties of an object that determine whether and how it can support one's motivational needs." [20]. *Motivational needs* are users' psychological or social desires (e.g., autonomy and competence) that they want to be fulfilled. This hypothesis was applied in the context of gamification by Weiser *et al.* [21].

To this end, the author applies a taxonomy of motivational affordances [21] as an analytical framework to identify properties of concrete elements (e.g., life points and blob behaviors) in *Super Colliders* and clarifies what motivational needs these elements afford.

3.1 Motivational needs

Taxonomical research [20] and [21] identified the following motivational needs:

- *Autonomy* - the desire to make decisions by themselves rather than being forced to follow a particular regulatory guideline;
- *Competence* - the desire to acquire better skills through challenges that are "neither too easy (boredom) nor too difficult (frustration)" [21];
- *Relatedness* - the desire to engage with others through, for example, recognition, acceptance, and being valued;
- *Achievement* - the desire to demonstrate one's competence to others;
- *Affiliation and Intimacy* - the need for other people's approval and the inclination toward secure and rewarding relationships, respectively;
- *Leadership and Followership* - the desire to gain authority and impact, control, and influence others and the desire to support or be subordinate to a leader, respectively.

3.2 *Mechanics*

When *mechanics* help meet the aforementioned motivational needs, users perceive the experience as playful while interacting with a system. Therefore, these components are particularly relevant to interactive system design. The aforementioned study [21] identified six *mechanics*, which are defined as "possible means of interaction between a user and the system" that can help meet motivational needs.

- *Feedback* - visual and aural information about the user's current actions. This mechanic can optimize users' actions and increases their motivation to achieve a goal.

- *User education* - advice that compensates for the user's lack of knowledge and helps achieve a goal. This mechanic can fulfill the need for competence and, to some extent, satisfy the need for followership.

- *Challenges* - something difficult to overcome, such as a task or quest. This mechanic fulfills the desire for competence.

- *Rewards* - something valuable (e.g., life points or money) given in exchange for the user's accomplishment. Rewards can satisfy the need for achievement and competence.

- *Competition and comparison* - competition is a situation where a player has to win a challenge against rivals. This mechanic can fulfill the player's need for achievement and leadership. Competition is often between players.

- *Cooperation* - collaborative action with other players to achieve a goal. This mechanic can fulfill the desire for affiliation and leader-/followership.

3.3 Featured *Elements*

These *mechanics* (i.e., means of user-system interaction) can be implemented by various design components called *elements*. The term *elements* refers to specific tasks or objects that support *mechanics*, such as quests (for challenges), points (for rewards), and leaderboards (for achievement). Although Weiser *et al.* [21] identified seven types of *elements* as universal and context-free categories, this subsection explains the author's observations about how these *mechanics* are embodied as concrete *elements* for this piece in a context-specific way.

3.3.1 Feedback

The featured *elements* for *feedback* are as follows:

- obedient avatars;

- responsive collision sounds.

Responsive avatars and collision sounds give players immediate visible and audible feedback regarding their performance on the screen-score. This feature satisfies the need for self-determination during the performance and, thus, can afford the need for *autonomy*. Importantly, the predictable reaction of musicians' avatars to their sounds gives musicians the possibility of gaining better control of their avatars. When the musicians are immersed in gameplay, this possibility affords the desire of *competence*.

3.3.2 User education

The featured *elements* for *user education* are as follows:

- written instructions;

- rehearsals;

- rounds.

These *elements* complement the player's lack of knowledge regarding what is required in their performance and, thus, support the need for *competence*. If competence contributes to improving the performers' skills enough to win, user education can also help fulfill the need for *achievement*. The written instructions explain the concept of the piece and how to play it, thereby giving the performers a general understanding of the piece and helping them prepare for the performance. Rehearsals give performers the opportunity to learn how to play with the interactive computer system in action. They may conceive strategies to win the game, as well as contribute to musicality in the performance.

The rounds set up a heuristic process for performers during the performance. The rounds in this piece are designed similar to each other, with slight variations in rules and mechanisms. Thus, the rounds can be a learning opportunity where the performers develop their skills and interpretation further as the piece proceeds.

3.3.3 Challenges

The featured *element* for *challenges* is as follows:

- moving blobs.

The players' task is to clash the moving blobs with their avatars. Succeeding at this task can satisfy the performers' motivational need for *competence*. This task requires performers to be alert to the visual information and virtuosity for agile responses to the time-sensitive nature of the blobs' behavior. The task needs to be optimized to strike a balance between boredom and frustration among players. In this piece, blob behaviors are the essential factor that controls the difficulty level of the challenges. A concrete example of the challenges is presented in Section 4.

3.3.4 Rewards

The featured *element* for *rewards* is as follows:

- life points.

This mechanic rewards the player's success in clashing with blobs and, thus, fulfills the need for *competence*. Since the life point indicators explicitly show the audience the player's success, this mechanic can afford the need for *achievement*. The concept of life points places players in a competitive mode in which each player's health is compared to other players, and it may be threatened enough to 'die' in the game. The concept of life points is a crucial element that engenders both competition and cooperation. While comparisons to others may predispose players to view others as rivals, the threat of 'dying' in the game could make them view others as strategic temporal collaborators to avoid 'dying' due to difficult challenges. Thus, this element can also satisfy the need for survival, which is related to *leadership and followership*.

3.3.5 Competition

The featured *element* for *competition* is as follows:

- leaderboard.

The leaderboard announces the winner at the end of each round. After all four rounds have been played, the leaderboard announces the grand winner of the game to the audience. This element addresses the need for *relatedness* and *achievement*.

3.3.6 Cooperation

The featured *element* for *cooperation* is as follows:

- 'all dead' scenario.

The game can end with an 'all dead' scenario in which all players miss so many blobs that their life points are depleted. Although the players are primarily rivals, as only one player can win the game, this scenario gives players an incentive to cooperate and avoid missing blobs. Therefore, this element can satisfy the need for *leadership and followership* between ensemble members.

4. ANALYSIS OF A PERFORMANCE

This section describes an analysis of how the featured *element* of moving blobs affects the emergence of playfulness and musical results, referring to two video recordings of a performance by members (Vln I, Vln II, and Vla) of a professional string quartet, the Ligeti Quartet. The performance took place at Victoria Gallery Museum - Leggate Theatre at the University of Liverpool in UK on October 30, 2019. One of the recordings shows the performers and the screen-score on stage [3], while the second recording shows the same performance, but only the screen capture of the screen-score is shown. It is noted that the electronic sounds were not played due to a technical problem in this performance. Thus, the performance will not be analyzed from the perspective of how the electronic sounds influenced the performers.

The studies on motivational affordances [20][21] suggest that playful human-computer interaction emerges when *motivational needs* are met, and *competence*, one of the motivational needs, is afforded at the highest level when the level of difficulty is commensurate with the performer's skills. Hence, it is suggested that the target scenario of the game is a 'close battle,' which means players, including the computer system, win fairly throughout the game with close life-point scores. Notably, moving blobs are the only variable *element* the system uses to adjust the difficulty of the challenges to the performer's skill level. Other *elements* are not designed as variable parameters that players' actions can influence. Therefore, blob behaviors appear to be the most essential *element* that sways the emergence of playfulness in this piece.

There is, unfortunately, no way to compare the final life-point scores of the performance, as the precise life-point data were not recorded. However, the approximate life points on a video recording of the screen-score are available. Additionally, since close battles result in a longer round, it is possible to infer the closeness of the battle by comparing the duration of each section in the performance. This comparison reveals how close the battles were indirectly.

Therefore, the analysis focuses on the following:

- how many times each performer won throughout the performance;
- how long each round lasted in the performance.

The author's analysis illuminates the influence of blob behaviors on the time structure and choice of the following parameters:

- playing techniques - techniques used to play instruments, such as normal bowing, pizzicato, and tremolo;
- contour types - types of phrases, such as linear, leaps, and accelerando;
- dynamics - loudness of the performances, such as fortessimo and pianissimo;
- ensemble - a musical unit built by a performance of more than one player.

The following subsections describe the author's observations of how the blob behaviors affected the performers' choice in each round.

4.1 Round one

4.1.1 Game-end conditions

This round ends when one of the following game-end conditions is met:

- when one player reaches the maximum life points (i.e., 1,080 points);
- when the sequence data ends;
- when all players die.

4.1.2 Blob behavior

The blobs move from left to right on the screen in the order of musical notes in the preprogrammed sequence data of the Bach Invention. The speed of the blobs are constant, and it takes them approximately seven seconds to arrive at a point where the avatars can intercept them. The blobs are widely spread out vertically, and it is reasonably difficult to capture all the blobs.

[3] The video recording is available at https://vimeo.com/382978696.

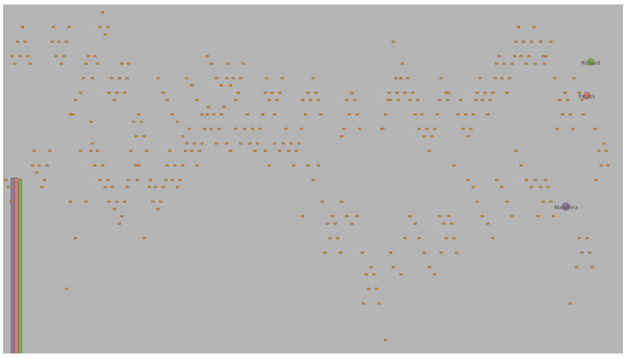


Figure 4. Widespread blob behavior in round one.

4.1.3 *Performers responses*

The blob sequence ended before any of the performers were able to win the first round by earning 1,080 points. Therefore, the second violinist, who earned the most life points during the round, was considered the winner. The end scores of all the players were low and close to each other. This result suggests that if the blob sequence had lasted longer, they all might have died. The round lasted for 55 seconds.

4.1.4 *Musical results*

Structurally, the round was separated into three sections: (1) before the moving blobs arrived in the vicinity of the avatars, (2) after the arrival of the moving blobs, but before all the blobs passed, and (3) after all the blobs had passed. Three playing techniques were observed: normal bowing with pizzicato during the first section, normal bowing with tremolo and pizzicato during the second section, and normal bowing during the third sections. Three contour types were also observed: a quick and short scaler phrase, long linear glissandi, and accelerating repeating notes. Four levels of dynamics were found. During the first and third section, the dynamics floated between piano and mezzo piano. During the second section, the dynamics rose to the range between mezzo forte and forte. Two ensemble units were found. The first ensemble unit was at the beginning of the first section, where the viola and violin II played a contrapuntal phrase together[4]. When the viola was performing an ascending phrase, violin II played an inversed descending phrase fairly concurrently. Another ensemble unit was in the third section, during which the viola and violin II played a cadence in collaboration[5].

4.2 Round two

4.2.1 *Game-end conditions*

There are two game-end conditions, as follows:

- when one player reaches the maximum life points (i.e., 1,080 points);
- when all players die.

[4] 0:19 in the video
[5] 1:01 in the video

Similar to round one, in round two, the players are challenged to earn more life points than the other players. However, the round continues until one player achieves the maximum number of life points (1,080 points). This is different from round one, which ends when the fixed blob sequence concludes, regardless of the player's life points.

4.2.2 *Blob behavior*

The moving blobs drew linear arrays during this round. These array forms were the result of the Markov chain algorithm optimizing blob progression. The linear arrays mean that the avatars engaged in less vertical movement to collide with blobs during the preceding round. As a result, the algorithm entered a higher weight in the repetition of the same pitches in the STM (see Section 2.4 for more details.)

Figure 5. Linear blob behaviors in round two.

4.2.3 *Performers' responses*

The round ended in an 'all dead' scenario occurred and no one won. The round lasted for 2 minutes and 24 seconds, which seems to be the longest round of the piece. The long duration of the second round suggests that, although all the players died, the players had a close battle with each other, as well as with the interactive computer system. This further indicates the blob behaviors were optimized as intended.

4.2.4 *Musical results*

Structurally, the round was divided into two sections: (1) before the moving blobs arrive in the vicinity of the avatars and (2) after the arrival of the moving blobs.

Four playing techniques were found: normal bowing and pizzicato during the first section and normal bowing, tremolo, pizzicato, and overpressure during the second section. Four contour types were also observed: a quick and short scaler phrase, long linear glissandi, repeating short notes, and accelerating and repeating notes. Four dynamic levels were found. The first drew the gradual dynamic change from piano to forte toward the second section. The second section showed a louder dynamic range between mezzo forte and forte. One ensemble unit was found. At the beginning of the first section, the viola and violin II played a heterophonic unit[6] in which the

[6] 1:10 in the video

viola first played a glissando, delineating a swell in pitch. Next, violin II played an ascending scale to the same pitch as the highest pitch that the viola had just played, drawing a shadow of the viola's part.

4.3 Round three

4.3.1 Game-end conditions

There is only one game-end condition in this round:

- when one player reaches the maximum life points.

4.3.2 Blob behavior

The third round is comprised of different blob behaviors from other rounds in two ways. First, the round introduces the concept of gravity for the blobs' motions. Unlike other rounds, the blobs gradually fall to the bottom of the screen, then bounce back to a slightly lower height than their original position. After repeating the bounce several times, the blobs gradually stay at the bottom of the screen. Second, the round restricts the space within which the blob can move. In the other rounds, the blobs move away and off the screen, but in the third round, they bounce back at the frame of the screen-score. As a result, once the blobs appear, they remain on the screen-score until clashed by the avatars. Therefore, this round guarantees that one performer wins by preventing an 'all dead' scenario from occurring, but this type of blob behavior may deflate the value of rewards (i.e., life points). The Markov chain algorithm is used to determine the initial vertical position of the blobs. However, the optimization seems irrelevant in this round as all the blobs remain within view of the players.

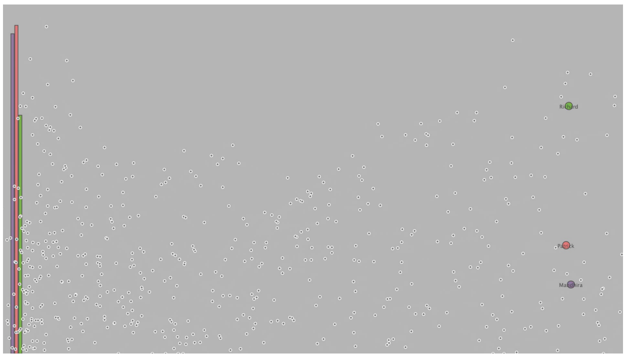


Figure 6. Bouncing blob behaviors in round three.

4.3.3 Performers' responses

Violin II won the third round, but it was a close battle, as violin I's final life points were extremely close to violin II's. The round lasted for 1 minute and 48 seconds.

4.3.4 Musical results

Structurally, the round was divided into three sections. The borders between these sections are, unlike the other rounds, not according to the blob's horizontal positions, such as one division when the blobs arrived in the avatars' territory and another when passing from the territory. The vertical position seemed to have more influence. The first section was mainly played by several quick, short glissandi. This suggests that the players were eager to reposition their avatars to the hight of their target bouncing blobs. The second section was mainly performed by pizzicato. This suggests that the players were motivated to intercept several blobs at every pluck, as one pluck of the pizzicato results in a quick pop of the avatar further to the left than normal bowing. If several blobs are on the path of the avatar, this playing technique helps the player clash with multiple blobs. The third section was played by normal bowing with which the players drew gradual increments of loudness to the end of the round. The change in playing techniques from pizzicato to normal bowing seems to have less impact on the competition and contributed to drawing a tutti at the end of the round.

Three playing techniques were found: normal bowing and pizzicato during the first section and normal bowing, tremolo, and pizzicato during the second section. Three contour types were observed: quick and short glissandi, repeating short notes played by pizzicato or normal bowing, and irregular rhythmic step-wise motions. Three dynamic levels were also found. The entire round showed a gradual increase in the dynamic level from mezzo forte to forttessimo. One potential ensemble unit [7] was found. All the instruments together intensified the loudness and created a tutti until the end of the round. However, it is unclear whether this musical effect was the performers' intentional choice or an incidental emergence influenced by the game system.

4.4 Round four

4.4.1 Game-end conditions

There are two game-end conditions, as follows:

- when one player reaches the maximum life points;
- when all the players die.

4.4.2 Blob behavior

The blob behaviors show an intermediate property between the widespread sequence in the first section and the linear arrays in the second round. This is because the Markov chain algorithm weights the likelihood of successfully clashed blobs during the first round and missed blobs during the second round. As a result, the difficulty level of the blob behavior is supposed to be higher than in, for example, round two. In addition, the speed of the moving blobs gradually increases over time. This also heightens the difficulty of intercepting them since the blobs pass by momentarily as their speed increases.

4.4.3 Performers' responses

The 'all dead' scenario occurred, and the round lasted for 56 seconds, which suggests that the challenge was too difficult.

[7] from ca.5:09 to 5:27 in the video

Figure 7. Blobs showing an intermediate property between linear arrays and a widespread constellation in round four.

4.4.4 Musical results

Structurally, the round was divided into two sections. During the first section, before the arrival of the blobs, the performers played a fugato. The second section was mainly performed by normal bowing. This choice was optimal for adjusting the height of their avatars to their target blobs.

Two playing techniques were found: normal bowing during the first section and normal bowing, tremolo, and pizzicato during the second section. Two contour types were observed: quick and short glissandi and repeating short notes played by pizzicato or normal bowing. Two dynamic levels were found: mezzo forte during the first section and forte during the second section.

One ensemble unit was observed, a fugato [8] at the beginning. Since strict phrasing rules, such as a fugato, do not normally contribute to adjusting an avatar's position to the ideal height for clashing with the blobs, performing the fugato seems to be for the purpose of enriching the music rather than gaining a strategic advantage in the game. It is worth mentioning that the phrasal material of the fugato was a short and quick step-wise motion used as a component in the ensemble unit performed at the beginning of the first round. This shared material draws a link to the historically recognized cyclic form.

4.5 Discussions

The analysis illuminates the influence of the featured blob behaviors on two different aspects of the music performance: competitiveness and the emergent musical structure. Competitiveness is related to performer's engagement with the game aspect of the music, while the emergent musical structure is related to performers' contributions to the form and components in the music.

For competitiveness, the result of the performance highlights the importance of the avatars' responsiveness to be equal across all the performers. The video recording shows that the second violin's avatar pops further left than other avatars with similar dynamics. This implies that the amplitude mapped to the horizontal position of the avatar was increased somewhere between a microphone and the screen-score. This flaw gave the second violinist an unintentional advantage to intercept the moving blobs earlier than other performers.

In addition, the result of the performance brings into question the effectiveness of the Markov chain algorithm in optimizing the difficulty level of the blob behaviors. As mentioned above, close battles are a crucial component of engendering playfulness in game-play. However, the analysis reveals a strong deviation in the results of the game. The second violin won twice, and the 'all dead' scenario occurred twice. Neither of the other two players won a round. It is remarkable that the rate of victories by the computer against human players was even. This result suggests that the challenge was optimal for the three performers playing against the computer as a group. Furthermore, the longer duration of the second round indicates that the Markov chain algorithm's optimization of the blob behaviors was effective to some extent. However, the challenge is not balanced between the three players. One prospective modification for this problem is to implement some kind of functionality to impose different levels of challenges to each of the three musicians, which means introducing the concept of handicapping in some way such as decreasing the input gain of an advanced player.

Regarding the emergent musical structure, it became clear that the moving blobs often resulted in dividing a round into at least two sections, before and after the blobs arrived in the avatars' vicinity. During the former, the musicians performed some complex units in collaboration, such as inversed contours (in round one) and a fugato (in round four). The moment before the arrival of the moving blobs seems to have been 'free time' for the musicians.

In contrast, after the moving blobs arrived in the avatars' vicinity, the performers chose to play more favorable materials with advantageous dynamics for winning the game rather than performing a complex musical unit as an ensemble. It tended to draw a clear border between the section before and after the moving blobs arrived in the avatars' vicinity. This observation suggests that there might be compensation between the risk of defeat and musical freedom. If this hypothesis is proven to be reasonable, competition might not be always the most effective gamification archetype to invoke playful interactions between human players and a computer.

It is remarkable that, in the third round, the border between the section one and section two was not divided according to the horizontal proximity between the moving blobs and the avatars. Instead, the border emerged from the choice of playing techniques that were advantageous for clashing the dense area of blobs near (and not moving away from) the bottom of the screen-score. Although all the players changed their choice of playing techniques, the timing of their changes was not as synchronous as the change from the first to the second section in other rounds. This lessened the clarity of the border and created a gradual transition. This example suggests that blob behavior influenced the musical structure.

Additionally, the transition from section two to section three in round three seems less relevant to winning the game. The author hypothesizes that this transition was the

[8] from 5:47 to 6:00 in the video

result of the performer's choice of musicality induced by the infinite accumulation of the blobs in the screen-score. The accumulation deflated the value of the blobs towards the end of the round, resulting in a situation where the avatars can clash with the blobs regardless of the choice of playing techniques. This further suggests that the excessive amount of the rewards delivered a certain degree of freedom to the performers. Interviews with the players would help explore this choice further.

Notably, the blob behaviors play a pivotal role in weighting the emphasis of the performance on either competitiveness or a musical structure. Different forms of blob groups influenced the choice of musical parameter settings. For example, a linear array often invoked competitiveness. Performers tended to await the arrival of the blob array at the same position in height. In this case, the performers tended to play louder to intercept the blobs further away before the other performers reached them. This situation turned the performance into a dynamic level competition. In contrast, the widespread blob sequence emphasized the musical structure. Performers tended to reposition their avatars vertically on the screen-score by playing the glissandi. The pitch changes sometimes entailed interweaving the contrapuntal lines (e.g., at 0:19 in the video). Thus, the performance was more focused on a musical structure.

5. EVALUATION AND FUTURE WORK

5.1 Evaluation

The piece can be evaluated as partially successful in fulfilling the artistic objective, which was to integrate historically inherited compositional devices and enthusiasm of game-play into a piece of music. One approach was to use the midi sequence data of the Bach Invention as the initial contents on the screen-score in the first round. This approach worked fairly effectively, since the approach resulted in the contrapuntal interweaving between voices during the performance in the first round. Another approach was to use the Markov chain algorithm in order to involve enthusiasm of the game-play into the musical structure while keeping the elements of the compositional devices in the Bach Invention. The author found that the second approach requires some modifications to fulfill the artistic goal. Although competitiveness in the piece invoked enthusiasm of the game-play, the aforementioned conventional compositional devices (e.g., a fugato, a cyclic form) appeared only during the sections where the blobs symbolizing pitch notes in the Markov-generated midi sequences are away from the performers. The absence of the compositional devices questions the effectiveness of the Markov chain algorithm in the game context to create a synergy of the performers' game-play and the compositional devices.

5.2 Future work

In order for competitiveness to contribute to the enriching musical structure, the author envisions two further explorations: more drastic changes of the blob behaviours and the use of various mapping combinations between acoustic parameters of the instrumental sounds and behavioural parameters of the avatars.

For the former, one idea is to implement vertical blob motions from top to bottom in order to induce more contrasts in dynamics in music. The author expects that the vertical motions will induce a competition on dynamic level control instead of the competition on their loudness level observed in round one, two and four, since the performers need to adjust the avatar's horizontal position onto the course of the target blobs by a precise loudness control. Another idea is to implement horizontal blob motions from right to left for the emergence of a quiet music, as the performers would compete to adjust the avatars' vertical positions while playing as silent as possible to stay further ahead to the right on the screen.

For the latter, one idea is to map spectral centroid of the instrumental sounds to the avatars' vertical position. This mapping could invoke a competition on brightness in timbre, which may induce the use of various alternative playing techniques for the avatars' control, thereby contributing to enriching the musical textures. These modification ideas call for a further exploration on the influence of behavioural patterns of the blobs on various musical parameters for achieving the contribution of competitiveness to the musical structure.

6. CONCLUSIONS

Super Colliders applied the concept of gamification to a screen-score to invoke playfulness in performer-computer interactions. The paper introduced conceptual and technological aspects of the game design (i.e., the goals and rules, the behaviors of performing agents, and the design and implementation of the gamified screen-score). Next, elements relevant to the emergence of playfulness in light of the taxonomy of motivational affordances were identified. Finally, a performance was analyzed to investigate how the performers play a close battle with the game system and what musical results the performer-computer interactions entailed. Comparisons with the motivational affordance study illuminated the significance of the motivational need for *competence* in this piece. However, the analysis found the interactive computer system overpowered human players. This result indicates the need for modifying the second-order Markov chain to optimize challenges (i.e., blob behaviors) more effectively, which is essential to evoke playfulness. The author believes that the gamified screen-score has the potential to support the future adjustment of game mechanics, as the gamified screen-score is not just an intermediary of musical symbols between the Markov chain algorithm and human performers. Instead, it is a performance ecosystem in which multiple active agents continuously interact with each other through various modalities regardless of whether they are humans or computers.

Acknowledgments

The author would like to thank the members of Ligeti

Quartet for their commitment to the performance, McGill University and CIRMMT for production support, and the FRQSC for financial support.

7. REFERENCES

[1] B. Terrill. (2008) "my coverage of lobby of the social gaming summit". Accessed: 2019-12-15. [Online]. Available: http://www.bretterrill.com/search/label/gameification

[2] S. Deterding, R. Khaled, L. Nacke, and D. Dixon, "Gamification: Toward a definition," in *CHI 2011 Gamification Workshop Proceedings*, Vancouver, BC, Canada, 2011.

[3] T. Winkler, *Composing Interactive Music: Techniques and Ideas Using Max*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[4] G. E. Garnett, "The aesthetics of interactive computer music," *Comput. Music Journal*, vol. 25, no. 1, pp. 21–33, 2001.

[5] J. Drummond, "Understanding interactive systems," *Organised Sound*, vol. 14, no. 2, pp. 124–133, 2009.

[6] G. E. Lewis, "Too many notes: Computers, complexity and culture in "voyager"," *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

[7] J. Chadabe, "Interactive composing: An overview," *Computer Music Journal*, vol. 8, no. 1, pp. 22–27, 1984.

[8] A. Di Scipio, "'sound is the interface': From interactive to ecosystemic signal processing," *Organized Sound*, vol. 8, no. 3, pp. 269–277, 2003.

[9] G. E. Winkler, "The realtime-score. a missing-link in computer-music performance," in *Proceedings of the 1st Sound and Music Computing Conference (SMC2004)*, Paris, France, 2004.

[10] D. Kim-Boyle, "Real-time Score Generation for Extensible Open Forms," *Contemporary Music Review*, vol. 29, no. 1, pp. 3–15, 2010.

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

[12] P. Auslander, *Liveness: Performance in a Mediatized Culture*, 2nd ed. London, England: Routledge, 2008.

[13] J. Brackett, "Some notes on john zorn's cobra," *American Music*, vol. 28, no. 1, pp. 44–75, 2010.

[14] D. van der Schyff, "The free improvisation game: Performing john zorn's cobra," *Journal of Research in Music Performance*, vol. 5, no. 1, pp. 1–11, 2013.

[15] I. Xenakis, *Formalized Music Thought and Mathematics in Composition*, 1st ed. Bloomington, IN, USA: Indiana University Press, 1971.

[16] K. Giannoutakis. (2015) "contraction point". Accessed: 2019-12-15. [Online]. Available: http://www.kosmasgiannoutakis.eu/contraction-point/

[17] C. Ressi. (2017) "game over". Accessed: 2019-12-15. [Online]. Available: http://christofressi.com/projects/game-over/

[18] "Guitar hero," 2020, accessed: 2019-12-15. [Online]. Available: https://www.guitarhero.com/

[19] C. Hope and L. Vickery, "Screen scores: New media music manuscripts," in *Proceedings of the International Computer Music Conference 2011*, Huddersfield, UK, 2011.

[20] P. Zhang, "Technical opinion: Motivational affordances: Reasons for ict design and use," *Commun. ACM*, vol. 51, no. 11, pp. 145–147, 2008.

[21] P. Weiser, D. Bucher, F. Cellina, and V. D. Luca, "A taxonomy of motivational affordances for meaningful gamified and persuasive technologies," in *EnviroInfo and ICT for Sustainability 2015*, 2015.

CONDUCTING ANIMATED NOTATION: IS IT NECESSARY?

Aaron Wyatt
Monash University
aaron.wyatt@monash.edu

Cat Hope
Monash University
cat.hope@monash.edu

ABSTRACT

At the 2019 Perth Festival, Western Australia's largest and most nationally significant arts event, a new animated notation opera by Cat Hope was premiered. This sixty minute staged work for thirty piece orchestra, thirty voice community choir and four vocal soloists ran over six nights and was led by musical director Aaron Wyatt. The score was delivered over 26 networked iPads in the orchestra within the Decibel ScorePlayer application [1], controlled by the musical director, and both the choir and vocal soloists memorised the graphic score. In addition to directing the preparations for the performance, Wyatt conducted the orchestra and singers from the podium each night.

This paper discusses the role of the conductor in this performance, and examines the role of the conductor in works performed from animated notations more broadly. A questionnaire was sent to the orchestral members who participated in the performances of 'Speechless' asking a series of questions about the impact and role of the conductor in the work, and their responses inform the body of this paper. Overall, the responses indicate that although the score presentation for the work on the networked iPads was very exacting, the role of the conductor was essential for extracting musicianship and nuance in each performance.

1. THE SCORE FOR SPEECHLESS

The score for Speechless is divided into five movements: Overture, Act I, Act II, Interlude and Act III. The scores for each of these movements are contained within a single score file and in performance, a python script connected to the ScorePlayer automatically loads the next movement once the previous one is finished. They are started manually by the conductor so that there is more control over the amount of time before each movement. Because of the networked nature of the iPads, the only interaction that the musicians need to have with their score is to choose their specific part. Additionally, the script used to partially automate the playback of the score has a curses [2] interface (see figure 1) that shows information about the current playback state, allows for control of the score and most importantly shows the list of the iPads that should be connected to the network. As can be seen in the figure, this device list is in the left hand panel, and the device names appear in white here because their status is not polled during playback in order to limit any non-essential network traffic. Had the score been stopped, connected devices would be shown in green, disconnected ones in red, and the server iPad in blue, allowing the user to very quickly ascertain the status of the network. The top right panel of the interface shows any OSC messages that have been sent by the server, and the lower panel shows the current playback status, including a progress bar.

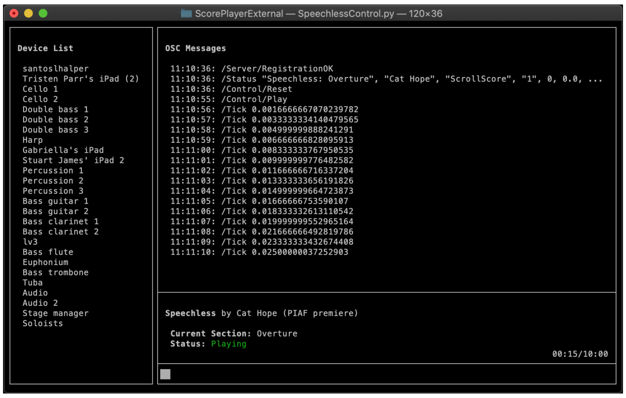


Figure 1. The interface to the Speechless controller script, displayed here during playback of the overture.

There are fourteen parts in the score. The mixed community choir is divided into four mixed groups, each made up of a combination of age, ability and gender, and each group is connected with one of the vocal soloists, as there are sections when the soloist will join their connected choir group. A bass orchestra – that is, an orchestra of musicians playing below Middle C in pitch – is also featured [3] and is subdivided into like instrument groups. Each group is designated by different colours; red for low brass (tuba, euphonium, two bass trombones), green for low winds (4 bass clarinets, contrabass clarinet, contrabass bassoon, two bass flutes), blue for low strings, (four cellos, six double basses) purple for percussion (the percussionists including one rock drummer playing a bass drum kit, two percussionists playing bass drums, tam tam, cymbals, a.m. radios and other instruments) yellow for electronics (including Theremin), and orange for piano, two bass guitars and harp. There is also a colour for each of the vocal soloists, and a related shade of that colour for each of the four choir groups [4].

The score in the Decibel ScorePlayer can be seen with the complete orchestration, as required by the conductor, or with one part highlighted, obtained by presenting the remaining parts at a low opacity. The score opens at the full score, and the highlighted part can be revealed by the individual iPad user dragging the finger in an upward motion, scrolling through until their part is revealed [5]. The score progresses at a pace set in the ScorePlayer, and each musician reads the score at the point of the playhead [6].

2. AIMS FOR AN OPERA CONDUCTOR

In spite of the animated, graphical nature of the score, many of the considerations for conducting the opera were the same as they would have been for a more conventional production, and the conductor took on a largely familiar role. They were responsible for rehearsing the soloists, chorus and orchestra, first separately and then together. And one of their main roles during the performances was to ensure a high level of coordination between the singers on stage and the orchestra, alongside managing balance. There were a number of factors that complicated this, some of which were unique to the type of score, and others more commonplace.

Figure 2. The orchestra for speechless, situated on the stage behind the soloists and choir. The scrim behind the conductor is the only separation.

The staging decision to place the orchestra behind the singers is one that is becoming increasingly common in newer productions, and it makes a great deal of sense when performing in a venue not originally designed for opera or musical theatre: the lack of an orchestra pit generally necessitates a more creative approach to be taken with spacing and the positioning of the ensemble. (See figure 2. Additional video documentation can be found online [7].) The obvious downside to this is that direct eye contact with soloists or members of the chorus becomes impossible. Cues have to instead be given into a camera, losing any sense of directionality. It becomes incredibly important that the singers know the structure of the work well and for there to be at least some visible differentiation between cues from the conductor so that the soloists and chorus know exactly which ones are meant for them. It also makes it harder for the conductor to react to what is happening on stage, although this can be alleviated slightly using technical solutions, such as cue lights to signify when to start movements that are dependent on staging. A benefit of the lighting desk and the ScorePlayer both supporting OSC [8] meant that the process of sending cues tied to lighting states could be largely automated. (See figure 3)

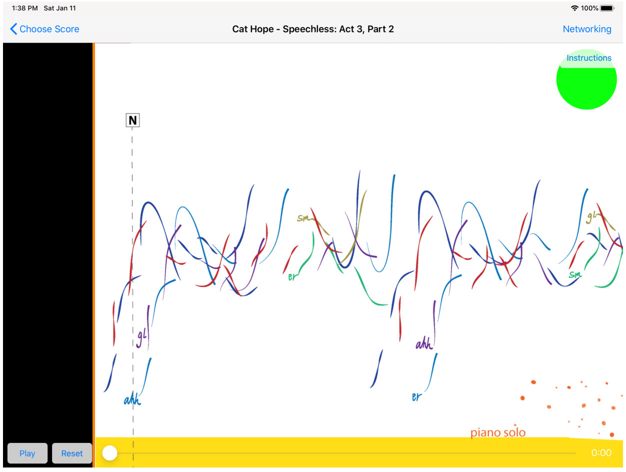


Figure 3. The start of Act 3 Part 2 showing a cue light delivered directly to the conductors score, triggered by a change in lighting state.

The main challenge faced as a conductor in directing the work is the lack of traditional pulse or rhythm. This also proved to make memorisation for the singers a slightly daunting task, and so a large part of the role became guiding the soloists through their phrases, and helping the choir with their entries. While the music often achieves a timeless quality through the use of long drones and glissandi, the timing of these is generally deceptively precise, and the continuous, scrolling nature of the score makes it hard to offer any concession to the singers. So with no need to conduct any sort of beat pattern, the conductor is freed (using largely vertical gestures) to outline the vocal contours for the soloists as an aid to memory and as a way to keep the timing exact. The other hand is left free for cuing, or can even be used to show a second line in the case of a vocal duet.

It is important to note as well that while the opera eschews traditional notions of rhythm and tonality, it is still highly structured, and has many features that you would expect to find in a more traditional work. There are aria and recitative-like passages for the soloists. There are recurring motifs and phrases that are repeated, sometimes in inversion or retrograde. There are places where the singers echo the contours of the orchestra and vice versa. Being aware of these structural elements and knowing how they fit together is vital in interpreting the work, and in this way the role of the conductor remains unchanged [9].

3. CONDUCTING PREPARATION

Four members of Decibel new music ensemble were in the orchestra. Decibel is a six piece new music group that focuses on the reading and creation of animated notations, as well as being the research team that developed the Decibel ScorePlayer. Wyatt and Hope are also members but did not play in the orchestra. The four members were able to lead sectional rehearsals where musicians were coached in the required approach. The Decibel musicians, percussionist Louise Devenish (leading percussion), clarinettist Lindsay Vickery (leading winds/brass), cellist Tristen Parr (leading strings) and electronic musician and pianist Stuart James leading all other instruments. Cat Hope coached the electronic bass players. This allowed the first full rehearsal to be carried out very efficiently with minimal need to stop and start to clarify the notation.

The conductor worked with the choir and the soloists independently, grouping material into different sections to aid in memorising the material. There vocalists were all able to see the conductor in monitors visible form the stage, meaning cues were clear for the start, end and dynamic range of materials. Each group in the choir had a leader that would be the most familiar with the material so that the group could refer to them as a memory aid. The nature of community choirs was such that it was not guaranteed that the same singers would be at every night of the season, making this role particularly valuable. This person would also be the one actively keeping an eye on the monitor, to avoid a whole choir 'looking' at the monitor above the stage. The rest of the section could then follow them aurally.

Unlike the soloists, the parts for the choir were often much more indeterminate, using various simple vocal effects to create a rich soundscape. As a result, their parts could be condensed down to a few pages of material to study, and over the course of a few rehearsals they were able to go from following the projected score to singing from memory with the aid of group leaders and conductor cues. By the time of the sitzprobe, they were working without the projection. The conductor developed an innovative technique that showed what sounds were required using a combination of the shape of their mouth, and hand gestures that matched either their percussive or flowing nature. For example, in their first entry in Act 1 (see figure 4), the choir are making 'tk' sounds, starting sparsely and then gradually building in density. These are shown with short, sharp, flicked hand gestures that get busier as the density increases. As much as possible, these gestures aimed to closely capture the feel of the sounds to draw out an instinctive musical response from the performers, much as would be the case with traditional expressive conducting gestures, or the sculptural gestures of sound painting [10]. (Unlike soundpainting, the precomposed nature of the work removes the necessity for more formally codified functional gestures.)

While it would seem that having everyone follow the score exactly as it hits the playhead should be enough to guarantee synchronised entries, past experience in the Decibel new music ensemble has shown that traditional ensemble skills are still important: there are often slightly different interpretations of the exact moment that an event hits the playhead, and having players moving together and either leading or following as they would in any other traditional chamber work helps to unify the ensemble. It stood to reason that the same would apply in this context, and that having a conductor would help to guarantee that section entries and cut offs were tight. Through the rehearsal process, we found this to be the case, particularly for very exposed entries. When players looked up from the score and followed un upbeat given by the conductor, some particularly problematic entries (such as the wind entry in figure 5) were much cleaner.

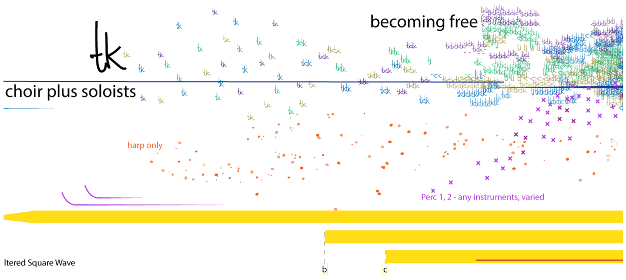


Figure 4. The first entry of the choir in Act 1, represented in the upper half of the image by the small 'tk' symbols.

Figure 5. The blue line is a solo cello line that starts the overture. The green line is a very exposed, pianissimo wind section entry that was much cleaner when players watched the conductor instead of the score.

To gain the perspective of the orchestral musicians and to see if they agreed with our assessment of the situation, we invited a number to take part in a survey related to their experience in the production. They were asked a series of yes or no questions and were then given the opportunity to elaborate where desired. Seven of the thirty musicians took part, and six of them responded to every yes/no question. (The responses to these can be seen blow in table 1.) Six out of the seven agreed that the conducting had an affect on the accuracy. And while opinions were split on whether the opera could have been performed without a conductor, the six who responded to the question all believed that the opera could not have been rehearsed without a conductor.

When asked to elaborate on how the experience might have been different without a conductor, a number of common themes came up in the responses. While many of the respondents picked up on the issue of the cohesion of entries and cut offs, some also went further, suggesting that unity of articulation, interpretation and approach were all aided by the presence of a conductor. Here are some of the ways in which respondents felt the production may have been different if unconducted:

> Much less accurate, no feel of how the piece fits together

> There would not have been a strong link between the vocalists and the orchestra. There also needs to be a central point fo [sic] balance and articulations.

> Sectional entries and exits would have been less uniform and controlled.

> Less cohesion among players More hesitant [sic]

> It would have been a messier performance, in terms of variations of interpretation of notation, in terms of ensemble togetherness, in terms of unity of approach.

> I feel as if the musicians would have kept their heads down a lot more and stayed unaware of the overall context of their parts. With time though this could be a learnt skill but very difficult for orchestral musician without someone to follow whether it is a conductor or principle player.

The last response raises an important point for consideration. Given that the timing of events in the score is delivered through a constant stream of animation in a manner that doesn't neatly align to any quantised, internal pulse for the musicians, it is very easy for them to become highly focussed on the iPad directly in front of them. Having a conductor as an alternative point of focus helps them to look up and engage more with their surroundings.

4. CONDUCTING THE PERFORMANCE

In conducting the performance, the main focus was on maintaining the level of ensemble and musicality that had been developed throughout the rehearsal process, as would be the case for any production. The soloists generally took priority, especially when they had either duets, or long aria-like phrases, and it was also important to encourage and reassure the choir as much as possible. In spite of this attention given to the stage, it was still possible to cue many of the orchestral entries and cut offs, especially ones that were exposed or that involved multiple sections of the orchestra.

Additionally, gesturing the shaping of phrases and showing the dynamics helped to heighten the level of expression produced by the ensemble. While there are a number of noisy, climactic moments in the opera, there are also quite a few long passages that need to be kept as soft and still as possible. The danger in these passages is that the volume gradually starts to creep up after a period of time, and actively gesturing for the dynamics to remain soft helps to prevent this, particularly across repeat performances where it is easy for complacency to settle in.

Question	Yes	No
Does the conductor offer something important beyond the score?	6	1
Does the conducting have an effect on accuracy?	6	1
Do you look at the conductor as much as you would when performing from a conventional score?	5	2
Do you feel the work could have been performed without a conductor?	3	3
Could it have been rehearsed without a conductor?	0	6
Were the conducting gestures unique for this type of score?	3	3
Did the conductor provide something different or do something different to what they usually would?	4	2
Did you find the Decibel ScorePlayer easy to use?	6	0
Was the score easy to understand?	6	0
Did the conductor elaborate on the score in any way?	5	1

Table 1. The results of the musician survey.

While this outlines the main aims that Wyatt had while conducting the performances, the question remains as to how effectively these aims were met. And if they were, did they enhance the performance to an extent that justifies the presence of a conductor? To answer these questions, we need to turn once again to the results of the musicians' survey.

4.1 Ensemble and Expression

Was the conductor's focus on maintaining ensemble and guiding the phrasing and musicality of the work effectively conveyed to the players? As a follow up to their thoughts on whether the gestures used were unique and whether the conductor provided something different on this occasion, the survey respondents were asked where the conductor's emphasis shifted given the lack of beats in the work. Their answers showed that they perceived the shift exactly as was hoped.

> The conductor helped to indicate changes in section, timbre, articulation, dynamics and so on.

> Mainly towards dynamic changes and entries. He was excellent in this regard!!!!!!!!

> dynamic, overall mood

> Conducting isn't really about beats for me anyway....the emphasis was the same in that the

> conductor shows start and end points of phrases, indication of balance/dynamics/energy etc. Until I read this question, I didn't even notice that the conducting wasn't about standard beat patterns.

This last response helps to paint a picture of why there was such an even split on the previous two yes/no questions. While the gestures for the singers were somewhat unique, many of the expressive gestures directed at the orchestra were ones that the players would have experienced in other contexts. It is partly this familiarity that helped to make them as effective as they were. And while there were more respondents who thought that the conductor did something different to what they would usually do, two out of the three who clarified their answer identified the lack of beat as the primary or sole difference, lending credence to the idea that there were more similarities than differences.

4.2 The Score and Beyond

When asked if the score was easy to understand and whether the ScorePlayer itself was easy to use, all six of the respondents answered yes. In elaborating on this, a couple of the respondents noted that they were already familiar with the software and with the style of music, while others had these sorts of observations about the ScorePlayer:

> Easy to understand and very clear

> Totally suited to this kind of music. Very clearly notated.

And similar comments about the score itself:

> Graphics were easy to read and follow

> It takes some getting used to reading a score of this type but is quite intuitive to read once more familiar.

It is worth noting as well that one of the respondents who was already familiar with the ScorePlayer had encountered technical issues when using it previously, but found that "everything worked well for these rehearsals and performances." This may be the result of stability improvements that have happened throughout the ongoing development of the app, the highly controlled networking environment that was used for the production, or some combination of the two.

The general clarity of the score, and the perceived ease of use of the player help to explain why half of those who responded thought that the show could potentially go ahead without a conductor once we made it through the rehearsal process. (Remember though that this survey was only of the members of the orchestra: the singers, not having the score in front of them, may have had other ideas about how effective that might have been.) But did the conductor help the musicians to get more from the score than was suggested by just the scrolling images themselves? Some of the previous responses have touched on this to an extent, but when asked to expound more directly on whether the conductor elaborated on the score in any way, respondents generally agreed that the conductor provided additional clarity and further explanation of articulations and markings, and of the general structure of the work. Here are a couple of representative examples:

> The focus was on attack and articulations and the visual representation of these parameters. This lead to a very cohesive orchestral sound.

> Helped to make sense of the markings and had the overall context of the piece in his head which he was able to convey to us. I feel most orchestral musicians will listen to recordings to gauge the context of their parts, so it was helpful for the conductor to go through which parts were playing which "roles".

Tying this all together, six out of the seven respondents agreed that the conductor offered something important beyond the score. Their elaborations on this question provided some of the most compelling arguments as to the utility of engaging a conductor in such a production:

> Aaron provided a greater depth of musicality to the score. Like any orchestral notation the information is on the page however he ensured that the entire orchestra played as one entity. The conduction also created better accuracy of entries within each individual section. Aaron also acted as the conduit between the orchestra and singers / soloists and visa versa.

> Emotion, further guidance, feedback. I feel a strong emotional connection to the conductor so will react better to their facial expressions and gestures than purely to a screen.

> Indications of ensemble balance, encouragement of personal interpretations of notation, clarity of start and end points of whole-ensemble sound where there is a risk of dozens of interpretations of 'end', a sense of unity. (Maybe)

Having a conductor adds a human touch that would otherwise be lacking if the musicians were focussed solely on the screens in front of them. Feedback from the conductor helps them to gauge balance, to feel confident and unified in their entries, and helps them to feel that they have an understanding of how their part fits into the work. All of these benefits suggest that there is still a place for a conductor in large scale, animated notation works.

5. CONCLUSIONS

While animated graphic scrolling scores of the type used in Speechless can be very prescriptive, particularly in how they convey the timing of events, our experience with the production and feedback from the orchestral players show that there are still advantages to be gained from engaging a conductor in the interpretation of larger scale works that are based around them. Just as a conductor can bring out nuance, encourage phrasing, and foster unity in a more traditional score, so too can they bring these abilities to the direction of a scrolling score, both in rehearsal and performance. They add a human element that complements and arguably enhances this new technological mode of delivery, offering familiarity in this more experimental setting to those who come from a traditional orchestral background.

Acknowledgments

The authors wish to thank the other members of Decibel new music ensemble; Lindsay Vickery (Clarinet), Stuart James (electronics), Louise Devenish (percussion) and Tristen Parr (cello). Tristen Parr also made a significant contribution to the production as orchestral leader and producer. Also, thanks to Tura New Music, who produced the premiere and development, and co-commissioners the Festival of Perth, Australia Council and WA Department of Local Government, Sport and Cultural Industries.

6. REFERENCES

[1] C. A. Hope, A. K. Wyatt, and L. Vickery, "The decibel scoreplayer – a digital tool for reading graphic notation," in *Proceedings of TENOR, First International Conference on Technologies for Music Notation and Representation, 2015*, Institut de Recherche en Musicologie, 2015, p. 59.

[2] A. M. Kuchling and E. S. Raymond, "Curses programming with python," 2020. [Online]. Available: https://docs.python.org/3/howto/curses.html

[3] "Australian bass orchestra manifesto," accessed 12 January 2020. [Online]. Available: https://www.cathope.com/noise-performance

[4] C. Hope, "Speechless," 2 2019. [Online]. Available: https://doi.org/10.26180/5c591dc038001

[5] C. Hope, A. Wyatt, and D. Thorpe, "Scoring an animated notation opera – the decibel scoreplayer and the role of the digital copyist in 'speechless'," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2018, p. 200.

[6] C. A. Hope, L. Vickery, and A. K. Wyatt, "The decibel scoreplayer – new developments and improved functionality"," in *Proceedings of the International Computer Music Conference*, 2015, p. 315.

[7] C. Hope, "Speechless, an opera by Cat Hope," Feb. 2020. [Online]. Available: https://vimeo.com/389950595/e942e00897

[8] M. Wright, "The open sound control 1.0 specification," 2002. [Online]. Available: http://opensoundcontrol.org/spec-1_0

[9] R. Holden, *The technique of conducting*, ser. Cambridge Companions to Music. Cambridge University Press, 2003, p. 13–14.

[10] H. Minors, "Music and movement in dialogue: Exploring gesture in soundpainting," *Les Cahiers de la Société québécoise de recherche en musique*, vol. 13, no. 1-2, pp. 87–96, 2012.

OPTICAL MUSIC RECOGNITION: STATE OF THE ART AND MAJOR CHALLENGES

Elona Shatri
Queen Mary University of London
e.shatri@qmul.ac.uk

György Fazekas
Queen Mary University of London
g.fazekas@qmul.ac.uk

ABSTRACT

Optical Music Recognition (OMR) is concerned with transcribing sheet music into a machine-readable format. The transcribed copy should allow musicians to compose, play and edit music by taking a picture of a music sheet. Complete transcription of sheet music would also enable more efficient archival. OMR facilitates examining sheet music statistically or searching for patterns of notations, thus helping use cases in digital musicology too. Recently, there has been a shift in OMR from using conventional computer vision techniques towards a deep learning approach. In this paper, we review relevant works in OMR, including fundamental methods and significant outcomes, and highlight different stages of the OMR pipeline. These stages often lack standard input and output representation and standardised evaluation. Therefore, comparing different approaches and evaluating the impact of different processing methods can become rather complex. This paper provides recommendations for future work, addressing some of the highlighted issues and represents a position in furthering this important field of research.

1. INTRODUCTION

Music is often described as structured notes in time. Musical notations are systems that visually communicate this definition of music. The earliest known scores date back to 1250-1200 BC in Babylonia [1]. Since then, many notation systems have emerged in different eras and different locations. Common Western Music Notation (CWMN) has become one of the most frequently used systems. This notation has evolved from the mensural music notation used before the seventeenth century. Current work in Optical Music Recognition focuses on the CWMN; nonetheless, studies are also carried out for old notations, including mensural, as shown in Table 1.

Classifying music based on its difficulty is highly subjective. Nevertheless, Byrd and Simonsen [30] in their attempt to have a standardised test-bed for OMR, name four categories based on the complexity of the score [30] (see Figure 1):

1. Monophonic: music in one staff with one note at a time;
2. Polyphonic: music in one staff with multiple voices;
3. Homophonic: music in multiple staves, but each staff is monophonic;
4. Pianoform: music in multiple staffs and multiple voices with significant structural interactions.

OMR has been researched for the last five decades; nonetheless, a unified definition of the problem is yet to emerge. However, Calvo-Zaragoza [31] offers the following definition of OMR.

Definition 1.1 *"Optical Music Recognition is a field of research that investigates how to read music notation in documents computationally."*

The importance of OMR is evident both in the abundance of sheet music in archives and libraries, much of this is yet to be digitised, and in the common practice of musicians. Paper remains the first medium authors use to write music. By taking a picture of a score, OMR would enable us to later modify, play, add missing voices and share music using ubiquitous digital technologies. It also enables search capabilities, which are especially crucial for long pieces or large catalogues in music information retrieval and digital musicology. Other advantages of OMR include conversions to different sheet music formats (e. g. Braille music notation) and the ability to archive musical heritage [32].

Fundamentally, OMR's goal is to interpret musical symbols from images of sheet music. The output would be a transcribed version of the sheet, which is also machine-readable, i.e., musical symbols can be interpreted and manipulated computationally. The usual output formats are MusicXML and MIDI. These formats will include musical attributes and information such as pitches, duration, dynamics and notes.

OMR has previously been referred to as Optical Character Recognition (OCR) for music. However, music scores carry information in a two-dimensional structure, ordered sequence of musical symbols and the set of most often co-occurring musical events. In contrast, OCR deals with sequences of characters and words that are one-dimensional.

The field of OMR saw its beginnings in the late 1960s with pioneers like Pruslin [33], Prerau [34] and continued with Kassler [35] reviewing the dissertations mentioned above. Subsequently, other researchers including Fujinaga [2], Carter [36], Ng [4], Bainbridge and Bell [10] joined the field.

More recently, the success deep learning has had in improving text and speech recognition has triggered a

References	CWMN	Old	Typeset	Handwritten
Fujinaga [2], Coüasnon et al. [3], Ng and Boyle [4], Chen et al. [5], Vidal [6], Bui et al. [7], Huang et al. [8]	✓		✓	
Ng et al. [9], Bainbridge and Bell [10], Gocke [11], Rebelo et al. [12], Fornés et al.[13], Pinto et al. [14], Hajič and Pecina [15], Roy et al. [16], Pacha et al. [17], Tuggener et al. [18], Baró et al. [19, 20]	✓			✓
Calvo-Zaragoza and Rizo [21], Wen et al. [22], Pacha and Eidenberger [23, 24], Calvo-Zaragoza et al. [25]	✓		✓	✓
Calvo-Zaragoza et al. [26, 27], Huang et al. [28], Tardón et al. [29]		✓		✓

Table 1: Studies conducted in CWMN (Common Western Music Notation), and old notations (used before the CWMN, mostly mensural notations

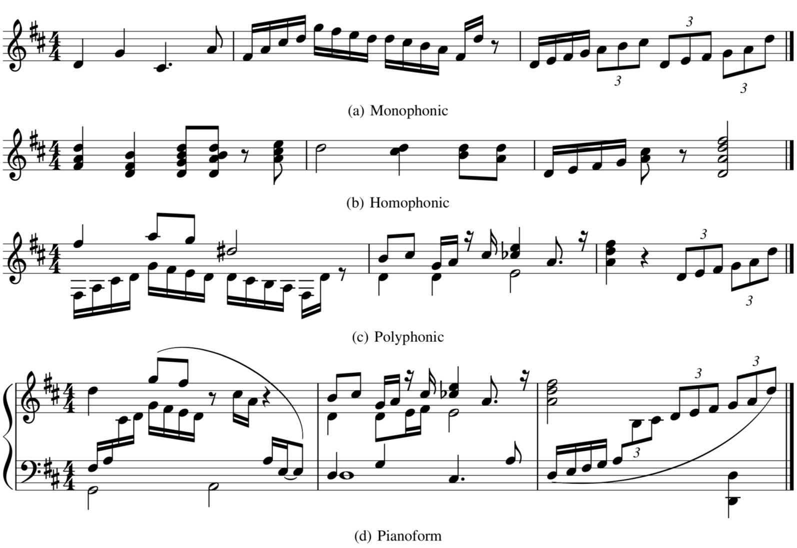


Figure 1: A visual representation of the four categories of music notations [30, 31]

paradigm shift in OMR as well. One of the most comprehensive reviews on OMR was written in 2012 by Rebelo et al. [37]. However, at that time, the field had not yet seen the emergence of machine learning approaches. This position paper aims to update on these approaches.

State of the art works in OMR perform well with digitally written monophonic music, but there is plenty of room for improvement when it comes to reading handwritten music and complex pianoform scores [21, 22, 23]. The difficulty thus increases with the complexity of the music notation.

2. OMR PIPELINE

The standard OMR pipeline given by Rebelo et al. [37] is depicted in Figure 2:

1. Image preprocessing;
2. Music symbol recognition;
3. Musical information reconstruction;
4. Construction of a musical notation model.

In the first stage, images of sheet music are subject to techniques such as noise removal, binarisation, de-skewing and blurring in order to make the rest of the OMR processes more robust. Subsequently, reference lengths, such as staff lines thickness and distances between them are calculated. Typically, the next stage is musical symbol recognition. This stage consists of staff line processing and musical symbol processing and ends with classification. Primitives of musical symbols will be used in the third stage in order to reconstruct semantic meaning. Finally, all retrieved information should be embedded in an appropriate output file. A summary of these stages and the particular image processing and machine learning techniques employed in each stage are summarised in Table 2.

3. IMAGE PREPROCESSING

Image preprocessing is a fundamental step in many computer vision tasks. The primary outcome of this stage is an adjusted image that is easier to manipulate. Most common image manipulations include enhancement, de-skewing, blurring, noise removal and binarisation [2, 4, 39, 13, 29, 14, 26, 28, 22]. Image enhancement can include filters and adjusting the contrast or brightness for optimal object detection. De-skewing eliminates skewness and helps

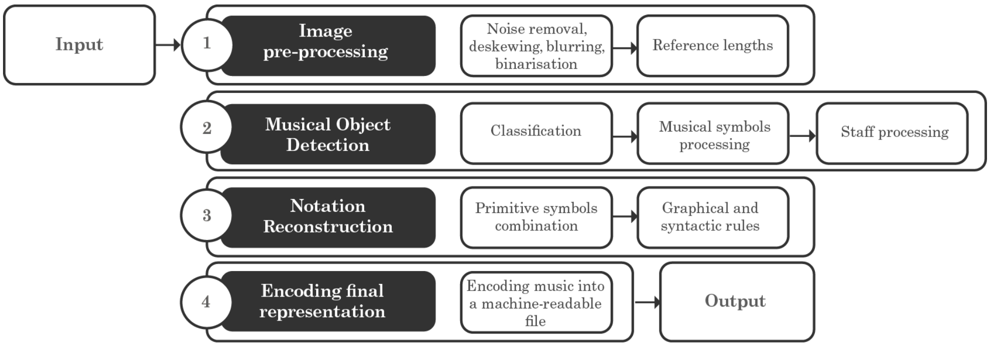


Figure 2: Conventional OMR pipeline

Stage	Related Work
Image preprocessing	Fujinga [2, 38], Ng and Boyle [4], Fornés et al [39, 13], Tardón et al. [29], Pinto et al. [14], Calvo-Zaragoza et al. [26], Huang et al. [28], Wen et al. [22], Ridler et al. [40], Gocke [11], Ballard [41], Bainbridge and Bell [42], Cardoso et al. [43], Dalitz et al. [44]
Symbol Recognition	Mahoney [45], Prerau [34], Tardón et al. [29], Pacha [46], Rebelo et al. [12], Ng and Boyle [4], Choudhury et al. [47], Bainbridge and Bell [10], Fornés et al. [39, 13], Huang et al. [28] Fujinaga [2], Wen et al. [22], Pacha et al. [17, 24], Chen et al. [5], Gocke [11], Miyao and Nakano [48], Yun et al. [49]
Musical Information Reconstruction	Prerau [34], Pacha et al. [50, 51], Roy et al. [16], Bainbridge and Bell [10], Coüasnon et al. [3], Ng and Boyle [4], Baró et al. [52, 19] Calvo-Zaragoza et al. [21, 25, 27]
Musical Notation Model	Droettboom et al. [53], Chen et al. [5], Choudhury et al. [47], Ng et al. [9], Tardón [29], Bainbridge and Bell [10], Huang et al. [28]

Table 2: Summary of the studies carried in each of the OMR pipeline stages

in obtaining a more appropriate view in the object detection stage. Most of the digital images during the acquisition, transmission or processing are subject to noise. Both colour and brightness contain signals that carry random noise. Depending on the features of the image, different types of filters are used to remove some of the noise. During the process of binarisation, images are analysed to decide what is noise and what constitutes useful information for the task. Techniques to choose a binarisation threshold include global and adaptive methods. A global threshold is typically determined for the whole image, while for the adaptive threshold, local information in the image should be considered. Ng et al. [4] adapt the global threshold proposed by Ridler and Calvard [40]. While adaptive threshold is used in several more recent OMR studies too [13, 39, 11].

Gocke's [11] pipeline starts with a Gaussian filter, thenceforth a histogram on each colour channel is built. Then, the image is rotated to find the best angle that maximises horizontal projections. The image is then segmented into smaller 30x30 pixels, and a local threshold is found for each tile. Following threshold selection, all elements smaller than 4 pixels in diameter are removed, making the image clearer. The image is finally ready for staff-removal and symbol recognition. Local thresholding in this case yielded better results than the global one.

Similarly, in Fornés et al. [13] binarisation is followed by de-skewing using the Hough Transform [41]. A coarse approximation of the staff lines is obtained using median filters with horizontal masks to reconstruct the staff lines later. However, in this process, some residual colour information is retained, especially where the lines intersect with musical symbols, hence, some noise is still left. This approach is not robust to damaged paper.

Pinto et al. [14] propose a content-aware binarisation method for music scores. The model captures content-related information during the process from a greyscale image. It also extracts the staff line thickness and the vertical line distance in staff to guide binarisation. This algorithm tries to find a threshold that maximises the extracted content information from images. However, the performance hugely depends on the document characteristics, limiting performance across different documents.

Calvo-Zaragoza and Gallego [54, 55] propose using selectional auto-encoders [56] to learn an end-to-end transformation for binarisation. The network activation nodes indicate the likelihood of whether pixels are foreground or background pixels. Ensuing training, documents are parsed through the model and binarised using an appropriate global threshold. This approach performs better than the conventional binarisation methods in some document types. Nonetheless, errors happen around foreground strokes and are emphasised along edges of the input windows, due to the lack of context in the neighbourhood.

4. MUSIC SYMBOL RECOGNITION

The next stage typically constitutes dealing with musical symbol recognition. Here, the three main steps are staff processing, isolating music notations and finally, classification. Usually, staff lines are isolated first, then detected and finally removed from the images. The model then isolates the remaining notations as primitive elements. These are later used to extract features and feed those features to train the classifier.

4.1 Staff processing

Staff lines are a set of five horizontal paths from one side of the music score to the other. Each line and gap represent a different pitch. For better object detection, the question of staff line removal has been of prime importance. Researchers take two different approaches; one is only detecting and isolating them, while the other approach goes one step further in removing them.

While in printed sheet music, staff lines are straight, parallel and horizontal, in handwritten scores, these lines might be tilted, curved and may not be parallel at all. These lines might also look curved or skewed depending on the image skew angle [12] or the degradation of the paper. The model needs to separate staff lines from actual music objects. Since the lines overlap with musical objects, simply cutting and removing them degrades the notes and make them harder to recognise, further limiting performance.

Consequently, an increasing number of studies take the approach of removing the staff lines in a more intelligent fashion [4, 47, 57, 10, 39, 13, 29, 28, 22]. In this section, we outline typical staff line processing approaches. Blostein and Baird [57] suggests using horizontal projections of the black pixels and finding their maxima. The drawback is that the method only considers horizontal straight lines. In order to deal with non-horizontal, the process is followed with image rotations and choosing an angle with a higher maxima.

Rebelo et al. (2007) [12] consider staff lines to be the shortest path between two horizontal page margins if those paths have black pixels throughout the entire path. The height between every two lines is first estimated and later used as a reference length for the following operations. Upon choosing an estimation, using the Dijkstra algorithm [58], the shortest path between the leftmost pixel and the rightmost pixel is found. Their method is robust to lines with some curvature and discontinuity since it follows continuous paths connecting line ends from both sides. However, this algorithm may sometimes retain paths that do not follow the staff line. This happens when there is a higher density of beamed notes, and the estimated path follows the beams or when the staff lines are very curved.

Cardoso et al. [43] propose stable paths, considering the sheet music image as a graph. The staff lines in the graph are the less costly paths between the left and right margins. Subsequently, the model should differentiate between score pixels and staff line pixels. This model is robust to discontinuities, skewness, curvature in staff lines and one-pixel thin staff lines. Both the shortest path and stable paths give a similar false detection rate of 1.4% and 1.3% respectively. However, the stable path approach is five times faster. This technique is often used in the preprocessing stage [27].

Another study [7] uses stable paths approach to extract staff line skeletons. Then, the line adjacency graph (LAG) [59] is used to cluster pixel runs generated from run-length encoding (RLE) of the image [60]. The last step involves removing clusters lying on the staff line. This step has two passes; the first step estimates the height line for each staff by averaging the section height being cut with the staff lines. The second pass filters out the noise left from the last pass. This method takes a similar approach with [61] grouping staff line pixels into segments.

Other studies follow the approach of keeping the staff lines during the next stages [9, 11, 62, 26, 16, 63]. They argue that the staff line removal task is very complex and often ends up being inaccurate and passes errors to the following stages. These studies usually detect and isolate staff lines ahead of object processing. Recent object detection studies show that removing staff lines does not add much improvement to this stage [17].

A more recent work [46] investigates how incremental learning can assist staff line detection using convolutional neural networks (CNNs) and human annotation. To begin with, a CNN model is fed a small amount of data with available annotations for training. Using this training, the model makes predictions on a larger dataset, and a human annotator rejects or accepts the predictions. The accepted predictions are added to the training dataset to repeat the process. This method enables the creation of a more extensive dataset. After four iterations, the dataset contains 70% annotated scores of the original set. One drawback of incremental learning is that if the annotator accepts samples with imperfect annotations, the error accumulates in each iteration, introducing inaccuracy, while it also needs a human annotator. This yields similar results with [43, 61], with a precision score of 0.95.

Despite the substantial research effort put into staff line removal, it is still far from being accurate in handwritten sheet music. Handwritten scores exhibit a wide variety in line length and distance, thickness, curvatures of staff lines and also the quality of the image.

4.2 Music symbol processing

The next step after removing the staff lines is to isolate the musical symbols. Staff line removal will strongly affect this step as it can cause fragmentation in the parts where staff lines and musical objects are tangent to each other. One widely used approach is hierarchical decomposition [37], where staff lines split a music sheet and then extract noteheads, rests stems and other notation elements [47, 53, 11, 48, 4]. Some approaches consider, for instance, a half-note instead of its primitives for the classification step. Mahoney [45] uses descriptors to choose the matching candidate between a set of candidates of symbol types. Carter [36] uses the line-adjacency graph (LAG) of an image for both removing the staff lines and providing a structural analysis of symbols. This technique helps in ob-

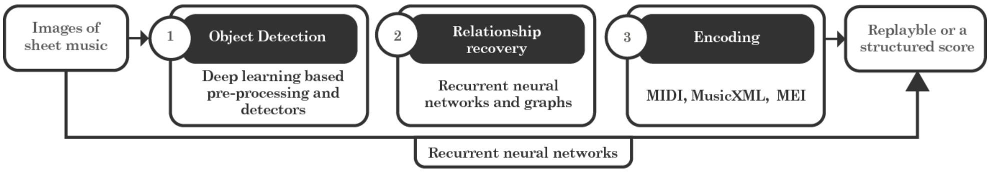


Figure 3: Typical OMR pipeline using deep neural networks

taining more consistent image sectioning, but it is limited to a small range of symbols as well as a potentially severe break-up of symbols.

Some studies skip segmentation and staff line removal [62, 63, 16] and use Hidden Markov Models (HMM). HMMs work on low-level features that are robust to poor quality images and can detect early topographic prints and handwritten pieces. Calvo-Zaragoza [63] split sheet music pages into staves following preprocessing. All staves are normalised and later represented as a sequence of feature vectors. This approach is very similar to [62], however, this study goes one step further and supports the HMM with a statistical N-gram model and achieve a 30% error rate. This performance could be further improved if lyrics are removed, light equalisation is performed and data variations are statistically modelled.

4.3 Music symbol classification

After the segmentation of musical primitives, the subsequent process is classification. Objects are classified based on their shapes and similarities. However, since these objects are very often densely packed and overlapping their shapes can become very complex. Therefore, this step is very sensitive to all possible variations in music notations. Fujinaga [2] uses projection profiles for classification, Gocke [11] uses template matching to classify the objects. Other methods used are support vector machines (SVMs), k-nearest neighbour (kNN), neural networks (NN) and hidden Markov models (HMM). A comparative study of the four methods [64], finds SVM performs better than HMMs.

Considering the success of deep neural networks (DNN) in many machine learning tasks, recent studies take this approach in music object recognition and classification. A typical pipeline is shown in Figure 3. These networks have many layers with activation functions employed before information propagates to the next layer. The deeper the model, the more complicated it gets and is able to detect hidden nonlinear relationships between the data, in this case, music objects. The problem with using DNNs in OMR is that they require a significant amount of labelled data for supervised training.

Object detection in images is a very active research field. Regional CNNs (R-CNNs), Faster R-CNN [65], U-nets [66], deep watershed detectors [18] and Single-shot detectors [67, 68] are among some of the approaches proposed recently. Pacha et al. [17] use Faster R-CNN networks with pre-trained models fine-tuned with data from Muscima++ (see Sect. 7 for a summary of OMR datasets). They achieve a mean average precision of up to 80 %. However, such performance is achieved with cropping the image into individual staff lines.

Tuggener et al. [18] use deep watershed detectors in the whole image. It is faster than Faster R-CNN approach in image snippets, and it allows some shift in the data distribution. Nonetheless, it does not perform well on underrepresented classes.

Going further into the pipeline, we should be able to capture and reconstruct the right positions, relationships between notes, and relevant musical semantic information such as duration, onsets, pitch.

5. NOTATION RECONSTRUCTION

After classifying and recognising musical objects, the next block should extract musical semantics and structure. As mentioned earlier, OMR is two-dimensional, meaning that recognising the note sequence as well as their spatial relationships are essential. Hence, a model should identify the information about the spatial relationship between the recognised objects. Ng et al. [9] believe that domain knowledge is key to improving OMR tasks and especially music object recognition, similarly to a trained copyist or engraver, to decipher poorly written scores, building on the authors' previous research on printed scores [4]. A multistage process is adopted, in which the first search is for essential features helping the interpretation of the score, verified by their mutual coherence, followed by a more intelligent search for more ambiguous features. Key and time signatures are detected after low-level processing and classification, using these global high-level features to test the earlier results.

Ng and Boyle [4] base their study on three assumptions: i) foreknowing the time signature, ii) key signature, and iii) that the set of the primitive feature set under examination is limited to ten. The first and second assumptions are overcome by geometrically predicting a limited symbol set such as numbers, flats and sharps. The input image goes through binarisation using a threshold, image rotation for de-skew, then the staff lines are detected and erased. Now the image has blocks of pixels, music object primitives and groups of primitives. Further segmentation based on some rules is needed for a group of primitives. After the segmentation process, a classifier uses only the width and the height of the bounding box for recognition based on a sampled training set. The recognised primitives are grouped to reconstruct their semantic meaning. The reconstruction consists of overlaying an ellipse and counting the number of foreground pixels, finding the pitch, search the neighbourhood for other features that might belong to the object and identifying the possible accidents using a nearest neighbourhood (NN) classifier. Music knowledge related to bars, time, and key signatures is applied at this stage. During segmentation, the process relies on straight edges of the objects, therefore is not robust to handwritten scores. The method fails if the symbols are skewed, for instance, when a stem is not perpendicular to a stave line.

Similar to the method mentioned above, another approach is formalising musical knowledge and/or encoding knowledge into grammar rules that explain, for instance, how primitives are to be processed or how graphical shapes are to be segmented [10, 3].

Prerau[34] proposes two levels of grammar. One being notational grammar while the other is a higher-level grammar for music. The first allows the recognition of symbol relationships, the second deals with larger music units. Many other techniques use musical rules to create grammar rules for OMR. Such rules can be exemplified as [37]:

- An accidental is placed before a notehead and at the same height;
- A dot is placed after or above a notehead in a variable distance;
- Between any pair of symbols: they cannot overlap.

The issue with music rules and heuristics is that these rules are very often violated, especially in handwritten music. Furthermore, it is challenging to create rules for many different variations and notations with a high level of complexity. As a result, this approach would not perform well with both typeset and handwritten complex notations, and it is difficult to scale to a broad range of notation and engraving styles.

Pacha et al. [50] propose using graphs to move towards a universal music representation. Considering that in music notations, the relationship between primitives contains the semantic meaning of each primitive; they suggest that OMR should employ a notation assembly stage to represent this relationship. Instead of using grammar and rules mentioned earlier, they use a machine learning approach to assemble a set of detected primitives. The assembly is similar to a graph containing syntactic relationships among primitives capturing the symbol configuration. The robustness of the model regarding variations in bounding boxes leaves room for improvement and so does the notation assembly stage, due to the lack of broader hypotheses on the detected objects.

Baró et al. [52] consider monophonic scores as sequences and use Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs) for reading such sequences to retrieve pitch and duration. For evaluation they use Symbol Error Rate (SER) defined as the minimum number of edit operation to convert an array to another. This approach shows to work well with simple scores such as monophonic scores, but fundamental remodelling is needed for more complex scores [52].

This stage reconstructs relationships, structure and semantics from the detected musical objects. A challenging problem in this stage is to model a musical output representation that encodes sheet music as a re-encoded score and the semantics (e.g. onsets, duration, pitch).

6. MUSIC NOTATION ENCODING

The output from the previous steps is used to construct a semantic model or data model. This model should represent a re-encoding of the score in the input. The output model should be expressible in a machine-readable format. Usual OMR output formats include MIDI, MusicXML, MEI, NIFF, Finale, and in some software, the music is even rendered into WAVE files. Musical Instrument Digital Interface (MIDI) [69] is an interchange medium between the computer and digital instruments. At the basic level, MIDI includes the temporal position when a note starts, stops, how loud the note is, the pitch of the note, instrument and channel. The main drawback of MIDI is that it cannot represent the relationships between musical symbols, or produce a re-encoded structured file, limiting the output to replayability only.

Notable formats that allow a structured encoding and storing notations include MusicXML [70, 71] and MEI [72, 73]. Both allow further editing in a music notation software. MusicXML is more focused on encoding notation layout. It is designed for archiving and for sharing sheet music between applications. There is ongoing research in the W3C Music Notation Community Group on improving MusicXML format to handle more specific tasks and applications.

The Music Encoding Initiative (MEI) [72] claims to be comprehensive, declarative, explicit and hierarchical. MEI has not been widely used as the final output of OMR systems yet. However, based on the characteristics mentioned above, MEI is able to capture and retain musical semantics better, e.g. relationships between voices, which may benefit music engraving.

There is also work converting OMR output into Semantic Web formats. Jones et. al. [74] propose the use of Linked Data to annotate and improve discovery of music scores using the Resource Description Framework (RDF). The captured information is limited to the number of voices, movements and melodies. Further extensions are needed to store more sophisticated music semantics that support harmony or melody analysis. Nevertheless, the use of Linked Data compatible formats may benefit OMR applications in multiple ways. Linking scores to other music related data on the Web [75] or even features of the audio of a performance [76] could support interactive applications such as score following or large catalogue navigation [77]. The ontologies governing these formats may be used to encode musical or engraving rules to complement probabilistic inference in machine learning models.

To decide which of the encodings to use, we have to think of what an application may require. Using the knowledge obtained in the previous steps and from different studies would assist this stage in its standardisation. Currently there is little research in OMR dealing with encoding, however, many works in other fields focus on encoding formats that better represent music and its structure.

7. DATASETS

Depending on the OMR task to be performed and the nature of the application, different datasets may be suitable. Existing datasets contain handwritten or copyright-free printed music sheets in mensural or CWMN notations. Calvo-Zaragoza et al. [78] introduced a new dataset called HOMUS (Handwritten Online Musical Symbols). This contains 15200 samples of 32 types of musical symbols from 100 different musicians. Universal Music Symbol Collection is a dataset of 90000 tiny handwritten and type-set music symbols from 79 classes that can be used to train classifiers.

As for staff line removal, a commonly used dataset is CVC-MUSCIMA [79]. It contains 1000 music sheets written by 50 different musicians. Each musician was asked to transcribe the same given 20 pages of music using the same pen and same style of sheet music paper. These pages include monophonic and polyphonic music, consisting of scores for solo instruments and music scores for choir and orchestra.

An upgraded version of CVC-MUSCIMA is MUSCIMA++ [15]. It is more suitable for musical symbol detection. It has 91255 symbols with both notation primitives and higher-level notation objects, key signatures or time signatures. Notes are captured using the annotated relationships of the primitives, having this way both low and high-level symbols. DeepScores is a collection that contains 300k annotated images of written music mainly for object classification, detection, and segmentation [80]. This dataset has large images containing tiny objects.

There are also datasets for an end-to-end recognition such as the Printed Images of Music Staves (PrIMuS) [21], or the extended version of this with distorted images to simulate imperfections Camera-PrIMuS [25]. These datasets have 87678 real-music scripts in five different formats: PNG, MIDI, MEI, semantic and agnostic encoding which is a sequence that contains the graphical symbols and their positions without any musical meaning.

Given that the performance of the deep learning methods usually depends on the amount of the data the model is fed, for future work, we propose creating a universal dataset that facilitates the intermediate stages but also an end-to-end system. We want to start by generating music files using a music notation software such as Dorico [81] or Rosegarden [82]. This work will be harmonized with the before-mentioned Muscima++ and DeepScores datasets.

8. OPEN ISSUES AND CONCLUSIONS

Low-quality images of sheet music, complex scores, handwritten music and alternate notations are still challenging for OMR, while most of the work focuses on monophonic scores. CWMN notation is highly complex, having dense scores, overlapping symbols, structural complexity, semantic rules that are sometimes violated. For a deep learning approach, in particular, class imbalance is one of the most significant issues; some note types are persistent while some others are rare. An further open issue is the lack of a large labelled dataset with a broad variety of image quality and balanced classes [83].

We can observe a shift in OMR from using conventional image processing and object detection to using neural networks, as shown in Figure 3. Recently published papers take novel approaches and use machine learning methods in all stages of the OMR pipeline. These stages are not necessarily in the order presented above or exhibit all the steps described.

Despite the introduction of deep learning, the field leaves space for improvement in all stages of the pipeline. New opportunities include creating more diverse and better balanced datasets, improving the detection of music objects and staff lines, the reconstruction of semantic meaning, and, perhaps most importantly, standardising the evaluation metrics and the output of the pipeline. A possible final goal is end-to-end learning that would not need intermediate steps. Neural networks are already applied to problems like text and speech recognition and machine translation in this manner. However, these systems are still not adapted to a two-dimensional output sequence such as music [31].

This paper summarised seminal and influential studies conducted in the field of OMR. We discussed different methods and approaches in prominent stages of the OMR pipeline. Our review aims to identify important older works and current state-of-the-art approaches, which can be used as a reference by researchers to begin further work in OMR. It also represents a position in several aspects of the field, including the need for incorporating more prior knowledge, theory and musical information in the processing pipeline, the need for finding new methods to incorporate these priors into statistical learning models such as deep neural networks and a need for more standardisation in OMR evaluation.

Acknowledgments

The authors acknowledge the support of the AI and Music CDT, funded by UKRI and EPSRC under grant agreement no. EP/S022694/1 and our industry partner Steinberg Media Technologies GmbH for their continuous support.

9. REFERENCES

[1] M. L. West, "The babylonian musical notation and the hurrian melodic texts," *Music & letters*, vol. 75, no. 2, pp. 161–179, 1994.

[2] I. Fujinaga, "Optical music recognition using projections," Ph.D. dissertation, McGill University Montreal, Canada, 1988.

[3] B. Coüasnon, P. Brisset, I. Stephan, and C. P. Brisset, "Using logic programming languages for optical music recognition," in *In Proceedings of the Third International Conference on The Practical Application of Prolog*. Citeseer, 1995.

[4] K. C. Ng and R. D. Boyle, "Recognition and reconstruction of primitives in music scores," *Image and Vision computing*, vol. 14, no. 1, pp. 39–46, 1996.

[5] G. Chen, L. Zhang, W. Zhang, and Q. Wang, "Detecting the staff-lines of musical score with hough transform and mathematical morphology," in *2010 International Conference on Multimedia Technology*, Oct 2010, pp. 1–4.

[6] V. H. C. Vidal, “Staffline detection and removal in the grayscale domain,” Ph.D. dissertation, Faculdade de Engenharia da Universidade do Porto, 2012.

[7] H.-N. Bui, I.-S. Na, and S.-H. Kim, “Staff line removal using line adjacency graph and staff line skeleton for camera-based printed music scores,” in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 2787–2789.

[8] Z. Huang, X. Jia, and Y. Guo, “State-of-the-art model for music object recognition with deep learning,” *Applied Sciences*, vol. 9, no. 13, p. 2645, 2019.

[9] K. Ng, D. Cooper, E. Stefani, R. Boyle, and N. Bailey, “Embracing the composer: Optical recognition of handwritten manuscripts.” in *ICMC*, 1999.

[10] D. Bainbridge and T. C. Bell, “A music notation construction engine for optical music recognition,” *Softw., Pract. Exper.*, vol. 33, pp. 173–200, 2003.

[11] R. Gocke, “Building a system for writer identification on handwritten music scores,” 2003. [Online]. Available: https://pdfs.semanticscholar.org/0bec/f150902ad48e080d0572dd5ef26686d442c6.pdf

[12] A. Rebelo, A. Capela, J. F. P. d. Costa, C. Guedes, E. Carrapatoso, and J. S. Cardoso, “A shortest path approach for staff line detection,” in *Third International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS’07)*, Nov 2007, pp. 79–85.

[13] A. Fornés, J. Lladós, G. Sánchez, and H. Bunke, “On the Use of Textural Features for Writer Identification in Old Handwritten Music Scores,” in *2009 10th International Conference on Document Analysis and Recognition*. Barcelona, Spain: IEEE, 2009, pp. 996–1000. [Online]. Available: http://ieeexplore.ieee.org/document/5277541/

[14] T. Pinto, A. Rebelo, G. Giraldi, and J. S. Cardoso, “Music score binarization based on domain knowledge,” in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2011, pp. 700–708.

[15] J. Hajič and P. Pecina, “The muscima++ dataset for handwritten optical music recognition,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 39–46.

[16] P. P. Roy, A. K. Bhunia, and U. Pal, “Hmm-based writer identification in music score documents without staff-line removal,” *Expert Systems with Applications*, vol. 89, pp. 222–240, 2017.

[17] A. Pacha, K.-Y. Choi, B. Couasnon, Y. Ricquebourg, R. Zanibbi, and H. Eidenberger, “Handwritten Music Object Detection: Open Issues and Baseline Results,” in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. Vienna: IEEE, Apr. 2018, pp. 163–168. [Online]. Available: https://ieeexplore.ieee.org/document/8395189/

[18] L. Tuggener, I. Elezi, J. Schmidhuber, and T. Stadelmann, “Deep watershed detector for music object recognition,” *arXiv preprint arXiv:1805.10548*, 2018.

[19] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, “From optical music recognition to handwritten music recognition: A baseline,” *Pattern Recognition Letters*, vol. 123, pp. 1–8, 2019.

[20] A. Baro, P. Riba, and A. Fornés, “Towards the recognition of compound music notes in handwritten music scores,” in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016, pp. 465–470.

[21] J. Calvo-Zaragoza and D. Rizo, “End-to-end neural optical music recognition of monophonic scores,” *Applied Sciences*, vol. 8, no. 4, p. 606, 2018.

[22] C. Wen, A. Rebelo, J. Zhang, and J. Cardoso, “A new optical music recognition system based on combined neural network,” *Pattern Recognition Letters*, vol. 58, pp. 1–7, 2015.

[23] A. Pacha and H. Eidenberger, “Towards Self-Learning Optical Music Recognition,” in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Cancun, Mexico: IEEE, Dec. 2017, pp. 795–800. [Online]. Available: http://ieeexplore.ieee.org/document/8260732/

[24] ——, “Towards a universal music symbol classifier,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2. IEEE, 2017, pp. 35–36.

[25] J. Calvo-Zaragoza and D. Rizo, “Camera-primus: Neural end-to-end optical music recognition on realistic monophonic scores.” in *ISMIR*, 2018, pp. 248–255.

[26] J. Calvo-Zaragoza, I. Barbancho, L. J. Tardón, and A. M. Barbancho, “Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation,” *Pattern Analysis and Applications*, vol. 18, no. 4, pp. 933–943, 2015.

[27] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, “Handwritten music recognition for mensural notation: formulation, data and baseline results,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1081–1086.

[28] Y.-H. Huang, X. Chen, S. Beck, D. Burn, and L. J. Van Gool, “Automatic handwritten mensural notation interpreter: From manuscript to midi performance.” in *ISMIR*, 2015, pp. 79–85.

[29] L. J. Tardón, S. Sammartino, I. Barbancho, V. Gómez, and A. Oliver, “Optical music recognition for scores written in white mensural notation,” *EURASIP Journal on Image and Video Processing*, vol. 2009, no. 1, p. 843401, 2009.

[30] D. Byrd and J. G. Simonsen, "Towards a standard testbed for optical music recognition: Definitions, metrics, and page images," *Journal of New Music Research*, vol. 44, no. 3, pp. 169–195, 2015.

[31] J. Calvo-Zaragoza, J. Hajič Jr., and A. Pacha, "Understanding Optical Music Recognition," *arXiv:1908.03608 [cs, eess]*, Aug. 2019, arXiv: 1908.03608. [Online]. Available: http://arxiv.org/abs/1908.03608

[32] G. Jones, B. Ong, I. Bruno, and N. Kia, "Optical music imaging: music document digitisation, recognition, evaluation, and restoration," in *Interactive multimedia music technologies*. IGI Global, 2008, pp. 50–79.

[33] D. Pruslin, "Automatic recognition of sheet music, 1966. a critical survey of music image analysis," *Structured document image analysis. Springer, Heidelberg*, pp. 405–434, 1992.

[34] P. D, "Computer pattern recognition of standard engraved music notation," in *Structured Document Image Analysis*. Springer, 1992, pp. 405–434.

[35] M. Kassler, "Optical character-recognition of printed music: A review of two dissertations," 1972.

[36] N. P. Carter, "Automatic recognition of printed music in the context of electronic publishing." Ph.D. dissertation, University of Surrey (United Kingdom)., 1989.

[37] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso, "Optical music recognition: state-of-the-art and open issues," *Int J Multimed Info Retr*, vol. 1, no. 3, pp. 173–190, Oct. 2012. [Online]. Available: http://link.springer.com/10.1007/s13735-012-0004-6

[38] I. Fujinaga, "Staff detection and removal," in *Visual Perception of Music Notation: On-Line and Off Line Recognition*. IGI Global, 2004, pp. 1–39.

[39] A. Fornés, J. Lladós, G. Sánchez, and H. Bunke, "Writer identification in old handwritten music scores," in *2008 The Eighth IAPR International Workshop on Document Analysis Systems*. IEEE, 2008, pp. 347–353.

[40] T. Ridler, S. Calvard *et al.*, "Picture thresholding using an iterative selection method," *IEEE transactions on Systems, Man and Cybernetics*, vol. 8, no. 8, pp. 630–632, 1978.

[41] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[42] D. Bainbridge and T. Bell, "Identifying music documents in a collection of images," in *ISMIR*. University of Canterbury. Computer Science and Software Engineering., 2006, pp. 47–52.

[43] J. dos Santos Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. Pinto da Costa, "Staff Detection with Stable Paths," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1134–1139, Jun. 2009. [Online]. Available: http://ieeexplore.ieee.org/document/4775904/

[44] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A Comparative Study of Staff Removal Algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 753–766, May 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4359356/

[45] J. V. Mahoney, "Automatic analysis of music score images," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering ..., 1982.

[46] A. Pacha, "Incremental supervised staff detection," in *Proceedings of the 2nd International Workshop on Reading Music Systems, Delft, 2019*. WORMs, 2019. [Online]. Available: https://alexanderpacha.files.wordpress.com/2019/11/2019-incremental-supervised-staff-detection.pdf

[47] G. S. Choudhury, M. Droetboom, T. DiLauro, I. Fujinaga, and B. Harrington, "Optical music recognition system within a large-scale digitization project." in *ISMIR*, 2000.

[48] H. MIYAO and Y. NAKANO, "Note symbol extraction for printed piano scores using neural networks," *IEICE TRANSACTIONS on Information and Systems*, vol. 79, no. 5, pp. 548–554, 1996.

[49] K. Yun, A. Huyen, and T. Lu, "Deep neural networks for pattern recognition," *arXiv preprint arXiv:1809.09645*, 2018.

[50] A. Pacha, J. Calvo-Zaragoza, and J. Hajic jr, "Learning notation graph construction for full-pipeline optical music recognition," in *20th International Society for Music Information Retrieval Conference*, 2019.

[51] A. Pacha and J. Calvo-Zaragoza, "Optical music recognition in mensural notation with region-based convolutional neural networks." in *ISMIR*, 2018, pp. 240–247.

[52] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, "Optical music recognition by long short-term memory networks," in *International Workshop on Graphics Recognition*. Springer, 2017, pp. 81–95.

[53] M. Droettboom, I. Fujinaga, and K. MacMillan, "Optical music interpretation," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2002, pp. 378–387.

[54] J. Calvo-Zaragoza and A.-J. Gallego, "A selectional auto-encoder approach for document image binarization," *Pattern Recognition*, vol. 86, pp. 37–47, 2018.

[55] A.-J. Gallego and J. Calvo-Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, pp. 138–148, 2017.

[56] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International conference on artificial neural networks*. Springer, 2011, pp. 52–59.

[57] D. Blostein and H. S. Baird, "A critical survey of music image analysis," in *Structured Document Image Analysis*. Springer, 1992, pp. 405–434.

[58] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[59] S. Iliescu, R. Shinghal, and R. Y.-M. Teo, "Proposed heuristic procedures to preprocess character patterns using line adjacency graphs," *Pattern recognition*, vol. 29, no. 6, pp. 951–975, 1996.

[60] T. Tsukiyama, Y. Kondo, K. Kakuse, S. Saba, S. Ozaki, and K. Itoh, "Method and system for data compression and restoration," Apr. 29 1986, uS Patent 4,586,027.

[61] N. P. Carter and R. A. Bacon, "Automatic recognition of printed music," in *Structured Document Image Analysis*. Springer, 1992, pp. 456–465.

[62] L. Pugin, "Optical music recognitoin of early typographic prints using hidden markov models." in *ISMIR*, 2006, pp. 53–56.

[63] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Early handwritten music recognition with hidden markov models," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016, pp. 319–324.

[64] A. Rebelo, G. Capela, and J. S. Cardoso, "Optical recognition of music symbols," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 13, no. 1, pp. 19–31, 2010.

[65] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[66] J. Hajic Jr, M. Dorfer, G. Widmer, and P. Pecina, "Towards full-pipeline handwritten omr with musical symbol detection by u-nets." in *ISMIR*, 2018, pp. 225–232.

[67] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[68] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[69] J. Rothstein, *MIDI: A comprehensive introduction*. AR Editions, Inc., 1995, vol. 7.

[70] M. Good, "Musicxml for notation and analysis," *The virtual score: representation, retrieval, restoration*, vol. 12, pp. 113–124, 2001.

[71] M. Good and L. Recordare, "Lessons from the adoption of musicxml as an interchange standard," in *Proceedings of XML*, 2006, pp. 5–7.

[72] P. Roland, "The music encoding initiative (mei)," in *Proceedings of the First International Conference on Musical Applications Using XML*, vol. 1060, 2002, pp. 55–59.

[73] A. Hankinson, P. Roland, and I. Fujinaga, "The music encoding initiative as a document-encoding framework." in *ISMIR*, 2011, pp. 293–298.

[74] J. Jones, D. de Siqueira Braga, K. Tertuliano, and T. Kauppinen, "Musicowl: the music score ontology," in *Proceedings of the International Conference on Web Intelligence*, 2017, pp. 1222–1229.

[75] G. Fazekas, Y. Raimond, K. Jakobson, and M. Sandler, "An overview of Semantic Web activities in the OMRAS2 Project," *Journal of New Music Research special issue on Music Informatics and the OMRAS2 Project*, vol. 39, no. 4, pp. 295–311, 2011.

[76] A. Allik, G. Fazekas, and M. Sandler, "An ontology for audio features," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2016, pp. 73–79.

[77] L. Turchet, J. Pauwels, C. Fischione, and G. Fazekas, "Cloud-smart musical instrument interactions: Querying a large music collection with a smart guitar," *ACM Transactions on the Internet of Things (In Press)*, 2020.

[78] J. Calvo-Zaragoza and J. Oncina, "Recognition of pen-based music notation: the homus dataset," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 3038–3043.

[79] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, "Cvc-muscima: a ground truth of handwritten music score images for writer identification and staff removal," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 3, pp. 243–251, 2012.

[80] L. Tuggener, I. Elezi, J. Schmidhuber, M. Pelillo, and T. Stadelmann, "Deepscores-a dataset for segmentation, detection and classification of tiny objects," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3704–3709.

[81] Steinberg, "Dorico," https://new.steinberg.net/dorico/ (Jan. 2020).

[82] C. Cannam, R. Bown, and G. Laurent, "Rosegarden," https://www.rosegardenmusic.com/ (Jan. 2020).

[83] J. Novotnỳ and J. Pokornỳ, "Introduction to optical music recognition: Overview and practical challenges." in *DATESO*, 2015, pp. 65–76.

EXTENDING NOTATION THROUGH EMBODIED RESEARCH

Solomiya Moroz
University of Huddersfield
Solomiya.Moroz@hud.ac.uk

ABSTRACT

Through engagement with embodied research, I challenge the use of notation as part of the 'paradigm of reproduction' [1] in which notation plays a central role in the musical work concept. In my work, I propose new collaborative methods which place an accent on performers' response and embodied memory, thus I anchor the idea of a work with collaborators of my projects in addition to any other methods of mediation such as a notated score. In this paper, I would like to discuss two of my latest works, *On Fragments* and *Motion Studies*, which rely on performers' embodied memory in order to execute the works.

1. INTRODUCTION

My concern for composition with the performers' physical gestures and embodiment came from my work with live electronics, where I started to embrace collaboration. The need for feedback on how the technology was working led to back-and-forth exchanges which led to further collaboration. Thus, I understood that composing with gestural controllers could introduce openness within a musical structure where performer's improvised movement contributed to the compositional process. In this process, producing new gestures takes place through embodiment. For me, embodiment is a technique of playing an instrument or a character, and as a practice where new instances of embodiment are generated through cross modal associations from performer's interpretation of audio or video documents. I call such instances of new knowledge imaginary gestures. The latter definition of embodiment relates directly to Ben Spatz's in *What a Body Can Do* [2], where one treats technique that anyone's body acquires as knowledge and practice as research that one engages with in order to gain insight into new embodied knowledge.

Traditionally, once the work is created, it assumes ideals related to the conditions of its reproduction and presentation. In *Beyond the Score* the musicologist Nicolas Cook terms this the 'paradigm of reproduction'. In this model, music is communicated through written notation and the performers mediate the composer's ideas to listeners who are expecting an adequate reproduction of the score in which the composer's intentions are located [1]. Through embodied research, my work assumes a different ontology to that of traditional chamber music because of its involvement with extensive collaboration, the search for new approaches outside such paradigm and the different possibilities for presenting work as part of the concert performance ritual. Here, I present a couple of my works which challenge the notion of score as part of the musical work concept.

2. TWO WORKS

2.1. On Fragments

In *On Fragments*, I treat the score like a script with performance instructions rather than a document of authority. I devised the score with nine scenes indicating changing setup configurations, and instructions for playing and movement. In *On Fragments*, I introduce sections which are based on field recordings from construction sites in a southwestern neighbourhood of Montreal. In the collaborative process, I asked the saxophone players from the Quasar saxophone quartet to imitate these field recordings both sonically on saxophones and physically with movements of their bodies. Later, I used their interpretation of the field recordings both as audio and gestural material to be included in different open scenes of the score which follow on from the notated sections. The embodied field recording sections would be different if the field recordings were interpreted by a different saxophone quartet because both sonic and gestural material would be based on responses from different musicians. Moreover, the final section of the piece is graphically notated and gives players the freedom to replace it with their own improvisation in response to the piece. The graphic score gives suggestions in terms of interacting in a quartet format between players, the field recordings used in the piece and the processing effects included in the electronic patch of the piece.

Initially, I was interested to see how saxophone players could reproduce field recording sounds and orchestrate them within the ensemble. Thus, the idea of self-organisation is present at the level of interpretation of the original recorded material. Since field recordings of construction work, sounds of industrial fans and trains are non-idiomatic to saxophone playing, it was fascinating to hear their reproduction on saxophones. After receiving the recordings, I decided at first to work with them as transcriptions. However, I later realised that essentially the players became embodied carriers of these sounds, thus, I could also compose directly by asking them to reproduce their interpretation of certain recordings in some places in the score.

Altogether, I sent Quasar nine recordings from different locations in Griffintown. In my research into imaginary gestures and embodiment of sounds, I was interested in the poietic physical responses to abstract sounds for the varied visual possibilities of composing with them. Thus, it was important to present the players with both complex and simpler sounds which could be embodied. For this reason, two of the recordings were processed through a max patch with an FFT filter where the amplitudes of certain bands were exaggerated, so that regular fan sounds became hybrid sound versions of industrial turbines (video documentation 5:45–6:15 and 7:01–7:27) [4]. For tracks 1, 2, 4, 6, 7 and 8, I asked the players to react physically by embodying the kind of movements that they imagined were associated with different tracks without playing their saxophones. I also asked them not to imitate each other and to avoid similarity between themselves. For tracks 3, 5, 9, I asked them to choose one movement that they could all agree on to perform together. The aspect of self-organisation here also helped with the overall ethos of the piece where I was leading the performers to contribute to compositional process without me telling them exactly how to execute each step. As the piece was collaborated on over distance (myself in the UK and Quasar in Montreal), I wanted the gestural response to be as natural as possible for the players without my external involvement in rehearsals.

The possibility for varied responses from different performers to sound samples of abstract or simple quality seemed an especially interesting way to generate varied responses as compositional material. Above all, the movements, however abstract or direct, added many different layers of interpretation to each recording. They became compositional material and part of the extra-musical content in the piece. I developed the piece by layering composed sections, original field recordings, their physical and sonic embodiment, and text about the state of labour economy from Paul Mason's Postcapitalism: A Guide to Our Future.

As seen in *On Fragments,* most of my works are incomplete when delineated through the musical notation only because they are composed through collaboration with performers where embodied sound and movement is retained in the memory of the performers, and act as living scores [3]. In those cases where the score is to be performed by a different performer, there are also additional forms of mediation that will need to be carried out, such as a new performer creating their own responses to audio or video. In addition, I do not have a single ideal reproduction because some of my works are ephemeral, based on specific performers and situations.

2.2. Motion Studies

In my work, new instances of embodied response can be constantly generated in an open interdisciplinary collaboration. My previous research on imaginary gestures, in which embodied movement was derived from listening and reacting to different sound files with musicians of the Quasar Saxophone Quartet [4], served as an impetus for the new research in combination with theatre researchers for whom embodiment as theatrical practice comes from a post-Grotowski lineage of physical theatre.

In the lab sessions, we were two musicians and two actors focusing on how practitioners from both disciplines respond, influence and react to each other's sound and movement in space. Throughout the lab sessions we looked at the possibility of recalling initially improvised movements and sounds in order to generate new instances of an open score work. Eventually, each participant's embodied knowledge in combination with embodied memory of the movements learned in the lab-sessions became the embodied score of *Motion Studies* [5].

In *Motion Studies*, we chose the initial structure to be an open session, like in the post-Grotowski practice where the emphasis is on embodied research as part of a lab environment [6]. It is relevant to notice that the theatre researchers helped in the dislocation of discipline-specific boundaries, since their embodied movement techniques and practices were spilling into the workflow of the lab sessions and extending the boundaries of the open musical work. On the other hand, the musicians' instrumental improvisations were influencing theatre participants' vocal response because the melodic and textural materials of sound were becoming sonically embodied and were open to change during the performance. In addition, the musicians, Colin Frank and myself, were open to a lot of different types of improvisation because of our previous background in musical improvisation and in interdisciplinary collaborations: thus moving and improvising also with our bodies seemed natural and normal.

The new instances of embodiment that we learned in the lab sessions were discovered through improvisation in pairs. This led to an easily repeatable technique where one member of the pair leads the other through sound (leading movement) or through movement (leading sound). Three main instances of repeatable movement and sound combinations emerged:

- Linear movement: accompanied by percussive sounds with linear square-like movements in space where pairings of performers are initially observed (Excerpt 1)

- Stretched out vocal section with high leaps, accompanied by slow movement, and where cymbal is usually used somewhere in the performance space (Excerpt 2)

- Circular movements: which could be carried out in pairs in which members alternate

> leadership roles between pairs where sound leads movement and vice versa (Excerpt 3)

A useful tool in our lab sessions became video documentation with a camera in a fixed position where it became a supplement for reflection and further composition. Another useful tool was reflection on the phenomenological presence of oneself during the improvisation in discussion with the other participants, during which we recorded our affects and feelings and analysed the relationships between each other during the improvisation to uncover which sensations and affects were important and interesting to explore further. Thus, we shared our reflections on each other's actions within the group and how we perceived they affected our sound and movement. The technical language in these exchanges became less important than the language concerning our personal multi-sensory experiences in relation to each other.

Our interdisciplinary improvisation became a ground for knowledge exchange amongst the group to do with spatial awareness, movement and sound composition in real time. This knowledge started to spill from one discipline to another as our responses became quickly entangled. Thinking about the philosophical implications of our lab sessions helped to ground our embodied actions within a larger structure of aesthetic considerations to do with the performance aspect of the work. Conceptually, thinking in terms of Deleuze's 'packets of sensation' as a boundary object of the open score work helped:

> Percepts aren't perceptions. They're packets of sensations and relations that live independently of whoever experiences them. Affects aren't feelings, they are becomings that spill over beyond whoever lives through them (thereby becoming someone else) ... Affects, percepts, and concepts are three inseparable forces, running from art into philosophy and from philosophy into art. [7]

Thus, 'packets of sensations' is what the repeatable embodied instances of sound and movement became in our collective work when they belonged to experimental instances of improvisation. These are not concrete knowledge but rather a phenomenological reflection for each participant on the physical and sonic actions in the moment of improvisation, what they are and what they could be in future reproductions.

The working methods of our lab-sessions could be compared to that of devising theatre-dance companies whose works collectively reflect collaborative working methods. William Forsythe's Dance Company uses similar working methods where there is no dramaturg or a final dance score of the production. The dramaturgies of Forsythe's pieces are usually distributed among individual and shared dramaturgical practices across different spaces and times utilising boundary objects in place of a specific dance dramaturgy [8]. In *Distributed Dramaturgies: Navigating with Boundary Objects* on Forsythe's Dance Company's dramaturgical process, Vass-Rhee defines boundary objects as "objects or concepts, which, although jointly deployed by members of a community, are utilised differently by different participants" [8]. The boundary objects in Vass-Rhee's view need to be flexible enough yet contain adequate detail to be recognised by multiple collaborators. In addition, Forsythe was known to practice dramaturgical silence in the devising process of the work. Thus, the boundary objects and his dramaturgical silence created radically open dramaturgy for both the participants and spectators where the boundary objects remained open to recognition and interpretation among different participants of the work. Similarly, "packets of sensation" which are individually experienced in different repeated embodied instances (Excerpts 1-3) of the embodied score of *Motion Studies* became our boundary objects. They contained enough variance in the interpretation by different members of the group, in addition to containing many details for their recognition between the participants.

In our work, an open embodied score based on boundary objects of "packets of sensations" [7] came first followed by conceptual responses later. These responses encompassed conceptual thinking when it came to aesthetic decisions of how to present the work again. Here, I present the lighting considerations which are based on boundary objects of our repeatable movements as demonstrated in this diagram for a performance at the REVERB series in Ormskirk. However, our conceptual aesthetic considerations could be made in reference to other features of a new performance space and not only for the lights.

Please note, the following are not parts of a musical or dramaturgical score but rather examples of spatial and lighting considerations that could be employed in preparation for a performance.

Figure 1 shows lighting considerations for the first scene where linear movements informed a lighting scene composed of spotlights on stands projected in three straight lines from sides of the stage.

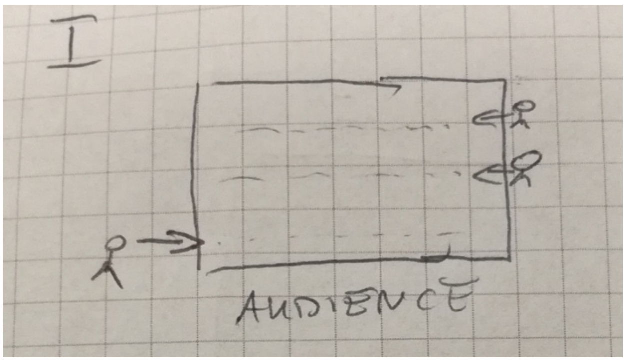


Figure 1. Scene I, linear movements (I. Krawczyk).

Figure 2 shows Scene II, where circular movements grow with more encounters between the pairs and slower interaction that could happen at the centre of the stage. These movements suggested spot lighting from above forming a larger circle.

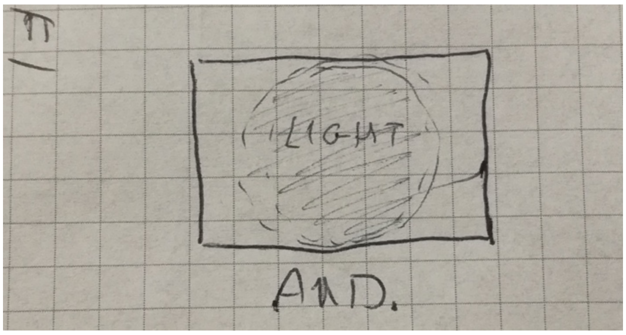


Figure 2. Scene II, circular movements (I. Krawczyk).

Third lighting scene (Figure 3) is more experimental in our structure as it relates to the kinetic energy that our embodied interactions generate. Our interactions in the third scene were represented through high energy leaps both in sound and movement where bodies could be coming in and out of the vertical corridor of light projected from the back and front of the stage.

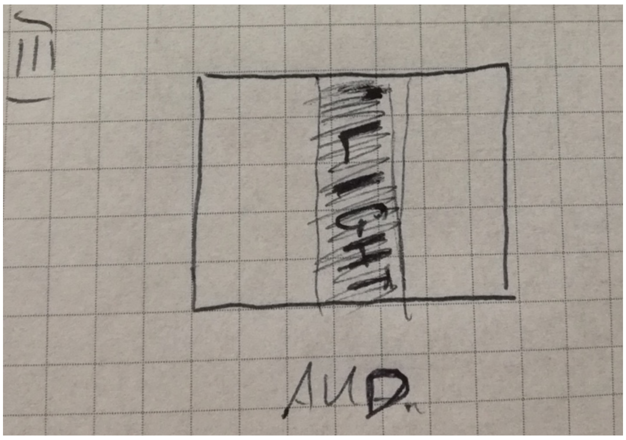


Figure 3. Scene III, vertical corridor (I. Krawczyk*)*.

Motion Studies is a process-based work that develops with each performance, and one with a flexible structure, where 'affects, percepts and concepts' [7] can flow in and out of each other within a conceptual framework chosen regarding a new venue.

3. CONCLUSION

Thus, both works, *Motion Studies* and *On Fragments* confirm the retainability of embodied knowledge where performers became carriers of this knowledge in relation to each other. In these works, a score is not the only set of instructions in order for the performance to take place. *Motion Studies* is a process-based work that develops with each performance, and one with a flexible structure within a conceptual framework chosen regarding a new venue. Whereas in *On Fragments*, the saxophone players successfully retained the embodied memory of their movements proposed through collaborative research sessions. The repeatability of the movements has been retained for different performance situations as the work has been performed already several times over the period of two years.

4. REFERENCES

[1] N. Cook, *Beyond the Score: Music as Performance*. New York: Oxford University Press, 2013.

[2] B. Spatz, *What a Body Can Do: Technique as knowledge, practice as research.* London: Routledge, 2015.

[3] L. Nickel, *Living Scores: a Portfolio of Orally Transmitted Experimental Music Compositions.* Doctoral thesis, Bath Spa University, 2017.

[4] S. Moroz [Solomiya Moroz], (2017, 21 June), *On Fragments – Quasar Saxophone Quartet* [Video file], retrieved from https://youtu.be/5eJ7wS4kI-0

[5] S. Moroz [Solomiya Moroz], (2020, 2 March), *Motion Studies* [Video file], retrieved from https://youtu.be/mDwsGAa8kKo

[6] J. Grotowski and E. Barba, *Towards a Poor Theatre* (1st Routledge ed.), New York, 2002.

[7] G. Deleuze, *Negotiations*. Trans Martin Joughin. Columbia University Press, New York, 1995.

[8] F. Vass-Rhee, "Distributed Dramaturgies: Navigating with Boundary Objects," in P. Hansen and D. Callison (Eds.) Dance Dramaturgy, pp.87–123, 2015.

ERGODIC AND EMERGENT QUALITIES OF REAL-TIME SCORES. *ANNA & MARIE* AND GAMIFIED AUDIOVISUAL COMPOSITIONS

Andreas Pirchner
University of Music and Performing Arts Graz
andreas.pirchner@kug.ac.at

ABSTRACT

This paper describes the functionality and aesthetic implications of the real-time score-system developed for the composition *Anna & Marie* by Marko Ciciliani. It originates from the artistic research project GAPPP and was first performed at Donaueschinger Musiktage 2019. By referring to examples of historic tendencies towards non-linear scores, the terms ergodicity and emergence are introduced to the understanding of the score properties of the real-time virtual performance space. This first part is then exemplified by describing the ergodic score of *Anna & Marie*. A particularity of this work is that two violinists navigate avatars in two virtual 3D environments by their manner of playing. The environment offers distinct audiovisual situations distributed in the virtual space and is identified as a spatial score. The musicians' musical effort of spatially traversing the virtual performance space consequently allows the audiovisual gestalt of the performance to emerge. The entanglement of spatial score and symbolic score, generated and presented on tablets, and mediated by the performers, is shown to be a characteristic of the composition. It is investigated, how the emerging performances question a notion of ergodicity where a prior text is followed by a technology reproducing it. In conclusion, the group of categories of real-time scores is extended by ergodic emergent scores.

1. INTRODUCTION

1.1 Technology of Real-Time Scores

The progress in and increased availability of computer technology have allowed artists and engineers to realize a multitude of setups for generating and rendering scores for performers in real-time[1], thus expanding the notion of what Umberto Ecos [2] described using the term "open work". Artworks of GAPPP fall into the category of open works. Common characteristics are that they are ergodic, that they involve one or more human performers as well as nonhuman agents, and that they find individual ways of generating and communicating scores and instructions to the performers. They also adopt a variety of media technological setups, some of which I will describe.

Real-time technologies: Recent developments in frameworks for network and server technologies like node.js [3] have made it possible to build lightweight and dedicated network services for apps and browser apps. Simultaneously, the rise of JavaScript promoted the use of the language for front and back end equally, making browsers a preferable target for application development. Frameworks like Electron [4] provide the advantage of development independently from mobile and desktop operating systems in a single JavaScript, HTML and CSS environment. Recently, some of these technological developments have been adopted for real-time generation of scores in browser environments [5, 6].

Spatial technologies: At the same time, the advancements of open-source real-time 3D development platforms like Unity [7] allow composers to realize their audiovisual ideas in 3D as well as to explore new forms of the open work by adopting strategies from digital games. These artistic explorations have encouraged new and nontraditional ways of communicating score elements and instructions to the performers. The interplay of entities that serve the function of a score in the composition *Anna & Marie* draws on both the spatial and real-time technology strands of these developments. The symbolic score[2] generated in real-time is tightly interwoven with the characteristics of the virtual performance space that equally function as a score.

[1] For an overview of recent works see for example [1].
[2] The term "symbolic score" is used in a semiotic sense and refers to scores communicating playing instructions to performers by using visual symbolic systems. These systems can include traditional musical notation symbols or individual artistic inventions of symbolic scores, often labeled as experimental or avant-garde. The term will later be used to distinguish these symbolic representations from the notion of spatial scores.

As some of the underlying concepts can be traced back to notions advocated at the GAPPP project, a brief background of the project will be provided along with an argument for possibilities for describing characteristics of scores for a gamified musical work.

1.2 Gamified Audiovisual Performances

GAPPP (*Gamified Audiovisual Performance and Performance Practice*) is an artistic research project currently carried out at the Institute for Electronic Music and Acoustics (IEM) at the University for Music and Performing Art in Graz/Austria. One of the goals of GAPPP is to explore the potential of digital games to offer models for audiovisual compositions and performances. The research is carried out from three points of view: audience perception is researched and informed by musicology, digital games theory and methods of social sciences (Andreas Pirchner); contemporary electro-acoustic audiovisual composition (Marko Ciciliani); and performance practice (Barbara Lüneburg). In an iterative process, the researched artworks are commissioned and composed for the project. Composers decide which elements of digital games they want to adopt for their work. Common characteristics demonstrated are that the artworks are *ergodic* [8], that they involve one or more human performers in addition to virtual non-human actors, and that they find individual ways of generating and communicating scores and instructions to the performers by adopting a variety of media technological setups.

2. ERGODIC AND EMERGENT SCORES

2.1 From Linearity to Ergodic Spaces

The score as an ergodic text: The notion of a space of possibilities, in which the actual game performance unfolds[3] is common in digital game theory [9]. By the middle of the 20th century, the idea of making decisions and possibilities a part of compositions had gained some popularity among the avant-garde. Composers such as Henri Pousseur (*Scambi*, 1957), Earle Brown (*Twenty-five Pages*, 1953), and Karlheinz Stockhausen (*Klavierstück XI*, 1957) delved into the potentials offered by an open form. Pousseur was cited by Eco on the composition *Scambi* (1957): "Scambi is not so much a musical composition as a field of possibilities, an explicit invitation to exercise choice" [2].

As scores explored different symbolic notations, composers invented new ways of giving playing instructions. They asked performers to make their own decisions on how to proceed through modules of the composition. The performers were invited to shape the form of the composition's performance both actively and in situ. However, these scores usually were still physically fixed by the composer prior to performances; their medium was mostly ink on paper, and the scores themselves were not generated or altered in real-time. Nevertheless, these characteristics foreshadow what later would be theorized by Espen Aarseth [8] through adopting the term "ergodic" to literature and art in the 1990s. The rise of hypertext and the network metaphor further popularized this term, and it became an important concept in early computer game studies:

> *During the cybertextual process, the user will have effectuated a semiotic sequence, and this selective movement is a work of physical construction that the various concepts of 'reading' do not account for. This phenomenon I call ergodic, using a term appropriated from physics that derives from the Greek words ergon and hodos, meaning 'work' and 'path'. In ergodic literature, nontrivial effort is required to allow the reader to traverse the text.* [8]

Following Aarseth's argument that a cybertextual process is not equal to reading a text, the process of navigating through a score of musical modules or even a score generated in real-time likewise is not equal to reading a score from a sheet of paper and demands that the performers make a non-trivial effort.

Pierre Boulez describes the initiative he requests from musicians performing his open-form composition *Constellation-Miroir* (1958/1959) as follows: "Certain directions are obligatory, others optional, but *everything* must be played. In some ways, this *Constellation* is like the map of an unknown city […]. The itinerary is left to the interpreter's initiative; he must direct himself through a tight network of routes" [10]. Boulez asks performers to navigate through the routes mapped out by the composition. Based on this this brief historical outline, we will refer to specific scores that are rooted in traditions of Western avant-garde music as ergodic.

The score as an ergodic space: The score as an ergodic space: Both the notion of a "field" of possibilities (Pousseur) and the metaphor of the "map" (Boulez) suggest that the ergodic process expands the one-dimensional linearity of text and notation towards a two-dimensional space. While it was not uncommon for composers throughout the centuries to explore spatiality as a sonic parameter, the orientation towards fields and maps in playing instructions as a notational and conceptional tool marks a significant departure from traditional scores. In contemporary music, one-dimensionality is seen to be overcome not only through scores generated in real time (as described in [5]), but also through three-dimensional space and its augmentation towards virtual performance spaces. Referring to Tom Johnson's composition *Nine Bells* (1979), Marko Ciciliani [12] characterizes the performance space not only as a parameter for composition but also as a mode of the score. He further demonstrates that the understanding of space as score applies not only to physical space, but also to virtual spaces. The distribution of sound sources in space can shape a composition's time factor, depending on how performers traverse the composed spatial environment. With performers navigating a virtual open-world

[3] We assume that a space of possibilities defines a class of all possible manifestations of a system. By making decisions, performers move through this space.

environment, there is no determined point in time at which sounds will occur. However, the performers are not without direction or constraint. The design of the topology, for instance, promotes certain spatial connections. Accordingly, some sounds are more likely to be played than others, because their occurrence is facilitated while others are more difficult to achieve. Ciciliani concludes that "each decision concerning the design of a 3D environment, such as the inclusion of obstacles and passages, will have indirect or direct musical consequences," and consequently the virtual space takes on the role of a score. The full argument can be found in Ciciliani's chapter "Virtual 3D Environments as Composition and Performance Spaces." [12].

Softwares such as Unity even allows artists to manipulate and alter the virtual performance space through code at any time, and static physical rules do not necessarily apply. A striking example of this is Christof Ressi's *Terrain Study*, which is discussed in detail in the chapter "Visiting the Virtual" (Lüneburg). These options offered to the artists lead the arbitrary space generated in real time to be regarded as a spatial score that is equally generated in real time. The virtual performance space's openness to the arbitrary manipulation of its basic rulesets by code differs significantly from the way the physical performance space can be composed. These contemporary artistic practices continue the described historical concepts of the field of possibilities, open works, and corresponding techniques. However, they differ from earlier tradition in that the virtual spatial score is also part of the performance space; it is visible to the audience and becomes an integral part of the aesthetic experience of the composition.

Recent technology has made it feasible not only to compose score modules, but also to realize truly generative scores in a bottom-up process alongside the performance. The following section describes some fundamental characteristics of the resulting art and scores.

2.2 Emergence as a Characteristic of Scores

The term 'emergence' allows us to examine another specific characteristic of certain scores in ludified audiovisual performances. This term stems from philosophy and physics and describes a particular characteristic of complex systems. However, this text does not aim to contribute to the vibrant discourse in the philosophical field of emergentism, which pursues other epistemological and ontological goals. Instead, our argument builds on an understanding of emergence found in scientific fields such as biology and computer science. Here, the term is used to describe the effects of complex systems, distinguishing 'emergent' from 'resulting' effects. Resulting effects can be reduced to the sum of their individual causes, whereas emergent effects are characterized by the fact that neither can they be fully explained causally, nor are their expected system properties entirely predictable. For example, emergence was used in the theory of evolution to describe the problem that 'higher' properties of complex organisms cannot be fully explained by the interaction of the system properties and therefore differ by being novel.

Structures characterized as emergent can be based on simple rule systems. While swarm behavior, the formation of dunes, and the shape of snowflakes illustrate emergent structures occurring in nature, a classic example of such systems in the computational field is James Conway's *Game of Life* [13] (see figure 1). The original version of *Game of Life* (GoL) proposed by Conway in essence is deterministic. However, further developments introduced elements of chance, meaning that GoL no longer could be fully causally explained [14]. The notion of emergence here refers to the fact that unexpected and unforeseeable *gestalts* arise from the system of basic rules. The original version already was deterministic only in a mathematical sense, as the class of virtual objects that emerged, resulting in a rich taxonomy (with objects labeled "Pulsar," "Beacon," "Glider," "Heavy-weight spaceship," etc.), was certainly not predictable based on the underlying rules.

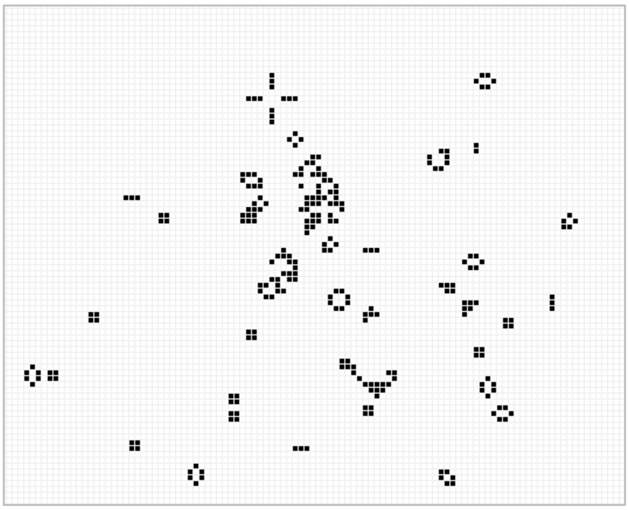


Figure 1. The visual structures of Game of Life are an example for a system where structures emerge from a basic set of rules (Screenshot of a JavaScript implemen-tation, github.com/pmav/game-of-life).

Translated to the situation of ergodic scores, the notion of emergence helps us to understand a bottom-up process where sets of basic rules constitute the space of possibility offered to the performer. Through their activity, performers traversing this space mediate which manifestations of the composition emerge. While in *Game of Life* mathematical rulesets allow unforeseen visual artifacts to emerge, real-time symbolic musical scores appear to inverse the direction of emergence. In these scores, visual symbols define an abstract set of rules that, as it stimulates the performer's activity, provides the fertile ground for the emergence of the musical performance.

Earle Brown's score for *December 1952* (see figure 2) exemplifies this notion. Brown asks the performers to follow a particular "path" as if in a two-dimensional open world. Of course, the flatland of the score itself does not offer sounding entities or sonic agents of the kind that contemporary composers might include.

In Christof Ressi's composition *Game Over* (2018) [15], however, the two-dimensional performance space is inhabited by sonic agents (see figure 3). An accelerometer augments the performer's clarinet, which functions as an interface, meaning that the performer is able to navigate the virtual performance space by moving the instrument. By exploring different parts of the 2D map and by interacting sonically with the virtual sonic agents of the virtual performance space, the performer contributes particularly actively to the way that the audiovisual performance emerges from the interplay of virtual performance space and virtual sonic agents.

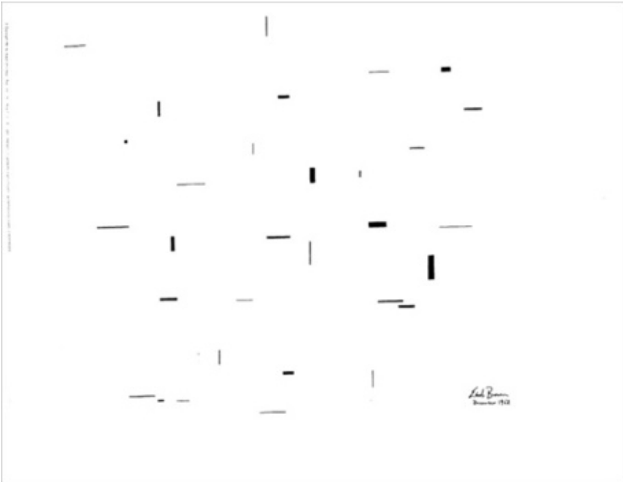


Figure 2. Earle Brown, December 1952.
Earle Brown Collection, Paul Sacher Foundation, Basel. © by Associated Music Publishers, New York, 1961

The notion of the score as an ergodic space gives rise to new explanations of how performances emerge in contemporary art. In *Kilgore* (Ciciliani, 2017/18) [16], the three-dimensional virtual performance spaces offer a multitude of ways to traverse space and interact with sonic agents. The performers' activity in these spaces allows the gestalt of the actual performance to emerge from the composed spatial, sonic, and agential properties.

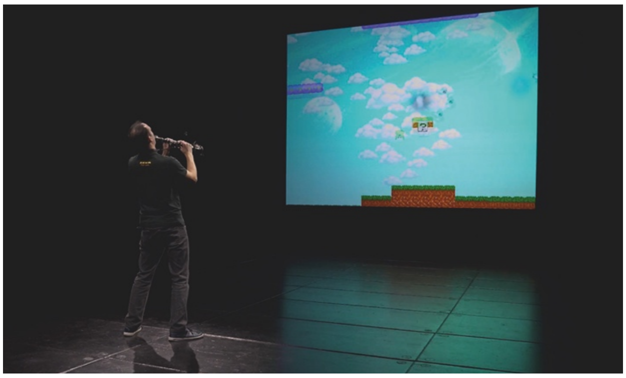


Figure 3. Christof Ressi Game Over (2018). © ndbewegtbild

The notion of this virtual spatial score explored by the performers thus differs significantly from the understanding of expanding one- or two-dimensional symbolic scores to three-dimensional score (as they are described for example in [17]). The interaction of the performers with the ruleset of the spatial score and thereby with sonic agents and the virtual world itself allows the musical work to emerge. In this way, spatiality, activity, and perceivable performance thereby become inseparably entangled.

2.3 Scored Reality – Reality as Score

Aarseth laid out a rich taxonomy for ergodic literature (including 2-D examples) which in turn computer game theory developed strongly towards technology. Here, the concept of ergodic cybertext (as explained before in 2.1) implicitly assumes three main components. First, there has to be a specific "text", or more generally, a space of possibilities that is open for reproduction and traversing. Second, there are players that traverse the text and explore the offered space of possibilities. And lastly, there has to be a technology capable of producing different manifestations according to the player's actions.

Figure 4. Barbara Lüneburg and Marko Ciciliani performing *Kilgore* (Marko Ciciliani, 2017/18) at Ars Electronica, Linz, 2018. © Andreas Pirchner

The reference to historical predecessors helped to find categories for the function of scores and virtual performance spaces used in ludified compositions. Their ergodic characteristics appear as twofold. On the one hand, performers can decide how to navigate through a score as a symbolic representation resembling a text like a field or a map. The score can be static, like in *December 1952* by Earle Brown, or real-time generated (as recently in [6]). On the other hand, the performance space itself, like in *Nine Bells* by Tom Johnson, may act as a score with performers traversing it and letting the composition emerge by their activity. Similarly, in several GAPPP compositions performers traverse virtual three-dimensional (audiovisual) performance spaces or encounter entanglements of physical and virtual spaces as for instance in *Terrain Study* by Christof Ressi. These special ergodic situations particularly support emergent qualities as a bottom-up process. Mediated by technology, the process is more fundamental for how the performance emerges than in the described rather modular historical predecessors. It promotes a close entanglement of ruleset (text), performance spaces (space of possibility) and performer (reader). The entanglement of elements of the performance space and non-human agents results in what for the present paper I want to call an "ergodic and emerging score".

3. THE COMPOSITION ANNA&MARIE

The following sections will briefly describe the composition *Anna & Marie* by Marko Ciciliani, based on the theoretical deliberations on emergence in open scores provided earlier and focusing on the aspects of the composition that are most relevant to the ergodic and emergent qualities of its real-time score system. A technical description of the different features of its score system and a subsequent analysis of its function and implications is provided.

Anna&Marie[4] was composed by Marko Ciciliani as part of the GAPPP project and was first performed at Donaueschinger Musiktage 2019 by performers Barbara Lüneburg and Susanne Scholz.

Narrative: The narrative elements of the composition unfold around the historical figures of two wax sculptors, Marie Marguerite Bihéron and Anna Morandi. Both were pioneer anatomists of the eighteenth century. The storyline speculates about a fictional meeting between both women and how it could have taken place. The composition asks the performers navigating inside this story to make decisions at specific junction points. These decisions affect how the narrative unfolds. Therefore, the narrative and the performance are ergodic, and the composition can be regarded as an example of an open work.

Performance spaces: The two performers, on Baroque violin and E-violin, traverse the virtual performance space in first-person view according to how they play their instruments. The three-dimensional environment was developed in Unity, and the individual view of each performer is projected on a screen. The virtual performance space holds separate topologies for each performer. While the performers move through the virtual space, their position in the physical performance space remains static.

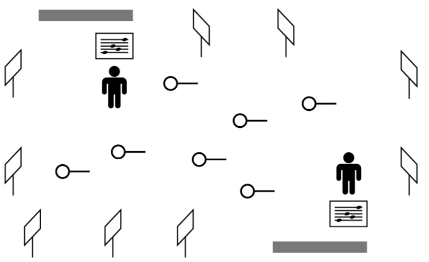


Figure 5. Elements of the physical performance space
Performers
Symbolic real-time score
Screens displaying virtual performance space
Audiovisual Augmented Reality Displays
Headphones for the audience

The audience sits on cushions distributed across the room. Wireless headphones are available to the audience that provide the spoken text narration of the composition. The setup includes panels that show additional images related to the topic of the narration. The audience is invited to use provided tablet devices to explore audiovisual Augmented Reality features of the panels (see Figure 5).

The composition comprises different layers of spatiality: (1) The physical performance space with the performers in fixed positions. (2) The virtual space, where the performers move by the way they improvise on their instruments. (3) Augmented reality panels that allow the audience to explore additional audiovisual elements individually. (4) Headphones lying in the room, allowing the audience to listen to the narrative parts of the performance individually.

Technological setup: The sound of both violins is analyzed in real-time by a computer program and fed into the performance system where synthetic sounds are generated in SuperCollider [18]. SuperCollider communicates with the Unity engine that generates the three-dimensional virtual performance spaces via Open Sound Control (OSC). SuperCollider also sends OSC messages to the dedicated score system generating symbols and playing instructions for each performer and rendering them on tablet displays.

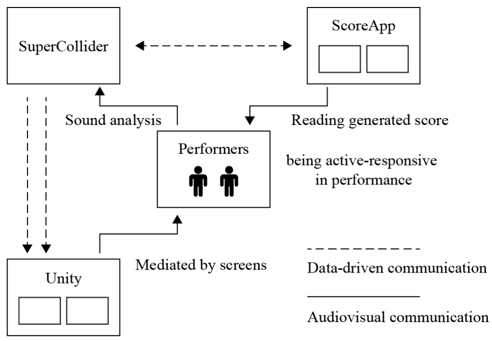


Figure 7. Basic technical setup of *Anna&Marie.*

On the one hand, *Anna & Marie* offers the performers a space of possibility to explore in a 3D environment. On the other hand, the performers' actions and decisions in the virtual space form the basis for generating and rendering the real-time score presented to them on tablet screens. The performers receive instructions on their manner of playing, dynamics, pitch sets, and how to relate to one another while playing. Within the confines of these instructions, they still have to make many individual choices on how to shape their parts in musical terms.

Navigating the spatial score: The performers traverse the spatial score/virtual performance space by playing their instruments (their sound is analyzed in real time and fed into the described performance system). Single long notes initiate turning points. If the notes following the long initial note are at a higher pitch than this note, this leads to a turn to the right, while notes below the initial note's pitch cause a turn to the left. When the players reach so-called junction points, they decide whether they walk past by it on its left or right side. This turn in the virtual space also marks a turn in the narrative and affects the way the relationship between the two main characters unfolds. The

[4] For further information see the website of the festival: https://www.swr.de/swrclassic/donaueschinger-musiktage/Donaueschinger-Musiktage-2019-Konzertante-Installation,veranstaltung-klangkunst-marko-ciciliani-19-10-2019-18-uhr-100.html

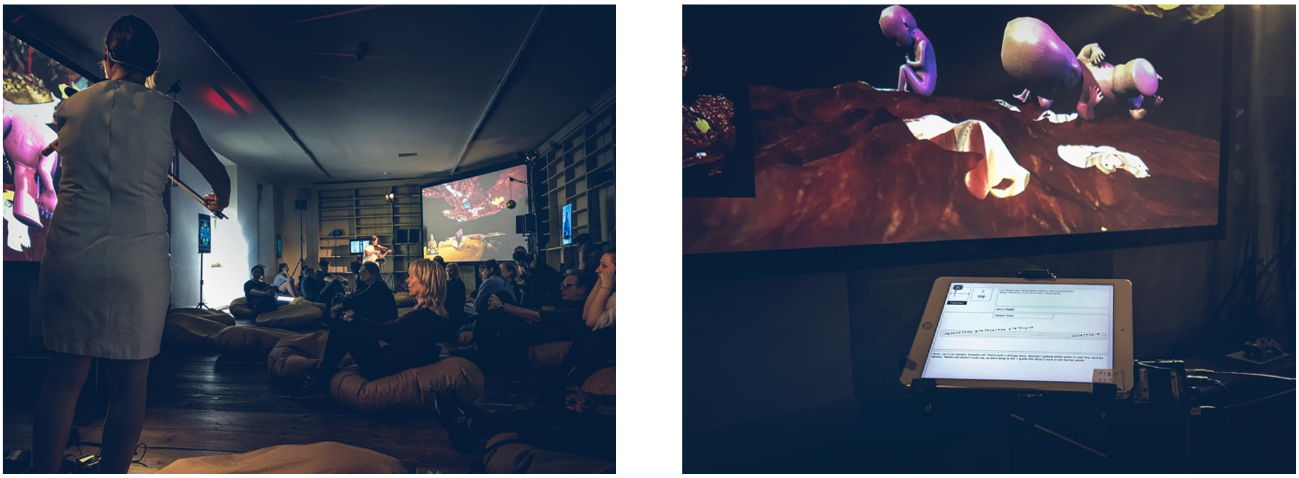


Figure 6. Left side: Performance situation in the library at Donaueschinger Musiktage 2019. Right side: Projection of the virtual performance space, with the symbolic real-time score in the front.

decision also affects the way the symbolic score and playing instructions are rendered on the tablets for the performers. The symbolic score is therefore dynamic; the characteristics of the spatial score and the behavior of the performers in it also affect the way in which the symbolic score is rendered. The following section provides a more detailed description of the manner in which this part of the composition's score is designed.

4. THE SYMBOLIC SCORE SYSTEM

The following part of the text describes the functionality of the score system in detail. Those readers less interested in technical aspects are encouraged to continue to part 5. In an attempt to keep the code sections short when possible, the text references the project repository on GitHub: github.com/asa-nerd/Anna-und-Marie.

4.1 Basic Design of the Score System

Based on previous experiences, due to the increasing role of JavaScript in network systems, and for greater flexibility, it was decided to develop the scoring system to run in web-browsers.

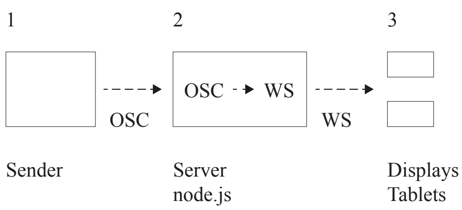


Figure 8. Schematic display of the developed symbolic scoring system.

The design of the system includes the following three modules (see Figure 8):

Sender (1): Ciciliani uses SuperCollider to send control messages to the system. Alternatively, any (musical) software that is capable of sending Open Sound Control (OSC) messages [19] can be used to communicate with the score system.

Host (2): The host is built on the node.js framework [3] and translates incoming OSC messages to messages that can be transmitted via the WebSocket (WS) protocol. It then distributes these messages to clients connected via (wireless) network. The software uses the node-osc [20] package to receive and process OSC messages and the socket.io [21] package to send WebSockets.

Display (3): Any device that is capable of running a web browser can be used as a client. In the case of the performances of *Anna & Marie*, the devices used were two iPads.

4.2 Desktop Host Application

The host is designed as an app that is available for Linux, iOS, and Windows operating systems. The Electron framework [4] was used to develop the executable app and its GUI. It can be run either on the same computer (localhost) that runs the sound software or on an independent machine. The host application establishes an HTTP-server using the express [22] package for node.js, allowing browsers to connect to the host and to receive the score templates. The goal was to makes it as convenient as possible to connect clients via network.

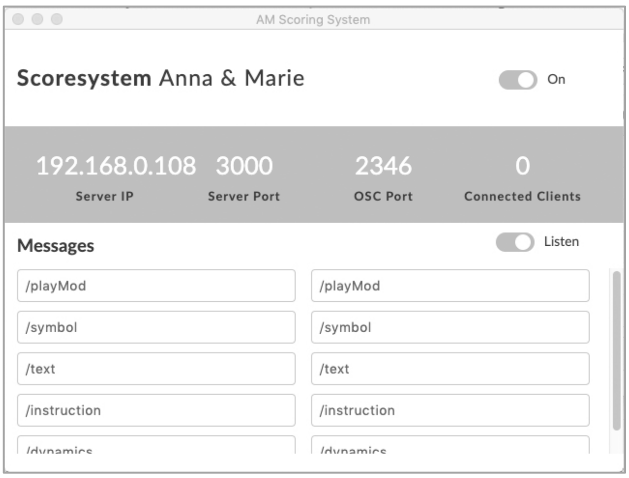


Figure 9. The graphical user interface of the host app.

The graphical user interface (see Figure 9) of the host application allows the configuration of the port number, shows the IP address used as the URL in the browsers of the client tablets, and offers settings to rename incoming OSC messages while translating them to WebSockets. One benefit is that this allows the host to be used not only to convert from OSC to WebSockets, but also to translate incoming commands according to the demands of the given project. A learn function traces all incoming OSC messages and displays them in a list. This is believed to facilitate the process of configuration if it becomes necessary to rename the messages.

4.3 Display for Performers

Any recent web browser can be used to display the score app to the performers in fullscreen-mode. CSS styling creates a floating and centered GUI that is adaptive to different display sizes and resolutions. Vector graphics were programmed using the Snap.svg library [23], making the display independent of resolution and responsive to screen sizes. The score displays the following sections for playing instructions (see Figure 10):

(1) Arrow symbols. These arrows function as instructions for how the musicians relate to each other in a chamber-musical sense.
(2) Dynamic symbols.
(3) Text field for playing instructions.
(4) Pitch sets for each of the two performers. The pitch sets consist of fragments of 6 different scales based on a microtonal scale.
(5) The transcription of the narration heard by the audience on the headphones.

Each of the performers are addressed by a separate color.

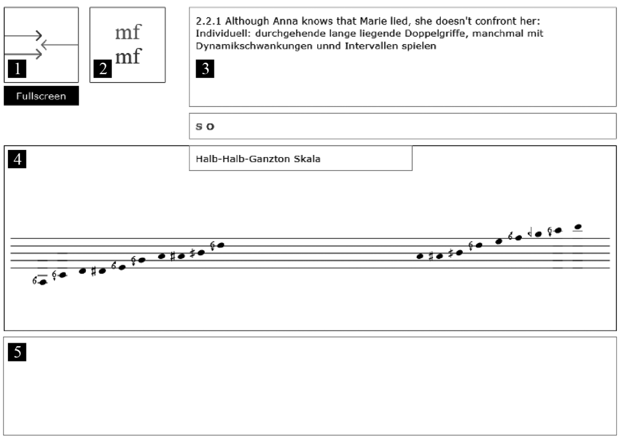


Figure 10. Display of the symbolic score

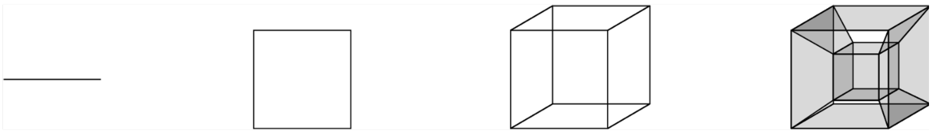
For the composer, the instructions in the text field (3) made up the main part of the symbolic score and were the key guidelines for the performers. It was essential that the instructions described the way the two musicians were to play music together, which formed the basis for everything else. After this, more detailed information about the kind of material and playing techniques followed. During the rehearsals, the composer and performers used the text field to take notes on how they negotiated and developed their parts together. This is thought to resemble taking notes and placing marks in printed scores, thus offering a similar feature in a generative score (e.g. by making marks on the touch display, like in apps displaying PDF scores) that could improve the experience and the usability for the performers.

5. CONCLUSION

Recent technological innovations and their widespread availability have allowed composers to invent new ways of combining elements of spatiality and score. This article has argued that both physical and virtual performance spaces can represent a score, and that scores can display characteristics of ergodicity and emergence. Table 1 illustrates a resulting systematic overview of score systems differing in their dimensionality, their medium and the resulting activity of performers.

Several compositions created as part of the GAPPP project display ergodic and emergent characteristics that are related to space in their score systems. These works include, for example, *game over* by Christof Ressi and *Kilgore* by Marko Ciciliani. *Anna & Marie*, however, appears to be in a special category, in which a symbolic real-time score on tablets is generated according to the performers' decisions in the virtual performance space. This results in a close composite of spatial and symbolic score. As both manifestations of the score depend on the basic ruleset of the composition, they are more than just linked. They emerge together in real time, thus differing significantly from the traditional notion of a top-down approach towards a score, which gives decision-making authority to the composer and the score she produces. In the type of score presented here, however, the virtual performance space itself emerges from the ruleset provided by the composition. The notion of a prior text, followed by a technology that reproduces it, is questioned by emergent qualities of scores exemplified by *Anna & Marie*. As the virtual performance space allows the rules of physical reality to be reconfigured, an abstract computational set of rules is provided with the composition. The performers traverse neither a text nor a score. As the performers are asked to move through the virtual space by playing freely according to rules largely determined by the virtual performance space, it is their playing that creates the next iterations of the symbolic score. Here, the performers do not simply traverse the (spatial and symbolic) ergodic score— they produce it at the same time.

The associated decision-making process is not primarily text based as in ergodic literature, but is largely musical, using note duration and pitch to start and make decisions. By taking decisions and traversing the spatial score, the human performers play an essential role in showing and exploring the composition and thereby creating the performance. Thus, performers mediating ergodic emergent scores assume increased agency.



Dimensionality	1-D	2-D	3-D	n-D
Metaphor	Linearity, temporality	Field, map, modularity	Space	Virtual space, arbitrary ruleset
Medium	Mostly Paper	Mostly Paper	Physical Reality	Virtual reality
Examples	Traditional scores	Boulez, Stockhausen	Nine Bells (Tom Johnson)	Anna&Marie, Terrain Study
Effort	Reading	Reading	Acting spatially	Acting spatially

Table 1. Spatial metaphors for notions of scores. (1) linear, one-dimensional. (2) field, map, two-dimensional (3) three-dimensional space (4) n-dimensional virtual space

The presented type of compositions was identified as producing a differentiated type of real-time score. In conclusion, we now will look at how this type fits in the taxonomy of real-time scores developed by Sandeep Bhagwati [24]. Based on Freeman [25], Bhagwati names four categories for real-time music scores:

- Permutational: Existent elements can be reordered by the performer at each performance (see *Constellation-Miroir*).
- Parametric: One or more parameters are left free to the performer.
- Auto-reflexive: The actions of the performer have an incidence on the unfolding of the piece.
- Co-creative: the conductor or the audience may contribute to the interaction with the score.

Bhagwati notes that contemporary real-time scores will to various degrees include aspects of all four categories. However, none of the ones listed would describe the strategies pointed out in this paper adequately. Another category seems necessary to complete the list, a category of real-time scores that exhibit the demonstrated characteristics of ergodicity and emergence:

- Emerging and ergodic: the score (symbolic or spatial) emerges ergodically from the basic rule set of the composition by mediation of the performers.

The suggested additional category classifies scores emerging from compositions that define abstract rules for spaces of possibility. These scores provide the performers with the agency to mediate the emergence of the work by freely traversing the space of possibility. Analogously to game theory's notion that the cybertextual process is not equal to reading a text, the process of navigating through a composed audiovisual space by playing an instrument is not equal to reading a score from a sheet of paper. It demands additional effort on the performer's part, an effort that requires additional choices that in turn lead to a set of consequences. This effort is closer to being active in space than to reading a score. As an activity, it actively contributes to the emergence of one instance of the compositions' multiple possible instantiations and mediates the entanglement of the score and the *gestalt* of the performance.

This entanglement challenges the notion of the dichotomy of text and technology. It allows the symbolic parts of scores to connect firmly with other parts of the work of art. These parts include the performance space, elements of the game world fulfilling a musical task, game-related playing instructions, and nonhuman agents. By continuous adaptions in real time, the entanglement mediated by the feedback from the performers' decisions affects the symbolic as well as the spatial score. Future systematic comparisons of different compositions with real-time scores displaying ergodic and emergent qualities may reveal more aesthetic implications of this new proposed category of scores.

Acknowledgments

GAPPP is funded by the Austrian Science Fund FWF as part of the PEEK program for artistic research, nr. AR364-G24. Thank you Elizabeth Melchor.

3. REFERENCES

[1] D. Kim-Boyle, “Visual Design of Real-Time Screen Scores,” *Organised Sound*, Volume 19, Special Issue 3, pp. 286-294, 2014.

[2] U. Eco, *The Open Work*, Harvard University Press, 1989.

[3] https://nodejs.org/

[4] https://electronjs.org/

[5] R. Gottfried, G. Hajdu, “Drawsocket: A Browser Based System for Networked Score Displa,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation,* Monash University, 2019, pp. 55-60.

[6] G. Hajdu, R. Gottfried, “Networked Music Performance in the Old Elbe Tunnel,” in *International Conference on Technologies for Music Notation and Representation*, Monash University, 2019, pp. 15-25.

[7] https://unity.com/

[8] E. J. Aarseth, *Cybertext - Perspectives on Ergodic Literature*, Johns Hopkins University Press, 1997.

[9] K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*, The MIT Press, 2003.

[10] P. Boulez, D. Noakes and P. Jacobs, “Sonate, Que me Veux-tu?,” *Perspectives of New Music*, Vol. 1, No. 2, pp. 32-44, 1963.

[11] C. Gresser, “Earle Brown’s ‘Creative Ambiguity’ and Ideas of Co-creatorship in Selected Works,” *Contemporary Music Review*, Vol. 26, No. 3/4, pp. 377-394, 2007.

[12] M. Ciciliani, “Virtual 3D Environments as Composition and Performance Spaces,” *Journal of New Music Research*, 49:1, pp. 104-113, 2019.

[13] M. Gardner, "The fantastic combinations of John Conway's new solitaire game 'life'", *Scientific American* 223, pp. 120-123, 1970.

[14] Gotts, Nicholas M. “Emergent Phenomena in Large Sparse Random Arrays of Conway’s ‘Game of Life.’” *International Journal of Systems Science* 31:7, pp. 873–94, 2000.

[15] C. Ressi, *game over. Computerspiel als inter-aktives Medium für audio-visuelle Kunst*, Master Thesis, 2019.

[16] https://vimeo.com/360015209

[17] D. Kim-Boyle, “3D Notations and the Immersive Score,” *Leonardo Music Journal*, Volume 29, pp. 39-41, 2019.

[18] https://supercollider.github.io/

[19] M. Wright; A. Freed, “Open Sound Control: A New Protocol for Communicating with Sound Synthesizers,” in *Proceedings of the International Computer Music Conference*, 1997.

[20] https://www.npmjs.com/package/node-osc

[21] https://www.npmjs.com/package/socket.io

[22] https://www.npmjs.com/package/express

[23] http://snapsvg.io/

[24] S. Bhagwati, “Vexations of ephemerality,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, Coruna, Spain, 2017, pp. 161–166.

[25] J. Freeman, “Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance,” *Computer Music Journal* (MIT Press) 32, no. 3, pp. 25-41, 2008.

COMMON GROUND, MUSIC AND MOVEMENT DIRECTED BY A RASPBERRY PI

Jonathan Bell
Aix Marseille Univ, CNRS, PRISM
Perception, Representations, Image, Sound, Music
Marseille, France
belljonathan50@gmail.com

Aaron Wyatt
Sir Zelman Cowen School of Music
Monash University,
Melbourne, Australia.
Aaron.Wyatt@monash.edu

ABSTRACT

This paper describes *Common Ground*, a piece for six dancing singers and electronics, in which the coordination between performers is ensured by a RaspberryPi-embedded node.js web application. The singers received synchronised scores in the browser of their phone, which they wore in head-mounted display in order to free their hands and enhance their scenic presence. After a description of the artistic project, the elaboration of the score is examined under the categories of movement notation (how trajectories are embedded in musical notation), spectral composition (microtonal tuning between synthesised sounds and human voices), algorithmic processes (how the recent *bell* coding language facilitates processes for which Max patching is ill-suited). The article finally describes the Raspberry implementation, outlining potential ameliorations of the current system, including dns support and unnecessary dependance on a dedicated router.

1. INTRODUCTION

Common Ground, by the Spanish artist Keke Vilabelda, is an immersive installation with large paintings, videos, and three tons of salt covering the ground. Commissioned by the Grau Projekt art gallery in Melbourne [1], it reflects on the common features of landscapes (salt lakes) situated at antipodes of one another (Spain Australia). The initial idea of the musical piece of the same name was to take advantage of this beautiful immersive space, and use it as set design for the performance of six female voices accompanied by electronics [2] (see Fig. 1). The poems chosen for the piece, by the English poet Robert Bell, take the sea as source inspiration - the horizon, natural elements, treated as points of departure for meditation upon everyday life.

From the beginning, the visual aspect of the piece revealed itself to be of primary importance, which is why the performative part had to integrate movements/dance, costumes, and find a way for the score to be part of this ecosystem without disturbing it. Indeed after more than ten pieces written for the Smartvox system, with choirs and ensembles of various sizes, the main issue in concert/performance situation concerns more the theatrical restitution of the piece than the music itself. The system allows singers to move freely on stage and around the audience whilst singing with confidence, which gives the work an interesting immersive feeling. However, the way the system has so far been visually presented needs to improve. Although the fact that singers wear headphones while singing should arguably be the most questionable source of interference between the singer and his audience, it is in fact their visual presence which is the most problematic when the singer has to break eye contact with his/her audience in order to watch the score [3]. Moreover, the presence of the smartphone itself as an object part of the performance seemed most problematic, which encouraged for the search of different solutions.

Figure 1. The *Common Ground* installation by Keke Vilabelda at *Grau Projekt*, Melbourne.

2. HMD

Following *Mit allen Augen*, a piece in which singers and instrumentalists wore head-mounted displays and walked freely around the audience, *Common Ground* carries on with similar concerns, trying to take this idea further by adding a precisely determined choreography. Placed above the head in HMD, smartphones are still rather cumbersome from a theatrical perspective, but SmartVox will probably take advantage in a few years of lighter solutions, such as Vufine glasses (see Fig. 2, left) which proved to be the

[1] https://www.kekevilabelda.com/common-ground
[2] A captation of the performance is available here: https://youtu.be/ZrLgbBw4xfU
[3] See for instance *SmartVox in India:* https://youtu.be/7_FMqLg9vHM

Figure 2. Experimentation with various lowcost Head-Mounted Display (HMD) solutions.

most discreet, allowing for a mirror display of the score (i.e. of the smartphone's screen) in the corner of one of the lenses of the performer's glasses. Although one-eyed, the display is comfortable and wide enough, unfortunately its hdmi connection too often interfered with the audio output of the phone, making it unreliable in a concert/performance situation. Furthermore, its relatively high cost made it inappropriate for large score distribution [4]. Solutions such as QLPP 90 ° FOV AR headset (see Fig. 2, center), evocative of Microsoft Hololens imitations, showed interesting results as they allow for holographic display of the score, but the curvature of their glass requires calibration depending on the performer's phone size and their pupillary distance, again inconvenient for efficiency purposes, because of the often limited time for rehearsals in the performance space.

The solution therefore adopted for *Common Ground* was simply a headset constituted of a double mirror (see Fig. 2, right) for a large and comfortable display slightly above the head of the performer, leaving free the lower field of view, which was appreciated as the performers need to move - sometimes rapidly - in the performance space.

Cat Hope [1] and Christian Klickenberg [2] have both used animated notation in several of their operas, giving evidence that these new forms of notation re-shape the roles traditionally assigned to conductor, singers and instrumentalists : the beating of the time by the conductor is not the main point of reference for singers, instruments, lighting etc... The scrolling of time with a cursor (as, for instance, in the Decibel ScorePlayer) allows for many processes to be automated, giving the music director a different role. Graphic notation also allows for more freedom of interpretation from the perspective of the performer.

However, those solutions still imply the presence of many screens on stage. An interesting field of research in the domain of AR scores [3][4] consists in using Hololens for display of 3D holographic structures (see Fig. 3), an area which soon hopefully interest a larger community of composers and performers.

3. SCORE ELABORATION

3.1 *bach* INScore

The notation in *Common Ground* is conceived as a mixture of singing and movement information. The movement is represented as a graph representing singers are the stage

[4] Performances using this system often require large ensemble, up to 80 performers in *Le temps des nuages* nuages, 12 instruments and 12 voices in *Mit allen Augen*, or 30 singers in SmartVox.

Figure 3. The Elision ensemble performing David Kim-Boyle's new work at TENOR 2019. Photographed by Cat Hope.

below the musical stave, and was controlled by spatial information stored in *bach*, so that the choreography could be written in the musical score directly. The movements are then sent from *bach* to INScore via OSC. INScore [5] is an environment for the design of augmented interactive music scores, opened to unconventional uses of music notation and representation, including realtime symbolic notation capabilities. It can be controlled in real-time using Open Sound Control [OSC] messages as well as using an OSC based scripting language, that allows designing scores in a modular and incremental way. INScore supports extended music scores, combining symbolic notation with arbitrary graphic objects. All the elements of a score (including purely graphical elements) have a temporal dimension (date, duration and tempo) and can be manipulated both in the graphic and time space. They can be synchronized in a master/slave relationship i.e. any object can be placed in the time space of another object, which may be viewed as ”time synchronisation in the graphic space”. As a result, a large number of operations can be performed in the time domain and in particular, moving a cursor on a score is simply achieved using the synchronization mechanism and by moving this cursor in the time space. Time in INScore is both event driven and continuous [6] , which makes it possible to design interactive and dynamic scores. The system is widely open to network uses [5]: it allows to use both local and remote resources (via HTTP), it provides a forwarding mechanism that allows scores to be distributed in real time over a local network. INScore has built-in solutions for monitoring the position and the speed of cursors in an efficient way.

bach supports by default spatial information in its 9th ‘spat’ slot. Inspired by Ircam Spat, this slot allows to store position information (x, y), as well as azimut indicating in which direction the source is facing [5]. Designing a movement of one of the singers therefore consisted in sending coordinate changes or interpolations (x, y and azimut) with a given duration (the length of the note, movements be-

[5] See https://www.youtube.com/watch?v=sTgTv9yqZhI for demonstration.

ing marked in blue, with empty noteheads) [6] . A compositional constraint consisted in avoiding overlapping between dance and singing : the singer never had to move and dance at the same time, rather he/she alternates between singing and moving.

3.2 Spectral Composition - Synthesis

The first pieces composed with SmartVox used mixtures of recorded and synthesised sounds [7] , but most recent works sound synthesis almost exclusively, because it allows for more precise control over sound as well as harmony. The first piece of the cycle [8] mainly consisted of an exploration of basic wave shaping techniques [9] (hence the overall metallic sonority of the piece). *Shir Hassirim* [10] , a piece written around the same time, was more focussed on FM synthesis [11] : through spectral analysis, the goal was to replicate in the choir phenomena like modulation increase. [12]

In the piece *Mit allen Augen* [13] , all the electronics were generated from the PRISM laboratory synthesizer [7], whose perceptive model aims for the emulation of sometimes inconceivable sounds, such as a liquid rain of metal. [14] This material was analysed in *bach* to extract pitch material [15] then ready to use for orchestration (for 12 voices and 12 instruments).

In *Common Ground*, sound synthesis primarily came from the Synthesis Tool Kit [8], accessed through the PerColate library in Max. The physical modelling objects, and brass in particular was of interest as it really captured complex timbral features of brass instruments elsewhere often discussed by Jean-claude Risset and David Wessel [9]. The second source of sound synthesis in *Common Ground* can be described as a subtractive model, in which dense FM spectra are filtered by FFT filters, so as to obtain clearly identifiable pitch content. [16]

The precise overlapping of voices and electronics is made possible in performance thanks to the recent improvement of smartphones' capacities. On the composer's side, a precise dialogue between electronics and the score is highly facilitated by *Max for Live* [17] .

3.3 Audioscore - Display

Audio scores have been a key concept for the research undertaken by one of the authors [10]. Since a survey by Bhagwati on this topic [11], audio scores seem to enjoy increasing popularity [12]. Our claim is that, when applied to vocal ensemble writing in particular, they allow

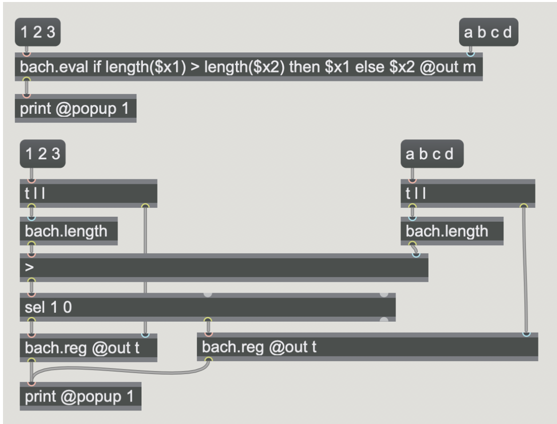


Figure 4. A comparison between an llll manipulation process described through a snippet of bell code (in the bach.eval object box, above) and the corresponding implementation within the standard graphical dataflow paradigm of Max (below).

unprecedented accuracy in the realm of spectral composition, when the singers need to match the harmonic content of the tape accurately.

Working with various ensembles and receiving each time important feedback has confirmed that visual and auditory information need to be anticipated as much as possible : pauses in a singer's separate part always provide aurally what comes next. Also visually, *Common Ground* adopted a 2-systems-per-page mechanism in which the system that is not playing always anticipates what comes next. [18]

3.4 Aspects of the bell language

3.4.1 Introduction: Vocaloid

Common Ground also marks an evolution in the way vocal generation is used in SmartVox, which hitherto consisted in storing path to samples of vocal speech inside each note or each melisma through the slot storing system in *bach*. [19] This method however presented significant drawbacks : although it made the end result slightly more expressive or convincing, it was extremely time consuming to make phoneme change correspond to note change whilst designing each vocal line. Also reading samples direct to disk often introduced delay which made the result imprecise.

The new method consists in using a vocal synthesizer (AlterEgo Plogue, a free equivalent of the japanese Vocaloid synthesizer[13][14]), using *bach*'s slot storing system to control the synthetiser via midi. This slightly more robot-sounding solution has the great advantage to be far more malleable algorithmically, since a new note can trigger a phoneme change. With the discovery of the new *bell* language [15] in *bach* [16], the method opened the door to promising experiments in the realm of algorithmic composition.

[6] See the part of Soprano 1: https://youtu.be/FcF4oNxJweg
[7] See *SmartVox* for instance: https://youtu.be/JZsJn7EEW-A
[8] See *In Memoriam J.C. Risset*: https://youtu.be/hQtyu1dcCaI
[9] See for demonstration: https://youtu.be/v-LlClEnxf0
[10] See *Shir Hassirim*: https://youtu.be/7GpArQa6mQ4
[11] See for demonstration: https://youtu.be/D6mCgx4pSxs
[12] As in the following extract: https://youtu.be/7GpArQa6mQ4?t=26
[13] See *Mit allen Augen*: https://youtu.be/ET_OBgFWx04
[14] See for demonstration: https://youtu.be/2kdIaqAhUGs
[15] See for demonstration: https://youtu.be/gZKONcOOhaE
[16] See for demonstration: https://youtu.be/zN95OkWSDHY
[17] Bach's playback notification can be redirected to ableton via M4L for precise synchronisation: see https://youtu.be/VJvY5wYl_cM
[18] See for instance the alto part: https://youtu.be/yWD6u2cPvSc. Slot No4 (the yellow one **here**) is the one that stores the sample's path.
[19] See for demonstration: https://youtu.be/s4qS2khwkT0

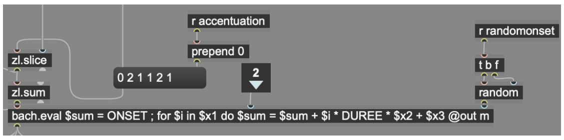


Figure 5. A loop calculates the onset of each syllable of a vocal line according to a starting onset (the variable "ONSET"), a given duration/tempo ("DUREE"), and prosodic accentuation (2 1 1 2 for long short short long).

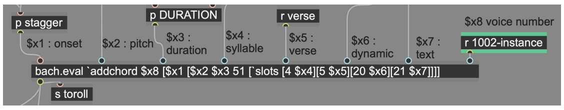


Figure 6. The *bell* language is mainly exposed in Max through the *bach.eval* object. $x1, $x2… correspond to the different inlets of the object. *bach.eval* makes the construction of lisp-inherited parenthesis structures much easier than with the data-flow *bach.wrap* system.

3.4.2 Bach Evaluation Language for llll - an overview of the bell language

The *bell* language in *bach* arose from an observation that the Max patching environment can be cumbersome when formalising algorithmic compositional processes: "It has been clear since the beginning of *bach* that non trivial tasks - in algorithmic composition - require the implementation of potentially complex algorithms and processes, something that the graphical, data-flow programming paradigm of Max [...], is notoriously not well-suited to."[15]. Single line snippets of code in bell often require many objects and cables in Max. The data structure in *bach* is called *llll* (lisp-like linked lists), so, within this language, objects or functions are specifically designed to operate on lists. Fig.4shows two different implementations of the same algorithm, which will output the longest list (in this case: 'a b c d') when the left input is triggered: the first (top) version is evidently easier to read that the one below.

Indeed, while the Max GUI can be extremely intuitive and efficient for many DSP processes, its data-flow paradigm can make message formatting efficient in Max (and hence in *bach*). As exemplified in Fig. 5, bach.eval allows for a single line of code to centralize message formatting, which would have formerly required dozens of objects, themselves most often bringing order or priority issues.

The implementation of variables in the *bell* language constitutes another major improvement of *bach*. The ability to name variables in Max (such as ONSET, or DUREE, as in the loop expressed in Fig. 5) and assign them a value helps again centralising information within simple equations, which the message-driven send-receive Max functionality would have made more prompt to error.

3.4.3 Algorythmic composition with bell

Although in germ in *Common Ground*, a more systematic approach to algorithmic polyphony generation was used in

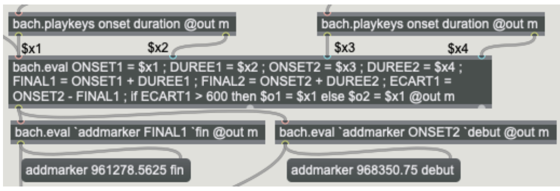


Figure 7. The following script adds markers only when two notes are separated by more than 600ms.

a *Deliciae* [20] , a piece composed just after *Common Ground*, while discovering the new *bell* language (see Fig. 6) [15] [17].

The polyphony of European tradition obeyed extremely strict rules throughout Europe during the Renaissance. Many of those rules discussed in the treatises of the time served as source of inspiration for polyphony generation with the tools exposed above. The first obvious parallel consists in treating each voice as equal, unlike for instance when writing for an instrumental ensemble of a modern orchestra. This is why most polyphonic passages in *Common Ground* and *Deliciae* were generated inside a poly~ in Max, with each instance (i.e. each voice) receiving the same information regarding text, prosody, and harmonic material, but only differing by vocal range (sopranos for instance cannot sing below middle C and so forth).

Contrast in Renaissance polyphony often consist in alternation between homophonic passages and contrapuntal ones, which inspired most parameters available to tweak for a given verse: when the variables RANDUR, RANDOMONSET, DECAL, and STAGGER are set to 0, the algorithm will generate a homophony [21] (singers articulate and move from one pitch to the next at the same time). If only RANDUR increases, voices will start at the same time, but their duration will differ between each other. If only RANDOMONSET increases, they will all have the same duration but start at different times. If only DECAL increase, voice will enter at regular intervals from the bottom-up (and inversely if DECAL is negative). STAGGER, finally, imitates a behaviour typical of the renaissance where two groups of voices are staggered or delayed by a given value.

3.4.4 Automatic cueing system

Since the beginning of SmartVox (see [18], Fig. 4), cueing the singers with what comes next appeared one of the main advantages of the system.

To identify appropriate moments for page turns and cueing the singers accordingly, the first step consisted in identifying the start and end of each phrase (see Fig. 7): with iterations on each note of the score two by two, we evaluate if the distance between two notes is superior to 600 ms:

[20] A video of the performance is available at: https://youtu.be/zxnznD0Gzo0

[21] See parameters tweaks on the right hand side for demonstration here: https://youtu.be/OKkiySEagm0. ONSET is in milliseconds and correspond to the position in the timeline where the generation is happening : as exemplified in the video 747618 ms correspond to 12'28". The term DUREE (French for duration) represents the duration of notes : the tempo speeds up when durations diminished

in the first case it isn't (see Fig. 8, the two notes are close to one another) and nothing happens. On the following iteration however, the gap between two notes is wider than 600ms (see Fig. 9), so the messages "addmarker fin" and "addmarker debut" are sent to the end of the phrase and to the beginning of the next phrase respectively.

When a performer has nothing to sing, this precious time is systematically used in the score to provide cues feeding the perfomer's headphone with what is coming next: using the markers previously generated to retrieve their onsets, if the pause is longer than the phrase to sing, (i.e. if the DURANTICIP is greater than DUR (see Fig. 10, and the "then" stance of the "if" statement in the code below), then the cue will need to start at the onset corresponding to the difference between entrance of the singer (START) and the end of his phrase (END), with a 300ms break between the two. If on the other hand, the pause is shorter than the phrase to sing (see Fig. 11, and the "else" stance of the if statement below), then the cue needs to start as soon as possible, i.e. as soon as the singers has finished to previous phrase (PREV):

```
START = $x2:($x1 1) ;
END = $x2:($x1+1 1) ;
PREV = $x2:($x1−1 1);
DUR = END − START ;
DURANTICIP = START − PREV ;
if DURANTICIP > DUR then ‘;
‘tocue ‘paste 2∗ START − (END + 300) 2
else ‘; ‘tocue ‘paste (PREV + 60) 2
```

Finally, onset information from the 'end' markers (the ones named 'fin', as in Fig. 8 at 0'16"200"') are used for display information : the domain to be displayed on the playing staff and on the preview staff (i.e. the staff-line that is coming next, as for page turns) of the *bach.roll*.

```
‘addmarker $x2:$x1
[‘play [$x2:$x1 ($x2:($x1+1)+ 200)]
‘preview [$x2:($x1+1) ($x2:($x1+2)+ 200)]]
```

Each time the cursor hits one of these markers, the domain display of both 'playing' and 'preview' staves are updated, provoking at the same time an alternation up and down between the position of those staves, so that the passive (or 'preview') roll looks like an anticipation of the active (or 'playing') one, resulting on a 2-staves display with constant preview. [22]

[22] See for instance the tenor part: https://youtu.be/NLpI_OpFcTs

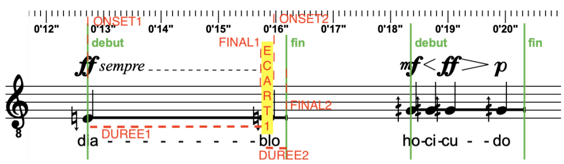


Figure 8. The first note (with lyrics "dia") has a duration that lasts until the beginning of the following note, (with lyrics "blo"). The distance between the two (ECART1, highlighted in yellow) is almost null.

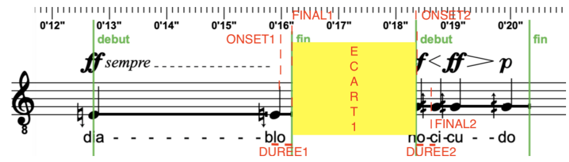


Figure 9. The two notes (with lyrics "blo" and "ho" respectively) are separated by a silence longer than 600 ms (ECART1 lasts a bit more than two seconds), therefore two markers are generated.

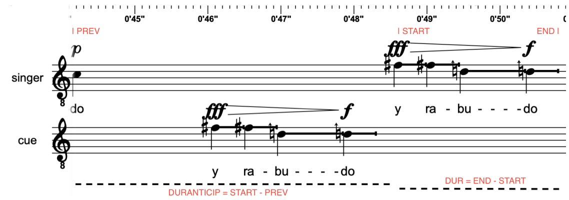


Figure 10. When the pause is long (or very long....) the cue needs to be provided as late as possible i.e. just before the singer's entrance. The corresponding onset value is 0'46" because START*2 - END = 48,5*2 - 51 = 46

4. A RASPBERRY PI HARDWARE EMBEDDED SYSTEM SOLUTION FOR LOCAL NMPS

In search of a light plug-and-play dedicated system to be sent over the post, the Raspberry Pi quickly appeared as the best option to host SmartVox on an embedded system. Node.js runs on Raspbian, and SmartVox proved to be very stable on a Raspberry Pi 3, so, once installed, the only two steps for a *0-conf* deliverable hardware were:

- Setting up a static address for a dedicated router (e.g. tp-link...).
- Starting SmartVox at boot using linux 'systemd' service.

Starting a script at boot can be done on Raspbian with a file containing the following in the etc/systemd/system:

```
[Unit]
Description=My service
[Service]
ExecStart=/home/pi/Desktop/hello.sh
[Install]
WantedBy=multi−user.target
```

With the hello.sh script containing the following to launch the server:

```
#!/bin/bash
cd /home/pi/Desktop/risset
npm run start
exec bash
```

This low-cost system now allows the sending of ready-to-use scores. Once the system is power-supplied, all the performers need to do is to join the dedicated Wi-Fi, and type the static IP address of the server on their smartphone/tablet (i.e. for the performers: 192.168.0.100:8000, and for the conductor: 192.168.0.100:8000/conductor). In January 2019,

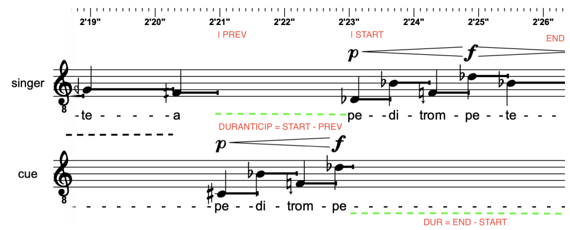


Figure 11. When the pause is short, the cue needs to be provided as soon as possible i.e. just after the previous singer's phrase (see the PREV variable).

the system was rented to the Caen French conservatoire via BabelScores, [23] thus proposing a rental of performing scores (separate parts) of a new kind.

To make configuration even easier in the future, a lightweight DNS server, like dnsmasq, could be installed and configured on the Raspberry Pi to allow performers to enter a friendlier, more human readable address to access the nodejs server. Additionally, new ways of updating the scores on the device could be explored to both simplify the process and to limit the amount of data that needs to be sent out to it. Currently, the most straight-forward way to update the device is to send out a new disk image to be written to the SD card at the other end. Piping the image through xz substantially reduces the image file size, particularly if the filesystem on the card contains a large amount of free space. This can be done using the following terminal commands to first create the image on one end and to then write it on the other:

```
sudo dd if=/dev/disk3 bs=4m | xz > common2.iso.xz
xzcat common2.iso.xz | sudo dd of=/dev/disk3 bs=4m
```

(where /dev/disk3 is replaced by the SD card device name).

To make this process easier for the end user, it could also be possible to have a web server on the device configured to accept the upload of update packages. These would then only need to contain newer versions of resource files (like the videos used for the scores), so that the entire system doesn't need to be refreshed for minor changes.

5. POSSIBLE AVENUES FOR A FUTURE EXTENSION OF THE WORK

The *bell* language [24] offers promissing perspectives in the realm of algorithmic composition. The linkage with the *ConTimbre* library in particular offers promising results in the instrumental domain, with at the same time an intuitive control over the generated result as well as a possibly more speculative approach related to machine learning [25]. A future extension of *Common Ground* with therefore include instruments as well as voices.

[23] Babelscores (https://www.babelscores.com/) currently supports actively supporting the SmartVox project: http://1uh2.mj.am/nl2/1uh2/lgi4u.html. The first piece performed in Caen with the Babelbox is available at the following address : https://youtu.be/wUyw0KQa5Wo

[24] To clarify, Bell is author of the article, albeit mere user of the coding language of the same name.

[25] See the following for demonstration: https://youtu.be/ByeIyRLnXw

6. CONCLUSIONS

This article presents an overview of the evolution of the SmartVox project, with an emphasis on the artistic project *Common Ground*, its more systematic use of algorithmic processes for composition thanks to the *bell* language in *bach*, as well as the Raspberry Pi implementation of the piece/server.

Acknowledgments

We would like to thank Cat Hope for having made this work possible in Melbourne.

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*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 193–200.

[2] C. Klinkenberg, "A combination of graphic notation and microtonal pitch notation in the video score of the opera "the cross of the engaged"," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 186–192.

[3] D. Kim-Boyle, "3d notations and the immersive score," *Leonardo Music Journal*, vol. 29, pp. 39–41, 2019.

[4] D. Kim-Boyle and B. Carey, "Immersive scores on the hololens," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*, C. Hope, L. Vickery, and N. Grant, Eds. Melbourne, Australia: Monash University, 2019, pp. 1–6.

[5] D. Fober, Y. Orlarey, and S. Letz, "Towards dynamic and animated music notation using inscore," in *Proceedings of the Linux Audio Conference — LAC 2017*, V. Ciciliato, Y. Orlarey, and L. Pottier, Eds. Saint Etienne: CIEREC, 2017, pp. 43–51. [Online]. Available: inscore-lac2017-final.pdf

[6] ——, "Inscore time model," in *Proceedings of the International Computer Music Conference*, 2017, pp. 64–68. [Online]. Available: inscore-icmc17-final.pdf

[7] M. Aramaki, C. Gondre, R. Kronland-Martinet, T. Voinier, and S. Ystad, "Imagine the sounds: anintuitive control of an impact sound synthesizer," in *Auditory display*, ser. Lecture Notes in Computer Sciences, Ystad, Aramaki, Kronland-Martinet, and Jensen, Eds. Springer Berlin / Heidelberg, 2010, pp. 408–422. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00462274

[8] P. Cook and G. Scavone, "The synthesis toolkit (stk)," 08 2000.

[9] J.-C. Risset and D. L. Wessel, "5 - exploration of timbre by analysis and synthesis," in *The Psychology of Music (Second Edition)*, second edition ed., ser. Cognition and Perception, D. Deutsch, Ed. San Diego: Academic Press, 1999, pp. 113 – 169. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780122135644500068

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

[11] S. Bhagwati, "Elaborate audio scores: Concepts, affordances and tools," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 24–32.

[12] C. Sdraulig and C. Lortie, "Recent audio scores: Affordances and limitations," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*, C. Hope, L. Vickery, and N. Grant, Eds. Melbourne, Australia: Monash University, 2019, pp. 38–45.

[13] H. Kenmochi, "Vocaloid and hatsune miku phenomenon in japan," in *InterSinging*, 2010.

[14] K. Y. Lam, "The hatsune miku phenomenon: More than a virtual j-pop diva," *The Journal of Popular Culture*, vol. 49, no. 5, pp. 1107–1124, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jpcu.12455

[15] J.-L. Giavitto and A. Agostini, "Bell, a textual language for the bach library," in *ICMC 2019 - International Computer Music Conference*, New York, United States, Jun. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02348176

[16] A. Agostini and D. Ghisi, "Bach: an environment for computer-aided composition in max," in *Proceedings of the 38th International Computer Music Conference (ICMC)*, Ljubljana, Slovenia, 2012.

[17] A. Agostini, D. Ghisi, and J.-L. Giavitto, "Programming in style with bach," in *14th International Symposium on Computer Music Multidisciplinary Research*. Marseille, France: Mitsuko Aramaki, Richard Kronland-Martinet , Sølvi Ystad, Oct. 2019.

[18] J. Bell and B. Matuszewski, "SMARTVOX - A Web-Based Distributed Media Player as Notation Tool For Choral Practices," in *TENOR 2017*, Coruña, Spain, May 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01660184

RATIOSCORE: A TEXT-BASED SYSTEM FOR JUST INTONATION

Julie Zhu
Stanford University
julie@ccrma.stanford.edu

Craig Stuart Sapp
CCARH/PHI, Stanford University
craig@ccrma.stanford.edu

ABSTRACT

Ratioscore is a text-based musical score representation system that expresses pitches as integer ratios, making it particularly useful for working with just intonation. We also provide open-source software to compile Ratioscores into Standard MIDI files as well as a website front-end that renders these MIDI files into MP3 files for online listening and downloading. Up to 15 voices/instruments can sound simultaneously with independent tuning control as well as independent glissandi in each voice within one octave of their starting pitch. Ratioscores are encoded using the Humdrum data format, which allows for basic data manipulation and transformation of the score, such as repeating segments of music, selecting between multiple timelines or sharing dynamics between different voices. We illustrate the use of Ratioscore as a prototyping system for composing a string quartet as a proof of concept. Ratioscore is particularly suited for algorithmic composition, both within and outside the traditional Western notation framework.

1. INTRODUCTION

The overlap between just intonation (JI) and the beauty of mathematics is substantial. Young composers inclined towards a kind of numerical perfection are increasingly drawn into the just intonation "cult." In addition to more pieces composed with just intonation, theory and resources grow alongside to help composers navigate the just world.

Before Marc Sabat and Wolfgang Von Schweinitz developed accidental-based notation, the Extended Helmholtz-Ellis Just Intonation (HEJI) pitch notation, there were numerous ways to notate just pitches. A plus or minus could be added to a note to indicate a rough deviation [1], or a cent differential from the equal tempered pitches as described in [2], or simply the JI ratio with a specified fundamental. In 2004, Plainsound Music Edition published Sabat and Von Schweinitz's catalogue of accidental groupings that indicate prime-groupings and thus, families of cent deviations from the equal temperament, which has become standard for JI notation [3].

JI playback has not been standardized. Max and SuperCollider, along with Thomas Nicholson's web application can calculate JI harmonic spaces, but that's only to collect individual frequencies, preparatory abstractions [4]. Without bespoke programming, timelines of JI music are not easily produced. In the last two years, Dorico, a music notation software, in collaboration with Plainsound composers, developed a microtonal playback system that Sabat endorses [5]. He describes what many composers have resorted to doing before Dorico: making the scores look good in notation software, but accidentals are "simply visualizations: text attachments without actual functionality." In other words, without a lot of manual adjustment, the notated music could not be played back accurately. Perhaps with the improvement of vibrato-less soundfonts (VSTs) such as NotePerformer, Dorico will become the go-to software for JI composers that are working with staff notation [6].

We propose Ratioscores as an additional resource, open-source and text-based, which offer easy access to tuning systems outside of traditional Western notation and equal-temperament for use with any General MIDI synthesizer. As with just-intonation tuning, pitches are given as fractions, which are ratios between the target pitch and a constant fundamental, which is notated by '1/1' or '1.'

2. RATIOSCORE REPRESENTATION

Ratioscores are based on the Humdrum file format, which is a generalized two-dimensional digital score representation organized similarly to common Western music notation (CWMN) scores without system breaks that are rotated 90° clockwise so that time progresses downwards in the file.[7] Each row represents events that occur simultaneously, and columns ('spines' in Humdrum terminology) represent different data streams running in parallel, such as time, pitch and dynamics. Humdrum data can represent music in CWMN [1] but also allows creation of non-standard sequential digital descriptions of music, such as Ratioscores. Figure 1 gives an example of a simple Ratioscore containing a timeline and a single voice. [2] The same musical content is shown below the Ratioscore in CWMN using HEJI accidentals.

The first spine is headed with the text '**time' and represents a timeline controlling the starting time and duration of notes found in '**ratio' spines(s). For example, the second note (D5) occurs one second after the start of the music, and sustains for one second until the next note at two seconds from the start of the score. All data spines should end with the end-of-data marker '*-' and each spine must be separated from each other by one or more tab characters

[1] https://verovio.humdrum.org

[2] Audio versions of figures are available at https://ratioscore.humdrum.org/tenor2021/

**time	**ratio
*	*Ivioln
*	*ref:G3
0	1
1	3
2	11/4
2.25	5/2
2.5	0
3	7
3.5	4/3
4	0
*-	*-

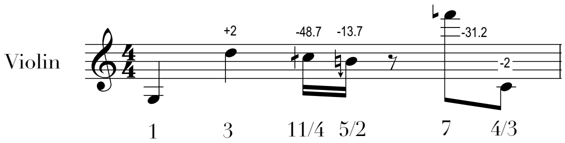


Figure 1. Sample music in Ratioscore and HEJI notation.

in a TSV (tab-separated-values) arrangement. Additional structural information about Ratioscores is available at the website, such as how to add metadata and comments to a score. [3]

2.1 Ratio spines

Each ratio spine represents a monophonic voice, with a new pitch ratio automatically turning off the previous note in the voice, or one can use '0' to silence a note without sounding a new one. Ratioscores are intended to be compiled into Standard MIDI files, so the number of ratio spines is limited to 15 since there are 16 MIDI channels available, minus channel 10 which is dedicated to percussion. Each ratio spine is mapped onto a separate MIDI channel, allowing independent tuning of notes in each voice with pitch-bend messages. Instruments can be specified, such as violin with the text '*Ivioln', or '*I#40' as a General MIDI instrument number. [4]

Pitches are indicated by integer ratios, either a single integer representing a harmonic or two integers separated by a slash. For example, '11/4' in Figure 1 is the 11th harmonic transposed down two octaves, 5/2 is the 5th harmonic transposed down an octave, and 4/3 is a just perfect fourth. Ratios can be adjusted by cent intervals, such as '3/2-1.955c', which is a just perfect fifth lowered by 1.955 cents to create an equal tempered (ET) fifth. Cents can also be interchanged with ratios for convenience, such as '701.955c' for a just perfect fifth.

The reference pitch for notes in a ratio spine is demonstrated in Figure 1 by '*ref:G3', where G3 is the G below middle C (assuming an equal temperament based on A-440), and all pitches for ratios are calculated from the reference pitch. If no reference pitch is given, then 'C4' (middle C) will be used, and the reference may be changed in a ratio spine at any time by indicating a new reference pitch. The reference pitch may also be tuned by adjusting a cent interval such as '*ref:G#3-12c', which means that the reference pitch is 12 cents flat of ET G#3. Reference pitches may also be described in Hertz by adding the letter 'z' after the frequency such as '*ref:261.63z' for 261.63 Hertz, which is equivalent to '*ref:C4'.

2.2 Exponent notation

Figure 2 demonstrates an alternate method for describing ratios with mathematical expressions. The clarinet part in this figure uses exponents to express stacks of just perfect fifths, while the violin part uses the same intervals multiplied out into a simplified rational number. The two parts play the same pitches, but the Pythagorean tuning is conveyed more clearly in the clarinet part. Currently only exponentiation is implemented, but other expressions to describe factorizations and roots may be added in the future.

**time	**ratio	**ratio
*	*Ivioln	*Iclars
*	*ref:G3	*ref:G3
0	1	(3/2)^0
0.5	3/2	(3/2)^1
1	9/4	(3/2)^2
1.5	27/8	(3/2)^3
2	81/16	(3/2)^4
2.5	243/32	(3/2)^5
3	729/64	(3/2)^6
3.5	2187/128	(3/2)^7
4	0	0
*-	*-	*-

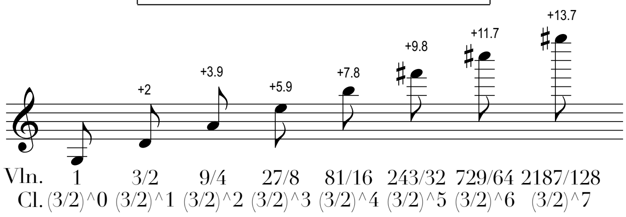


Figure 2. Ratioscore with rational number and equivalent exponential forms playing successive Pythagorean fifths.

2.3 Part alignment and barlines

The rhythm of all ratio spines is controlled by a single timeline spine, typically given in the first column of the score. To continue sustaining a note through an entry in the timeline, add a period/full-stop character ('.') in the sustaining note's column. Figure 3 has examples of these spacers where some parts are sustaining notes while others are attacking new ones. The figure also demonstrates addition of barlines in the score. These are indicated by equal signs ('=') in each column followed by an optional measure/bar number. Barlines do not need to be evenly spaced in time and can split sustained notes, but if a Ratioscore is intended to represent CWMN, then time signatures such as '*M6/8' can be added to a ratio spine, and barlines can be placed to match the meter (Figure 3 includes a pickup beat).

2.4 Alternate timing methods

The controlling timeline in a Ratioscore can be expressed in several ways for convenience. The '**time' spine illus-

[3] https://ratioscore.humdrum.org

[4] See https://ratioscore.humdrum.org/doc/instruments for a list of other instrument codes and numbers.

**time	**ratio	**ratio	**ratio	**ratio
*MM60	*Ivioln	*Ivioln	*Iviola	*Icello
*M4/4	*ref:G3+2c	*ref:D4+4c	*ref:C3	*ref:C2
0	1	.	.	6/5
0.5	.	1	6/5	.
=1	=1	=1	=1	=1
1	3	6/5	1	4/3
1.5	6/5	3	.	1
2	5	4/3	3	3/2
2.5	4/3	.	3/2	3
3	7	3/2	5	9/8
3.5	.	7	9/8	5
4	9	.	7	7/5
4.5	9/8	9	.	7
=2	=2	=2	=2	=2
5	11	7/5	9	9/7
5.5	7/5	11	9/7	.
6	13	9/7	.	13/11
6.5	9/7	13	13/11	11
7	15	13/11	13	.
7.5	13/11	15	.	13
=3	=3	=3	=3	=3
9	.	.	.	.
*_	*_	*_	*_	*_

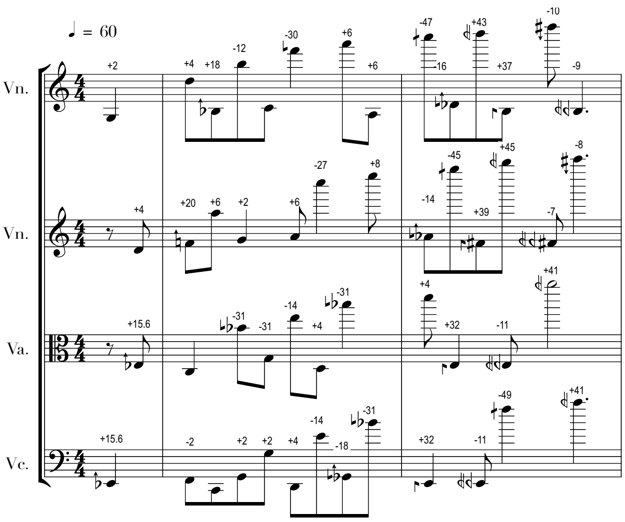


Figure 3. Ratioscore with rational numbers and equivalent exponential forms.

trated in Figure 1 expresses time in absolute seconds from the start of the music. Additional descriptions of time are given in this section.[5] In all cases, tempo adjustments can be made by adding a tempo indication in the form '*MM#' where '#' is a floating-point number giving the 'beat' count (seconds) per minute. The default tempo in timeline types is 60 beats per minute (bpm). Multiple timelines can be stored in a Ratioscore, but only the leftmost one will be used.

All timelines other than '**recip' can use floating-point time values as well a rational numbers and mixed fractions, so '1.5', '3/2', and '1+1/2' are all equivalent time values. Describing times with fractions allows tuplets to be easier to read from the scores, while floating-point values are useful for describing physical times. The '**recip' system is a fractional delta-time system.

[5] More details can be found at https://ratioscore.humdrum.org/doc/timeline.

2.4.1 *Delta timings*

Figure 4 demonstrates 'delta times' being used to represent the same timing information given as absolute timings in Figure 3. With delta times, time values are given in terms of the duration from the current line to the next. Expressing time in this format is similar to timings in MIDI files, and it allows for copy-and-pasting lines of the score to repeat musical content. In Figure 4, notice that the tempo is set to 120 bpm, which causes the delta times of '1' to be equivalent to 0.5 seconds.

**dtime	**ratio	**ratio	**ratio	**ratio
*MM120	*Ivioln	*Ivioln	*Iviola	*Icello
*M4/4	*ref:G3+2c	*ref:D4+4c	*ref:C3	*ref:C2
1	1	.	.	6/5
1	.	1	6/5	.
=1	=1	=1	=1	=1
1	3	6/5	1	4/3
1	6/5	3	.	1
1	5	4/3	3	3/2
1	4/3	.	3/2	3
1	7	3/2	5	9/8
1	.	7	9/8	5
1	9	.	7	7/5
1	9/8	9	.	7
=2	=2	=2	=2	=2
1	11	7/5	9	9/7
1	7/5	11	9/7	.
1	13	9/7	.	13/11
1	9/7	13	13/11	11
1	15	13/11	13	.
3	13/11	15	.	13
=3	=3	=3	=3	=3
*_	*_	*_	*_	*_

Figure 4. Ratioscore with delta timing. Each line except the last one has a duration of 0.5 seconds.

2.4.2 *Reciprocal timings*

Ratioscore also allows reciprocal timings, which describe time in terms of divisions of a whole note. Numbers are expressed in terms of the reciprocal of the divisions, such as '4' for a quarter note since it divides the whole note into four equal parts. Augmentation dots are represented by adding a dot after the rhythmic value, such as '8.' for a dotted eighth note. Tuplets function in the same manner, with '3' meaning a triplet half note, and '20' being a quintuplet sixteenth note. More complex ratios that are not integer divisions of the whole note are represented by placing a percent sign between the two fractional parts of the number, such as '3%2' which represents a triplet whole note since a triplet whole is 2/3rds of a regular whole note). Figure 5 gives an example of this timing system using the '**recip' data type for the time column.

Tempo markings in *recip* spines describe the rate of quarter notes ('4'), while in other timelines the tempo marking affects the duration of a second.

2.4.3 *Millsecond timings*

The data types '**ms' and '**dms' are similar to '**time' and '**dtime' respectively, but integers represent milliseconds rather than seconds.

**recip	**ratio
*M60	*Ivioln
*	*ref:G3
4	1
4	3
16	11/4
16	5/2
*MM44	*
8	0
8	7
8	4/3
*-	*-

Figure 5. Time described using musical rhythms (4:quarter note, 8:eighth note, 16:sixteenth note).

2.5 Glissandi

Glissandi are represented in Ratioscores by the letter 'H' attached to the staring note of the glissando and 'h' on the ending note as illustrated in Figure 6. Glissandi may extend up to one octave above/below the starting note, but no further due to the pitch-bend depth limits in MIDI. The Ratioscore conversion software analyzes the ranges of glissandi in each ratio spine and sets the necessary pitch-bend depth.

Intermediate notes within a glissando can be either sustained or attacked. Adding an underscore character will prevent a note from re-attacking during a glissando. The underscore character can also function as a tied note if the previous pitch matches the same note coming after it with an underscore marker. This is useful for encoding tied notes when the Ratioscore more closely represent a CWMN score.

The default update rate for pitch bends within a glissando is 50 milliseconds. For fast glissandi with large intervals, this may cause the discrete pitch-bend adjustments to become audible. In such cases the pitch-bend update rate can be manual adjusted, such as setting it to 10 milliseconds with the text '*grate:10' (meaning 'glissando rate'). [6]

2.6 Dynamics

Dynamics can be controlled through attack velocity and volume continuous controllers in MIDI files. Ratioscores allow separate control of these two types of loudness with '**vel' and '**vol' spines to the right of ratio spines that they apply to as illustrated in Figure 7. [7] If a dynamics spine is placed to the left of all ratio spines, then it will apply to all ratio spines unless a ratio spine has its own dynamics spine.

Dynamics can either be MIDI-like numbers from 1 though 127, or they can be symbolic musical values such as 'mf' or 'pp'. Symbolic values can be assigned specific MIDI numeric values as illustrated in the score, where '*pp:10' means to convert 'pp' to the MIDI attack velocity 10. The 'p' and 'f' dynamics will use the default mappings since no explicit mapping are given for them in this example. Crescendi are the pair of characters '< [' which are placed at the starting/ending times of the crescendi, and '>]' are used for decrescendi. The ending bracket character can be omitted if a crescendo ends at a dynamic mark.

[6] See https://ratioscore.humdrum.org/doc/glissandos for more information.

[7] See https://ratioscore.humdrum.org/doc/dynamics.

**time	**ratio	**ratio
*	*Ivioln	*Iclars
*	*	*grate:10
*	*ref:C4	*ref:C4
0	1H	0
2	_3/2	3/2
4	_1h	_3/2H
4.5	0	.
5	.	_1
6	.	.
6.5	3/2	_6/5
7	1H	.
8	_5/4h	5/4
9	.	_1h
10	0	1H
11	3H	_2h
12	_2h	.
13	0	0
*-	*-	*-

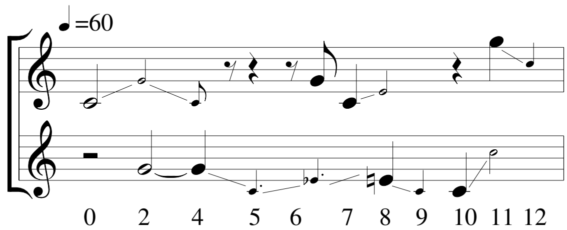


Figure 6. Sample score with glissandi

3. MIDI TUNING AND TIME QUANTIZATION

We provide software to convert Ratioscores into the Standard MIDI File format to realize just intonation. [8] Typically the temperament of notes is not specified in MIDI data but rather is set on the synthesizer for playback of MIDI files.[8] There is no standard way of setting the temperament in a MIDI file, but synthesizer-specific temperaments can be encoded as system-exclusive messages.[9] However, these messages will only be understood by a particular synthesizer and will be ignored by others. We wanted an easy way to use any synthesizer, so note tuning

[8] https://github.com/craigsapp/ratioscore

*time	**ratio	**vel
*	*Ivioln	*pp:10
*	*ref:G3	*
0	1	p
1	3	<
2	11/4	.
2.25	5/2	.
2.5	0	f >
3	7	.
3.5	4/3	.
4	0	pp
*-	*-	*-

Figure 7. Sample score with dynamics

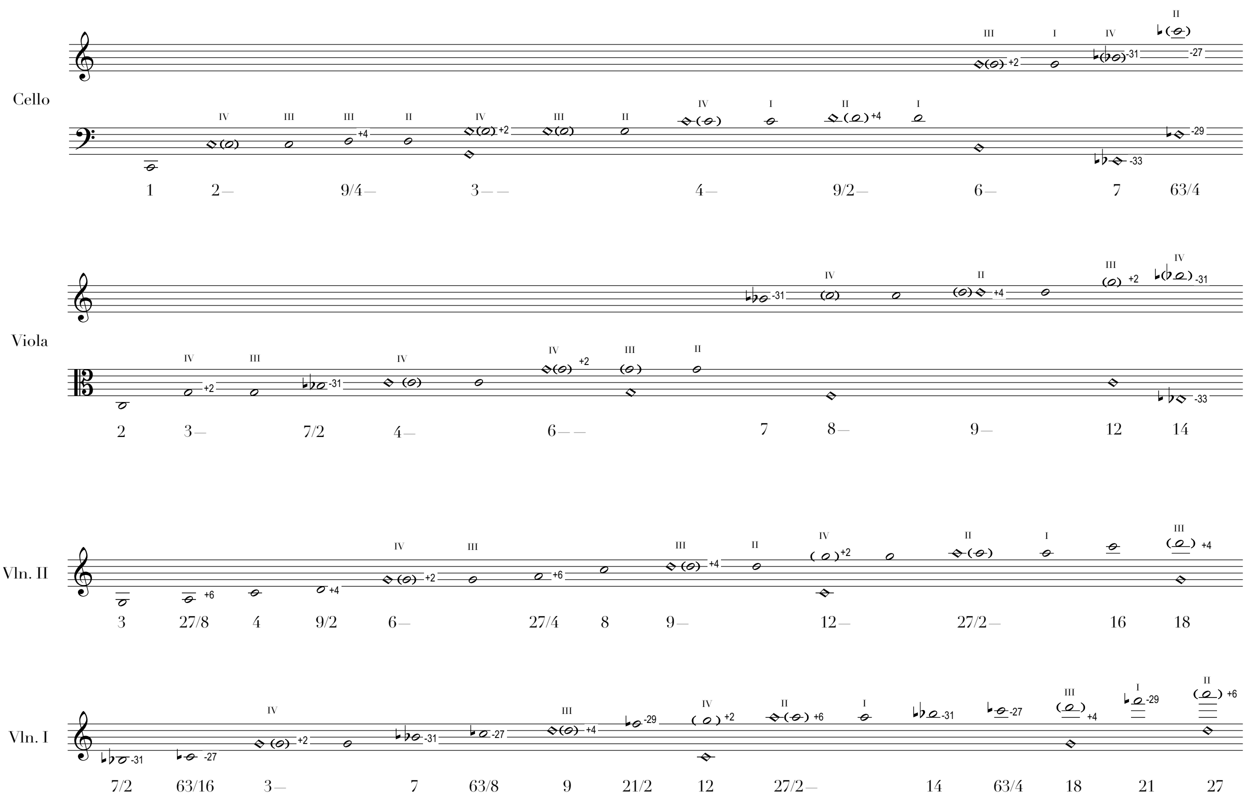


Figure 8. Pitch space for *The Answers* based on the fundamental C2, the lowest note of the cello. Sixteen notes (not necessarily unique pitch, or ratio) for each viol.

**time	**ratio	**ratio	**ratio	**ratio
*	*Icello	*Iviola	*Ivioln	*Ivioln
*	*ref:C2	*ref:C2	*ref:C2	*ref:C2
=1	=1	=1	=1	=1
4.231	.	.	.	27/8
6.182	2	.	.	.
7.441	.	.	.	3
=2	=2	=2	=2	=2
9.337	.	3	.	.
9.502	.	.	636	.
9.521	.	.	7/2	.
11.522	.	.	.	6
=3	=3	=3	=3	=3
13.571	.	.	7	.
13.958	.	.	.	9
14.814	.	.	.	6
15.358	.	.	12	.
15.457	1	.	.	.
15.567	.	.	.	27/2
=4	=4	=4	=4	=4
16.487	.	.	63/4	.
17.037	.	.	.	9
17.459	.	.	63/8	.
17.776	.	.	.	27/2
18.162	.	.	27/2	.
18.476	.	.	18	.
19.854	.	.	.	16
=5	=5	=5	=5	=5
20.038	.	2	.	.
20.223	.	.	21	.
20.594	.	.	.	12
20.837	.	.	27/2	.
22.064	.	.	.	18
22.103	.	.	27	.
22.11	.	.	9	.
22.816	.	.	.	27/4
23.334	.	.	14	.
23.673	.	.	.	12

Figure 9. First page of final performance score after quantization of the MIDI file compiled from by Ratioscore, along with original Ratioscore representation of the same music. Lines starting with an equal sign (=) represent barlines in the notation.

is instead controlled by pitch-bend messages.

A limitation of using pitch-bend to control tuning is that all notes on a particular channel share the same pitch-bend value.[8] With General MIDI, there are 16 channels with one reserved for percussion, so this means that 15 notes can have an independent tuning at any given time, with each monophonic voice assigned to its own channel.[10] Since the system was designed to prototype a string quartet, using MIDI files in this manner is not a problem and is compensated for by avoiding the need for specialized software or synthesizers.

3.1 Time quantization

To finalize rhythmic values of Ratioscore free timings, compiled MIDI files can be post-processed by other software at the discretion of composers who want to generate standard graphical notation. Most notation software and some digital audio workstations can rudimentarily quantize MIDI into metered, traditional Western notation rhythms. Some parameters can be controlled, such as the type of tuplets, minimum duration, and articulation (staccato or tenuto). OpenMusic, bach composer helper, and PWGL have plugins that can fine-tune quantization.

4. AS A COMPOSITIONAL TOOL

Ratioscore was in part developed out of necessity. We wanted to be able to hear the just-intonated composition written for the JACK quartet, *The Answers.*[9] Here, the composition can serve as a compositional illustration of how Ratioscore might be used.

The Answers is a string quartet whose timings are derived from the rotation, orbit, and vantage point of the first four planets in our solar system. Sixteen points (ratios) were superimposed in a 4x4 square that circumscribes the orbit of Mars (cello) in Cartesian space.

The rhythm is determined by the rotation of each planet at their relative positions over the course of a year on Mars, scanning the points like a light beam sweeping across a square grid of stars, changing to each corresponding ratio. From each vantage point, the order and timing of sonified stars permutate, which means the combined quartet of ratios in our pitch space morphs fluidly, though maintaining a constant insistence of the pitch space, centered around the fundamental C2.

These timings, exact to the thousandth second, and their corresponding ratios were calculated in Excel and exported as a textfile for Ratioscore. The Ratioscore converter created a MIDI file simulation of the composition, as well as for each part, which was then quantized in Sibelius. In Sibelius, the MIDI pitch bends were preserved, but they had to be removed when incorporating the HEJI accidentals because C-septimal flat appeared as an altered B♮, and so on. When changing note names, the pitch bends disappear. Thus, the score in Sibelius is solely graphical. Figure 9 shows the first five measures of *The Answers* engraved using Sibelius from the MIDI files with articulation adjustments and dynamic additions, as well as the Ratioscore equivalent text score for comparison.

[9] Documentation video: https://youtu.be/Dx5l-9MUZ1o

5. CONCLUSIONS

Though the JACK quartet did not need the simulation, as they are experts in the just intonation field, the simulation was incredibly helpful in rehearsal preparation. And at the nascence of the piece, Ratioscore afforded easy experimentation with different tunings and timings. The rhythm of *The Answers* would be markedly different if it had to begin with meter rather than absolute seconds. If a ratio needed to be switched out, a simple search and replace quickly manifested the change, with immediate playback. For future work, Ratioscore would greatly benefit from an additional label to specify note names so that when the MIDI file is loaded into the notation software, pitch names could be preserved and thus Sibelius could playback the correct tunings even after the score is adjusted to accommodate HEJI accidentals.

6. REFERENCES

[1] T. Nicholson and M. Sabat, "Fundamental Principles of Just Intonation and Microtonal Composition," 2018.

[2] R. A. Wannamaker, "The spectral music of James Tenney," *Contemporary Music Review*, vol. 27, no. 1, pp. 91–130, 2008.

[3] M. Sabat and W. Von Schweinitz, "The Extended Helmholtz-Ellis JI Pitch Notation," *Plainsound Music Edition*, 2004.

[4] T. Nicholson, "Plainsound Harmonic Space Calculator." [Online]. Available: https://www.plainsound.org/HEJI

[5] M. Sabat, "Microtonal notation in Dorico," August 2018, [posted 28-August-2018]. [Online]. Available: https://www.scoringnotes.com/reviews/microtonal-notation-in-dorico

[6] T. Nicholson, "Microtonal playback in Dorico," September 2018, [by Mikhael Rimsky-Korsakov in comments section; posted 4-September-2018]. [Online]. Available: https://www.scoringnotes.com/reviews/microtonal-playback-in-dorico

[7] D. Huron, et al., "Humdrum documentation." [Online]. Available: https://www.humdrum.org

[8] MIDI Association, "MIDI 1.0 Official Specification." [Online]. Available: https://www.midi.org/specifications/midi1-specifications

[9] ——, "Standard MIDI Files." [Online]. Available: https://www.midi.org/specifications/file-format-specifications/standard-midi-files

[10] ——, "General MIDI." [Online]. Available: https://www.midi.org/specifications/midi1-specifications/general-midi-specifications

USING MUSIC FEATURES FOR MANAGING REVISIONS AND VARIANTS IN MUSIC NOTATION SOFTWARE

Paul Grünbacher
Johannes Kepler University Linz
Software Systems Engineering
Linz, Austria
paul.gruenbacher@jku.at

Rudolf Hanl
Johannes Kepler University Linz
Software Systems Engineering
Linz, Austria
rudi@ruka.at

Lukas Linsbauer
Technische Universität Braunschweig
Software Eng. and Automotive Inf.
Braunschweig, Germany
l.linsbauer@tu-braunschweig.de

ABSTRACT

Music engravers need to manage both revisions and variants of digital music artifacts created with music notation software. However, existing version control systems such as Git fail to manage fine-grained revisions and variants in a uniform manner. This paper presents an approach that uses music features and applies a variation control system in the domain of music notation. In particular, we extended the variation control system ECCO to support the evolution of digital music artifacts encoded in LilyPond. We illustrate music features using a running example. We present basic feature-oriented workflows and discuss the architecture and implementation of our LilyECCO tool. We further present a preliminary evaluation based on an existing LilyPond music artifact.

1. INTRODUCTION

Music notation tools encode music as digital artifacts using languages such as MEI, MusicXML, LilyPond, or Humdrum to name but a few [1]. As with any digital artifact, music notation tools face significant challenges of managing changes. In particular, the continuous evolution leads to many *versions* of music artifacts. Two kinds of versions can be distinguished [2]: *revisions* are the result of evolution in time, e.g., adding some dynamics to a note, correcting a slur, or fixing the pitch of a note. Revisions thus denote sequential versions representing a snapshot of the evolution of a music artifact. *Variants* on the other hand stem from evolution in space, e.g., adding lyrics in an additional language or adding a voice for an instrument needed for a new edition of a piece. Variants thus denote versions of music artifacts that need to exist concurrently.

The issue of managing revisions and variants becomes essential especially if a team of music engravers collaboratively defines and evolves digital music artifacts. Currently, engravers use general purpose version control systems such as Git to manage revisions and variants. However, while existing version control systems are powerful for handling revisions of digital artifacts, they have deficiencies with respect to managing variants. In particular, the available branching and forking mechanisms of Git and similar tools conceptually create clones. This duplication of artifacts considerably increases the maintenance effort as changes will need to be propagated manually to all clones.

Managing the evolution of music artifacts has many similarities with managing the evolution of artifacts in other domains. For instance, a number of *variation* control systems [3] have been conceived in the field of software engineering. Variation control systems provide capabilities for uniformly handling both revisions and variants based on decomposing digital artifacts into finer-grained variable entities called *features* [4]. In software engineering, for example, features are used to describe user-visible characteristics of software systems (e.g., specific functions). Features are then used to manage different revisions and variants of software systems. Properly decomposing a software system into features is seen as essential for the immediate and long-term success of software systems [4] and case studies demonstrate the usefulness and feasibility of feature-oriented version control for large-scale systems [5].

This paper investigates the suitability of applying features and variation control systems in the domain of music notation. We demonstrate the basic idea of features in music artifacts using a running example. We then show how music features can be used to manage revisions and variants of music artifacts. In particular, we extended the variation control system ECCO [6, 7] to support revisions and variants of music artifacts encoded in LilyPond [8]. We present the feature-oriented workflows and briefly discuss the architecture and implementation of our LilyECCO tool. We further present a preliminary evaluation based on a complex music artifact.

2. MUSIC FEATURES

The question of what constitutes a feature depends on the application context and the domain of interest. For instance, a common definition from software engineering describes a feature as “a distinguishable characteristic of a concept (system, component, etc.) that is relevant to some stakeholder of the concept” [9]. Applying this rather general definition to music notation means to find distinguishable characteristics of music relevant to a music engraver. Figure 1 shows our running example demonstrating possible examples of music features. When studying the first few measures of this vocal score by Claude Debussy we

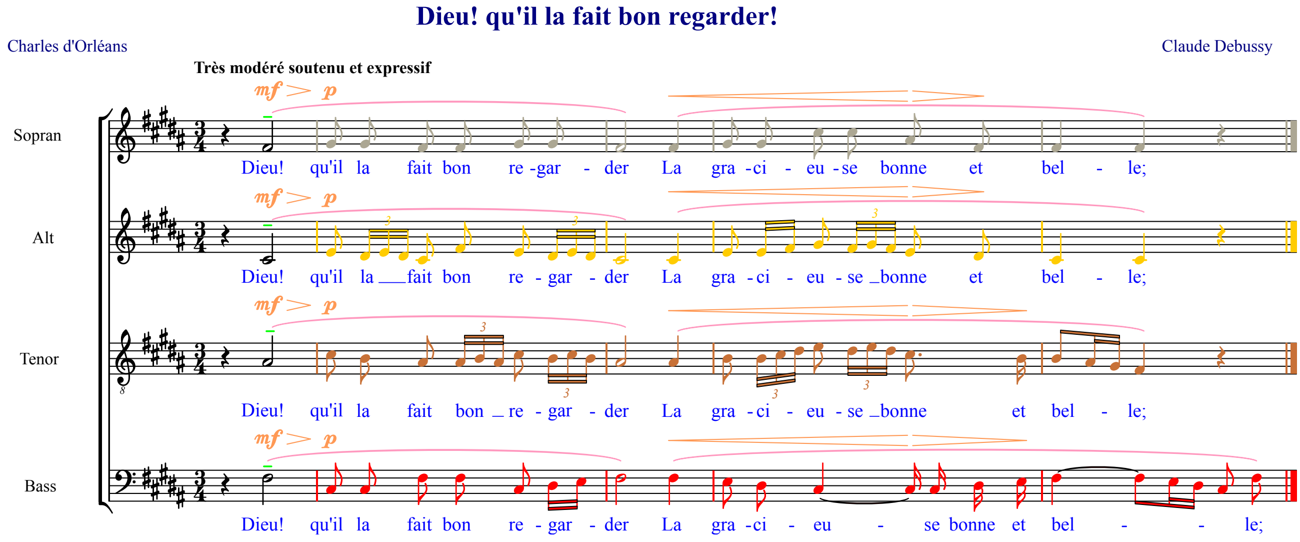


Figure 1. Ten music features in the piece *Dieu! qu'il la fait bon regarder!* (by composer Claude Debussy and poet Charles d'Orléans) – setup (black), bass notes (red), tenor notes (brown), alto notes (yellow), soprano notes (grey), dynamics (orange), slurs (pink), articulations (green), lyrics (light blue), and piece header (dark blue).

can find features for setting up the score (setup), for defining notes (soprano notes, alto notes, tenor notes, bass notes), for handling texts (lyrics, piece header), as well as features concerning articulations, dynamics, and slurs. In Figure 1 colors are used to visually indicate the mapping of elements in the score to features. The running example shows that a music artifact can be composed based on music features. The figure shows one particular version of the music artifact. However, different revisions and variants can be created by including or excluding different (revisions of) music features.

We now use this example to define and illustrate important concepts of music features based on existing work on feature-oriented software engineering [9, 3]:

Mandatory and optional music features. Common features that are present in all variants of a music artifact are referred to as mandatory features. For instance, we could assume that in our example this is the case for the feature soprano notes. Optional features on the other hand exist only in some variants, e.g., an English translation of the lyrics may not be provided in all its editions.

Revisions of music features. Revisions of features denote sequential versions and are the result of evolution in time, e.g., fixing the pitch of a note. For instance, the revision soprano notes.2 may fix a false note accidentally introduced by the engraver in the initial revision soprano notes.1.

Alternatives of music features. In many practical cases different alternatives of features are needed. For instance, one could imagine different editions of our running example created based on different translations of the lyrics.

Composing music features. Mandatory music features appear in all variants of a music artifact, while optional and alternative features result in different variants of a music artifact, which depend on the selection of the desired features by an engraver.

3. INTENSIONAL AND EXTENSIONAL VERSIONING

We investigate if and how music features can be used for managing revisions and variants in music notation. The field of software configuration management has developed a wide range of methods and tools, which pursue two important versioning strategies [2]:

Extensional versioning assumes that all existing versions are explicitly enumerated. It then allows to retrieve all versions that have been created before. Examples of tools supporting extensional versioning are Git or Subversion, which keep track of the evolution history by assigning revisions to states of a system over time. However, in practice, evolution is rarely just a linear sequence of steps, and current tools thus provide branching mechanisms for dealing with variants: for instance, short-term feature branches exist for as long as it takes to develop a new feature in isolation. Once the new feature is finished, the branch is no longer used and merged with the original artifact. However, at this point the new feature becomes inseparable from the rest of the artifact, i.e., its location in the artifact is not managed explicitly. The purpose of long-term branches on the other hand is to create clones of existing artifacts, based on which variants are then created. However, long-term branches quickly leads to maintenance problems as updates and fixes need to be propagated to all variants.

Intensional versioning aims at overcoming these limitations by providing fine-grained mechanisms for managing variants, thereby avoiding feature branches or variant branches. Furthermore, this strategy allows to create versions that have not been explicitly enumerated and committed before. Examples of tools supporting intensional versioning are variation control systems such as ECCO or SuperMod [3]. They use concepts like features, configurations, and construction rules to compose arbitrary versions.

We will show in this paper that such composition is pos-

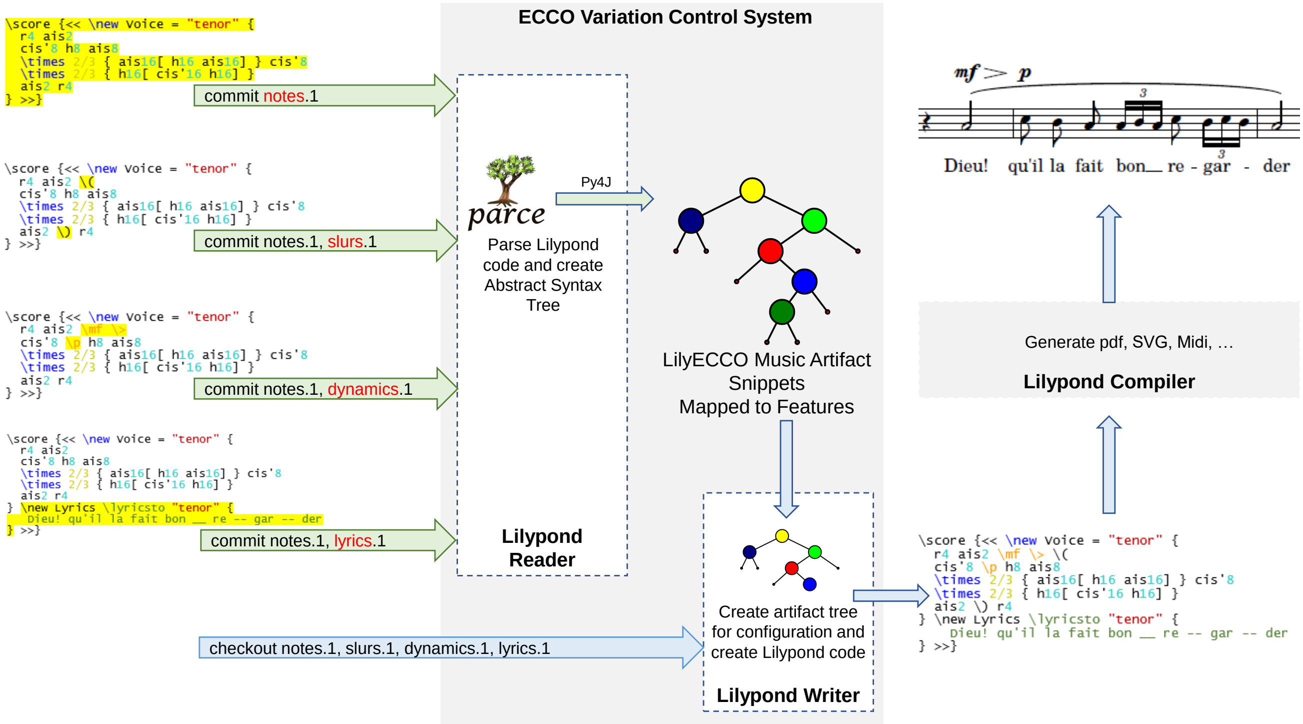


Figure 2. Workflow and architecture of LilyECCO.

sible for music features, however, there are two key challenges: (i) automatically composing features relies on *creating precise and fine-grained mappings of features to elements* of music artifacts (cf. Figure 1). Creating such mappings manually quickly becomes infeasible, thus making tools indispensible; (ii) features are not independent of each other and so *handling interactions between features* [10] becomes essential. As a simple example consider the last two bars of the bass voice in the running example. The alignment of the lyrics works if both the tie and the lyrics are present, while problems of misaligning the text would occur in LilyPond if excluding the (fictitious) feature ties.

4. LILYECCO: MANAGING FEATURE-ORIENTED REVISIONS AND VARIANTS OF MUSIC

Our LilyECCO approach for feature-oriented version control of music uses LilyPond [8], a computer program and domain-specific language for music engraving. The native text-based input language for LilyPond is comprehensive and provides commands needed for engraving classical music, complex notation, early music, modern music, tablature, vocal music, or lead sheets. LilyPond automatically computes the details of music layout, thereby allowing composers, transcribers and publishers to focus on the music instead of tweaking the layout.

To illustrate the typical feature-oriented workflow of LilyECCO we use our running example to illustrate its evolution in different evolution steps, ultimately resulting in different revisions and variants. We demonstrate how a music engraver can incrementally add music features to a repository and later compose variants based on the repository by specifying a feature configuration. We also illustrate how LilyECCO internally manages snippets of music artifacts in a feature-oriented manner.

Adding music features to an ECCO repository is possible by executing the operation commit, thereby telling LilyECCO which features (and feature revisions) are contained in a change. The left part of Figure 2 illlustrates how a music engraver would commit features to the ECCO repository incrementally to create the running example. Due to space restrictions only partial code can be shown. The engraver first commits the notes, and then adds slurs, dynamics, and lyrics. After each step, the engraver commits changes by defining combinations of (revisions of) features, as shown by the commit commands on the green arrows. As soon as the engraver commits a change, the newly added or changed LilyPond source code must be mapped to the correct features. In LilyECCO, this is done using automated code analysis to extract the newly added source code artifacts and then map them to the newly created feature(s). For instance, when committing notes.1 and slurs.1, LilyECCO determines that the code marked as yellow (the newly added phrasing slur) needs to be mapped to the new feature slurs, while the unchanged code is still mapped to the feature notes.

Creating a music variant based on an ECCO repository is supported by executing the operation checkout, which composes (combinations of) features already stored in the repository. A music engraver can create music variants at any time. For example, in our running example shown in Figure 2 the engraver checks out a music variant based on a configuration expression defining the first revision of the features notes, slurs, dynamics, and lyrics as indicated by

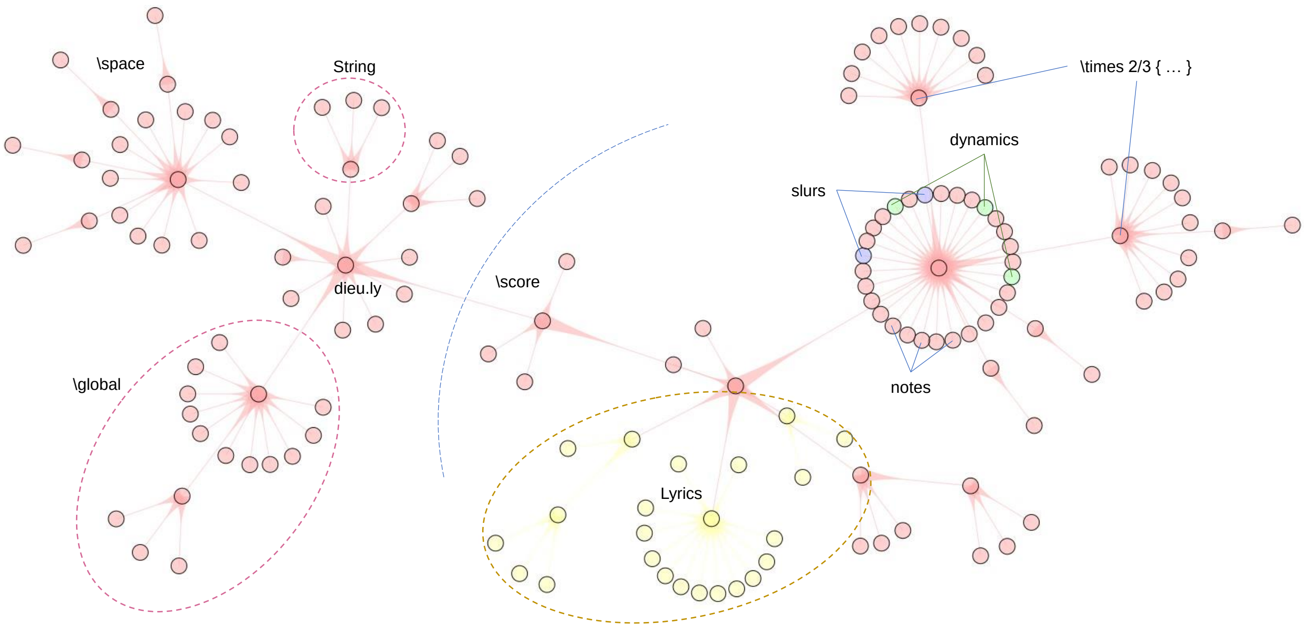


Figure 3. Selected music artifact snippets for the running example. Each node refers to an artifact snippet while colors represent features. Labels denote individual artifacts (e.g., \global, \times 2/3) as well as features like slurs or dynamics. The edges are directed and represent containment, thus resulting in a tree structure. Dashed lines are used to indicate artifacts subtrees (e.g., Lyrics).

the blue arrow. LilyECCO then automatically composes the code shown on the right. The workflow continues and the engraver can modify the generated code to create a new revision or variant, and again commit the changes to LilyECCO if desired.

Throughout the continuous evolution of a music artifact, the presented workflow is used to add new features or to extend and update existing ones. Conducting this feature-oriented workflow is infeasible without tool support for any non-trivial music artifact. For instance, feature mappings need to be continuously updated and feature interactions need to be managed. LilyECCO supports such a feature-oriented process.

5. LILYECCO ARCHITECTURE

LilyECCO's architecture shown in Figure 2 comprises the following key components:

Variation Control System. It has been shown that variation control systems can address both challenges of creating precise mappings and managing feature interactions [3]. We selected the variation control system ECCO for our purpose [6, 7]. It stores music features in a repository in the form of artifact graphs and maintains mappings between features and snippets of the music artifacts. ECCO further supports the evolution of features over time by considering feature revisions in the automatically computed traces.

For the purpose of illustration Figure 3 shows a simplified and partial view of the music artifact snippets of Debussy's piece after committing the features shown in Figure 1. This internal representation allows to determine how specific snippets of LilyPond code map to specific features. In particular, presence conditions determine whether the artifact snippets are part of a specific version. A presence condition is a propositional logic formula with feature revisions as literals [7].

We extended ECCO to allow storing and managing music features in an artifact graph structure shown in Figure 3. For that purpose, we developed a LilyPond Reader plugin allowing to recognize and store music artifacts, and a LilyPond Writer plugin to automatically compose LilyPond code for a chosen configuration:

LilyPond Reader. More technically, the Python package parce [11] parses LilyPond input into a tree structure based on the LilyPond language definition. This tree structure is then used by our LilyPond Reader plugin to analyze LilyPond input files and to create an artifact graph managed by ECCO. The ECCO system is implemented in the programming Java, while the parce parser is written in Python. We thus use Py4J (https://www.py4j.org) to execute parce (version 0.13) from our LilyPond Reader plugin written in Java.

LilyPond Writer. The LilyPond Writer plugin for ECCO is capable of creating a music artifact tree for a specific configuration. This is done by checking out a combination of (revisions of) features. LilyPond code is automatically created for the features selected by the music engraver in a configuration expression.

LilyPond Compiler. Finally, as shown in Figure 2, the LilyPond compiler processes the LilyPond code generated by the LilyPond Writer to produce output such as PDF documents, SVG files, or MIDI files.

ECCO is available as an open source system at https://github.com/jku-isse/ecco. We also plan to release our LilyECCO extensions in the future.

6. EVALUATION

Our evaluation of the LilyECCO approach was guided by two research questions:

RQ1 – Correctness. Are the mappings of features to music artifacts computed correctly? We checked the correctness of the approach by creating and replaying the evolution history of a music artifact and then automatically composing different variants.

RQ2 – Performance. Does the approach scale for real-world music artifacts? We measured the performance of executing our approach to assess its usefulness in realistic workflows.

6.1 Data Set

For our evaluation we applied the approach to a fairly complex piece of vocal music. In particular, we chose the motet ”Factus est Repente” by the Upper Austrian composer Balduin Sulzer as our data set. The piece was decidated to the vocal ensemble the first author is part of. The piece consists of two parts: the first part comprises six voices, while the second part comprises two voices. The complete piece comprises about 500 lines of code. Analyzing this piece with the parce parser used in our LilyECCO plugin results in 10,286 abstract syntax tree elements, with a maximum tree depth of six. Specifically, the most frequently occurring elements in this piece are 2,009 note pitches, 1,456 parts of lyrics, 1,170 durations of notes, 949 lyric hyphens, 688 built-in LilyPond commands (e.g., `\relative`), 632 brackets, 583 direction delimiters, 583 script literals, 354 beams, 265 numbers (e.g. 4/4), 246 rests, 188 slurs and 183 definitions of dotted notes.

We identified music features by considering both the structure of this piece (i.e., parts and voices) as well as the basic elements of musical scores (i.e., dynamics, lyrics, articulations, texts, etc.). Upon closer inspection, we identified 52 music features, which define note pitches and durations for the six voices of part 1 and the two voices of part 2 (e.g., partoneSopOneNotes, partoneBasOneNotes, parttwoSopTwoNotes), lyrics (e.g., partoneSopOneLyrics, parttwoSopTwoLyrics), score setup (header, scorePartOne, scorePartTwo, text), as well as articulations, dynamics, slurs, and beams for each voice for our evaluation.

In our experiment we regarded the complete piece as the final version of our revision history. Using the Frescobaldi editor (www.frescobaldi.org), we then manually removed one feature after the other from the LilyPond code, giving us an evolution history of 52 versions. Replaying this history reflects possible changes of a music engraver incrementally adding features to define the music artifact.

6.2 Research Method

Regarding the correctness (RQ1) of the feature-to-music mappings we performed three steps:

(i) We developed a script allowing us to automatically replay each evolution step of the data set. The script commits each version of the evolution history, thereby incrementally adding all features described above.

(ii) We checked out selected revisions and variants based on the music features defined in our data set using LilyECCO. Since LilyECCO can also produce LilyPond files that were never input by the engraver we selected examples of both *extensional and intensional versioning*: extensional versioning means to construct previously committed versions, while intensional versioning means to construct new versions based on feature combinations never committed before (cf. Section 3).

(iii) We manually inspected selected computed feature-to-music mappings in the ECCO repository. We also checked the integrity of the different variants by compiling the resulting LilyPond code. In case of syntax errors, we analyzed the reasons preventing successful compilation. Besides checking for syntax errors we also visually checked the variants in the resulting scores.

Regarding performance (RQ2) we measured the time required to analyze the different versions of the evolution history. Since the complexity of music artifacts grows over time, it is important to assess if later commits still scale, i.e., if the performance of the ECCO algorithms used to compute commonalities and differences between two successive commits allows to use LilyECCO in realistic workflows. We measured the time needed to parse the LilyPond code, and the time needed to commit the new version to the ECCO repository, which also includes the time needed to compute the commonalities and differences of different versions.

6.3 Results RQ1: Correctness

As outlined above, LilyECCO uses a tree structure to create artifact snippets, which are created and mapped to features based on the feature information in the commit command. In particular, the LilyECCO Reader distinguishes between different contexts (default, string, comment) resulting in tokens at different depths (default, brackets, keywords, lyrics, numbers, pitches and delimiters). More specifically, ECCO generated 11,786 artifacts snippets when replaying the evolution history for the data set we used in our evaluation. About half of the snippets (5,322) depend on the top six mappings to features: partoneSopTwoNotes (1,001), partoneSopOneNotes (992), partoneTenTwoNotes (869), partoneTenOneNotes (839), partoneBasOneNotes (817) and partoneBasTwoNotes (804).

Regarding *RQ1 (correctness)* we show the first page of the score for four variants created based on the Sulzer data set. Figure 4 depicts four different variants:

(a) Full piece. This variant was created by checking out all music features listed in Section 5.1. The generated code compiles correctly and the score meets the expectations. Creating this variant is an example of extensional versioning [2], i.e., retrieving a previously constructed version from the repository, in the case of LilyECCO based on explicit feature names. This demonstrates that the approach can successfully compose music artifacts based on artifacts snippets stored in the repository for the case of extensional versioning.

(b) Individual voice. This variant represents an interesting case of intensional versioning [2], i.e., to automatically

(a) Full piece (extensional)

(b) Individual voice (intensional)

(c) Soprano voices Part 2 (intensional, small fixes needed)

(d) Soprano voices, notes only (intensional)

Figure 4. The first page of the score for four variants created with extensional or intensional versioning based on our data set during the evaluation: (a) shows the score with all music features; (b) shows a score variant for a single voice; the score variant in (c) includes the two soprano voices of Part two; and variant (d) includes only the notes of the soprano voices. In cases (a), (b), and (d) the variant was created without errors while minor fixes were needed in case (c) to move and remove wrongly mapped code (cf. Section 6.3.).

construct a new combination of features on demand. The intention was to create a score variant with all features needed for a single voice. As in the first case the code of the generated variant compiles correctly and the score meets the expectations.

(c) Soprano voices of Part 2. This score variant extracting the soprano voices for part two is similar to the previous one at first sight. However, in this case of intensional versioning the initial attempt resulted in a syntax error detected by the compiler. The reason is that LilyECCO had never analyzed this specific variant, and there is an interaction of features that never existed before together (cf. our example at the end of Section 3). In particular, LilyECCO could not distinguish some global definitions and score definitions of part 2, as these features had never been committed separately. However, the workflow in LilyECCO in such cases is to simply fix the syntax errors in the checked-out variant and then commit the corrected variant, thereby telling LilyECCO how the specific variant looks like. In this case this was done by moving and removing two lines of code and commit the variant again, thereby allowing LilyECCO to distinguish these two features in future checkouts. Over time, this iterative process improves the mappings of features to music elements and allows the music engravers to correctly perform intensional versioning.

(d) Soprano voices, notes only. This variant is an example of successful intensional versioning. It includes the two soprano voices, but drops all other features. In this case LilyECCO was able to successfully construct the previously unknown variant.

6.4 Results RQ2: Performance

Regarding performance we report the execution times of analyzing the different versions of the evolution history. The experiment has been conducted using a Java 13 HotSpot 64-Bit Server VM on Windows 10 running on a PC with an Intel Core i5 with 3.5 GHz and 16GB DDR3-RAM. In our data set the size of the music artifact ranges from 39 elements (AST nodes) in the first version to 10,286 elements in the 52$^{\text{nd}}$ version. The algorithms in ECCO rely on computing tree-based commonalities and differences of code, so both the size of the artifact and the interaction of features have an influence on performance.

However, performance was acceptable for the data set given the complexity of the score, the number of features, and the size of the evolution history. The *time to commit* ranged from 0.1 to 28.2 seconds (mean: 5.8; standard deviation 7.8). User-perceived performance also depends on the time needed to parse the LilyPond code and to create a tree structure that can be handled by ECCO. This *time to parse* ranged from 0.2 to 2.1 seconds (mean: 1.3; standard deviation: 0.6). With respect to overall performance, the maximum time needed for parsing and committing a version was about 32 seconds for version 52, which indicates sufficient performance for practical workflows, given that commits would normally not be made very frequently.

6.5 Discussion

The preliminary evaluation was successful in that it demonstrates the feasibility of the approach to automatically map snippets of music artifacts to features and to compose new variants of music artifacts based on the features for both extensional and intensional cases. These promising results give rise to plenty of opportunities for further research:

The question of *what constitutes a feature* depends on the application context and the eye of the beholder. For instance, features may be used in an ad-hoc fashion to track increments and additions to music artifacts (e.g., adding a new voice). However, they may also be used in a more systematic manner by planning the purpose of the different required variants in advance. This means that features used for the purpose of creating variants for music education would differ from the scenario of a music publishing house creating different scores from the same base.

Evaluating the usefulness of music features will be necessary, e.g., by conducting user studies with music engravers or by analysing existing evolution histories such as the Mutopia archives. Such studies will also help to better understand the practical advantages of LilyECCO compared to a more traditional approach using Git or a similar tool. In our evaluation we chose a rather fine-grained definition of features for the purpose of evaluating both the correctness and performance of our approach. We plan to study the impact of different levels of feature granularity on both correctness and performance.

Another important area is to look at *usability*, in particular the cognitive complexity of specifying configuration expressions. Variation control systems like ECCO use logical expressions to manage variants with features. Depending on the number of versions and the interactions of features this task may become cognitively demanding. The Cognitive Dimensions of Notations framework [12] refers to such tasks as hard mental operations. For instance, creating configuration expressions for checking out variants is difficult for engravers who think in terms of music code but not in terms of features. It may help to let users point to artifact snippets that should be included in the variant rather than having them to consider logical expressions. Also, feature models may help to reduce the cognitive load by providing a higher-level and hierarchically-organized graphical perspective, as SuperMod [13] or FORCE^2 [5, 14] show.

In terms of possible tool support an interesting capability is to *color features* in music score editors, as shown for source code of programming languages [15]. Such a feature would also ease to systematically study the granularity of music features in realistic workflows. For instance, such studies have been conducted in the domain of software engineering to better understand how complex software artifacts can be decomposed into features of different granularity [15].

Our current LilyPond Reader and LilyPond Writer plugins have so far been primarily used and tested for vocal music. Further *implementation enhancements* are required to support the full scope of the LilyPond language. This will then allow a more comprehensive evaluation of using music features for different kinds of music. We also plan to improve the performance of our approach by executing the Java and Python code of LilyECCO in a single virtual machine.

The LilyECCO extension for ECCO is based on the LilyPond system. However, our approach can also be applied to *other music notation software packages*, if they allow parsing their music artifacts to create a tree structure, which can then be analyzed by ECCO. ECCO also supports XML files. In case no API is available, an alternative thus would be to use MusicXML as an intermediate artifact format. However, this approach seems to be risky and cumbersome, given the often unpredictable results of current MusicXML exporters and importers.

Our current evaluation focused on the case of a single engraver committing music features and composing different music variants. We can extend this to support *collaborative scenarios involving multiple engravers* based on the distributed operations of the variation control system ECCO, which allows to clone repositories and to pull music features from one repository to another as shown in [16].

7. RELATED WORK

LilyECCO uses tree-based code diffing to relate music features to music elements when committing changes to the repository. Antila et al. [17] discuss the limitations of line-based diffing approaches and also propose a hierarchical diffing approach for collaboratively editing music artifacts. Similarly, Herold et al. [18] present the MusicDiff tool for comparing two files with encoded music scores, which can also visualize the differences between these encodings. However, both approaches do not use features to label changes and to support music composition as in LilyECCO.

Fournier-S'niehotta et al. [19] propose an approach that leverages a music content model for defining virtual corpora of music notation objects that allow the development of search and analysis functions across music artifacts encoded in different formats. While the approach does not consider evolution, the idea to perform analyses across diverse digital artifacts is also fundamental to LilyECCO, which can work with different kinds of artifacts.

Dannenberg has proposed to provide views on a score [20], which "contains a subset of the information in the data structure and sometimes provides alternate or additional data to that in the data structure". This would allow that a change in a score can automatically be propagated to the parts (views on the score). The composition of a variant with LilyECCO based on features can also be seen as mechanism to create views on a score, and changes committed to the shared repository could be made available to other views (variants) via committing feature revisions and again checking out variants.

LilyECCO composes snippets, i.e., partial scores, to create new scores based on a selection of features. The idea to compose new scores based on existing ones has also been proposed by Lepetit-Aimon et al. [21]. In their approach a score can be composed as an arbitrary graph of score expressions.

8. CONCLUSIONS

We presented the LilyECCO approach for managing both revisions and variants of digital music artifacts created with music notation software. Our approach uses music features and a variation control system to manage the evolution of digital music artifacts using the LilyPond language. Our preliminary evaluation investigated both the correctness and the performance of LilyECCO, overall demonstrating its feasibility. The experiences gained in the experiment further allowed us to identify a number of interesting research directions for using music features in music notation workflows.

9. REFERENCES

[1] W. Lemberg, L. F. Moser, and U. Liska, "Music Engraving Conference, Music University Mozarteum, Salzburg," 2020. [Online]. Available: https://gitlab.com/MusicEngravingConference/2020

[2] R. Conradi and B. Westfechtel, "Version models for software configuration management," *ACM Computing Surveys*, vol. 30, no. 2, pp. 232–282, 1998.

[3] L. Linsbauer, F. Schwägerl, T. Berger, and P. Grünbacher, "Concepts of variation control systems," *Journal of Systems and Software*, vol. 171, p. 110796, 2021.

[4] T. Berger, D. Lettner, J. Rubin, P. Grünbacher, A. Silva, M. Becker, M. Chechik, and K. Czarnecki, "What is a feature? a qualitative study of features in industrial software product lines," in *Proceedings 19th International Software Product Line Conference*, ser. SPLC'15. Nashville, USA: ACM, 2015, pp. 16–25.

[5] D. Hinterreiter, L. Linsbauer, K. Feichtinger, H. Prähofer, and P. Grünbacher, "Supporting feature-oriented evolution in industrial automation product lines," *Concurrent Engineering*, vol. 28, no. 4, pp. 265–279, 2020.

[6] S. Fischer, L. Linsbauer, R. E. Lopez-Herrejon, and A. Egyed, "Enhancing clone-and-own with systematic reuse for developing software variants," in *30th IEEE International Conference on Software Maintenance and Evolution, Victoria, BC, Canada, September 29 - October 3, 2014*. IEEE Computer Society, 2014, pp. 391–400.

[7] L. Linsbauer, R. E. Lopez-Herrejon, and A. Egyed, "Variability extraction and modeling for product variants," *Software & Systems Modeling*, vol. 16, pp. 1179–1199, Jan 2017.

[8] "Lilypond – notation reference," 2020. [Online]. Available: https://lilypond.org/doc/v2.20/Documentation/notation-big-page.html

[9] K. Czarnecki and U. Eisenecker, *Generative Programming: Methods, Tools, and Applications*. Boston, MA: Addison-Wesley, 2000.

[10] P. Zave, "Feature interactions and formal specifications in telecommunications," *Computer*, vol. 26, no. 8, pp. 20–28, Aug 1993.

[11] W. Berendsen, "The parce module," 2020. [Online]. Available: https://parce.info

[12] A. Blackwell and T. Green, "Notational systems–the cognitive dimensions of notations framework," in *HCI Models, Theories, and Frameworks*, ser. Interactive Technologies, J. M. Carroll, Ed. San Francisco: Morgan Kaufmann, 2003, pp. 103–133.

[13] F. Schwägerl and B. Westfechtel, "Integrated revision and variation control for evolving model-driven software product lines," *Software & Systems Modeling*, vol. 18, no. 6, pp. 3373–3420, 2019.

[14] K. Feichtinger, D. Hinterreiter, L. Linsbauer, H. Prähofer, and P. Grünbacher, "Guiding feature model evolution by lifting code-level dependencies," *Journal of Computer Languages*, p. 101034, 2021.

[15] C. Kästner, S. Apel, and M. Kuhlemann, "Granularity in software product lines," in *30th International Conference on Software Engineering (ICSE 2008), Leipzig, Germany, May 10-18, 2008*, W. Schäfer, M. B. Dwyer, and V. Gruhn, Eds. ACM, 2008, pp. 311–320.

[16] D. Hinterreiter, L. Linsbauer, F. Reisinger, H. Prähofer, P. Grünbacher, and A. Egyed, "Feature-oriented evolution of automation software systems in industrial software ecosystems," in *23rd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2018)*, Torino, Italy, 2018, pp. 107–114.

[17] C. Antila, J. Treviño, and G. Weaver, "A hierarchic diff algorithm for collaborative music document editing," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'17*, H. L. Palma, M. Solomon, E. Tucci, and C. Lage, Eds. A Coruña, Spain: Universidade da Coruña, 2017, pp. 167–170.

[18] K. Herold, J. Kepper, R. Mo, and A. Seipelt, "MusicDiff – A Diff Tool for MEI," in *Music Encoding Conference Proceedings*, E. De Luca and J. Flanders, Eds. Humanities Commons, 2020, pp. 59–66.

[19] R. Fournier-S'niehotta, P. Rigaux, and N. Travers, "Is there a data model in music notation?" in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'16*, R. Hoadley, C. Nash, and D. Fober, Eds. Cambridge, UK: Anglia Ruskin University, 2016, pp. 85–91.

[20] R. B. Dannenberg, "Music representation issues, techniques, and systems," *Computer Music Journal*, vol. 17, no. 3, pp. 20–30, 1993.

[21] G. Lepetit-Aimon, D. Fober, Y. Orlarey, and S. Letz, "Inscore expressions to compose symbolic scores," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'16*, R. Hoadley, C. Nash, and D. Fober, Eds. Cambridge, UK: Anglia Ruskin University, 2016, pp. 137–143.

A WEB BASED ENVIRONMENT EMBEDDING SIGNAL PROCESSING IN MUSICAL SCORES

Dominique Fober **Yann Orlarey** **Stéphane Letz** **Romain Michon**
Grame-CNCM
fober@grame.fr

ABSTRACT

We present an online environment for the design of musical scores, also allowing for the embedding of signal processors and hence the publication of electronic works. This environment is part of the INScore project. Its latest version has been transcribed to WebAssembly/Javascript to provide in a web browser the same features as in its native counterpart: the diversity of music representations supported by INScore, the interaction capabilities and all the dynamic aspects of the score.

After some historical elements about distributed musical scores, we will provide some reminders about the INScore project and its associated description language. We will then describe the architecture of the system and the choices made for its portability to the Web. Then, we will present the extensions specific to the Javascript version and in particular the support of signal processing objects. Finally, we will show how INScore's communication system has been extended to allow online musical score control from a native version of INScore, paving the way for real-time performance on the web.

1. INTRODUCTION

The deployment of music notation tools on the Internet has been investigated since the late 1990s. The Guido Note Server [1], designed as a client-server architecture and based on the Guido Music Description Language [2] (GMN) is an example of such systems. It was followed by a large number of applications offering online music editing services in a design modelled on traditional score editors (e.g., MuseScore [1]), enhanced by sharing services. In this area, we can mention Noteflight, [2] Scorio, [3] , or also in the line of description languages associated to compilers, LilyBin [4] or the GuidoEditor, [5] the latter having the particularity to embed the compiler in a web page. All these systems are based on a traditional approach of musical notation and do not deal with problems related to network performances.

It is more recently and often thanks to the impulse of composers that distributed score systems have emerged. Quintet.net [3] – an interactive Internet performance environment enabling up to five performers to play music in real-time over the Internet under the control of a *conductor* – is among the first performance-oriented notation systems. The Decibel Score Player [4] is another approach to distributed musical score, based on a purely graphical notation music (as opposed to symbolic notation). It allows for the synchronization of the scores of a performing ensemble. However, these systems are implemented as native applications (Quintet.net is based on Max/MSP and the Decibel Score Player is a standalone application for the iPad) and are therefore potentially not suited to be distributed on the web.

Facing similar problems, SmartVox [5] uses a standard browser to distribute and synchronize musical scores, which are also accompanied by audio signals. In the same line but with a focus on improvised music, John, the semiconductor [6] is another web-based approach to music notation. It is more recently that Drawsocket [7] appears, a platform for generating synchronized, browser-based scores across an array of networked devices. Firmly rooted in web technologies (i.e., SVG, CSS, HTML and Javascript), it provides an API to develop networked scores.

INScore [8] (presented in section 2), is an environment for designing dynamic, interactive and augmented musical scores, built as a message-based system and controlled by OSC messages. It is naturally oriented towards network communication and is also open to web uses [9], in particular due to web server objects (`http` or `websocket`) that can be embedded in a score, and by providing a basic Web API allowing us to interact with the score from a browser. However, this approach is inherent to the native application and constrains its use as a client/server architecture, limiting both the ability to interact with the score and its dynamical aspects. We have therefore developed a Javascript version of the INScore environment, in the form of a library that can be integrated into a web page as a standalone engine. Taking advantage of the modular architecture of the Web, in particular thanks to the Node Package Manager (NPM), this implementation makes it possible to embed the Faust compiler [10] [11] and thus to provide signal processing objects within the score. This type of extension was never considered for the native version be-

[1] MuseScore `https://musescore.com/`
[2] Noteflight `https://www.noteflight.com/`
[3] Scorio `https://www.scorio.com/`
[4] LilyBin `http://lilybin.com/`
[5] GuidoEditor `https://guidoeditor.grame.fr/`

cause we considered that collaboration of specialized applications was a flexible model to implement. It is indeed possible to use INScore with Max/MSP as well as with SuperCollider or any other audio application, as long as it can communicate via OSC. The equivalent model of the Web lies rather in the aggregation of components.

Finally, a simple extension of the existing communication scheme has been designed to allow a web score to be controlled from the native version of INScore.

The next section is a quick review of INScore's approach to music representation. The following sections will detail the more technical aspects of the implementation for the web, the integration of Faust objects and the extension of the communication scheme, before concluding with the new perspectives offered by this environment and future works.

2. INSCORE ENVIRONMENT

INScore is an environment for the design of augmented, dynamic, and interactive musical scores [12]. It is the result of numerous research works dealing in particular with the extension of music notation to arbitrary graphical objects, time synchronization in the graphic space [13], dynamic and interactive scores [14], performance representation [15], and the extension of the score to network dimensions [16]. The design of a score is based on a specific scripting language and therefore also addresses the field of programming languages for the description of music [17].

2.1 Extended Scores

INScore allows you to extend symbolic notation practices, or even replace it, with arbitrary graphical objects: images, text, vector graphics, and videos. All these objects, including symbolic notation, have the same status as musical objects and an identical temporal dimension (i.e., date, duration and tempo). This homogeneity makes it possible to synchronize them in arbitrary combinations.

2.2 Representing the Time of Heterogeneous Objects

INScore takes advantage of the homogenous temporal dimension of the score objects to provide what we call *time synchronisation in the graphical space*, making it possible to represent the temporal relationships between objects using a synchronization mechanism. If we imagine that each pixel of an object carries a date (computed from the date and duration of the object) the synchronization system potentially makes it possible to graphically align all the pixels of two or more objects carrying the same date. The design of a cursor positioned at the current date of a score is achieved with a simple synchronization command. But above all, it becomes possible to reason in the temporal space and therefore in a metaphor close to musical thought, the rendering engine automatically *translate* the temporal dimensions in the graphic space.

2.3 Dynamic and Interactive Scores

The notion of *tempo* is an integral part of the temporal dimension of objects. By default, this tempo is set to zero: the object is motionless in time. When the tempo value is not zero, the object then moves in time at the speed specified by its tempo. Associated with the synchronization system, the use of the tempo allows you to create dynamic scores, whose form and content can evolve autonomously in time.

An interaction system complements these dynamic aspects. The time of a score, conceived as musical time, relative to a tempo, can also be *event based*, i.e. relative to asynchronuous events, which can be programmed in an arbitrary way. Among these events are the classic user interfaces events (such as mouse clicks, for example) but also *temporal* events, whose occurrence depends on the flow of time. Each object of the score is therefore capable of monitoring arbitrary events, including in the time domain: each event is associated with a set of messages that will be triggered at each occurrence of the event. These messages, expressed in INScore's scripting language, potentially allow the re-programming of all or part of the musical score.

2.4 The Network Dimensions of the Musical Score

INScore was originally designed to be driven by Open Sound Control (OSC) messages. It is therefore particularly suited to networks and the exchange of messages between INScore scores are native features. A simple *message forwarding* system, allows a score to control a set of other scores distributed over a local network, or to build a distributed music system on a client/server model [18].

As mentioned before, a score can also embed a web server [9], making it available from the Internet and providing control from a browser. On the other hand, the objects of a score can refer to resources distributed over the Internet, similarly to a browser that can aggregate content from different websites.

2.5 INScore Scripting Language

A score is described in a specific scripting language consisting of a textual form of OSC messages, extended with variables, control primitives, as well as symbolic score composition primitives. The following script is used to concisely describe the score in Figure 1:

```
/ITL/scene/title set txt "This is my first score !";
/ITL/scene/title scale 3;
/ITL/scene/title y -0.6;
/ITL/scene/title fontFamily Zapfino;
/ITL/scene/frame set rect 1.5 0.5;
/ITL/scene/frame color 230 230 230;
/ITL/scene/score set gmn '[ \meter<"4/4"> \key<-1> a f
g c c g a f ]';
/ITL/scene/score scale 0.6;
```

In fact, the above script mixes two languages: the INScore language whose general form is '/address parameters...', and the Guido language [2] which is used to specify the content of the `score` element.

Figure 1. A simple score, described in a few lines.

3. INSCORE WEB ARCHITECTURE

INScore is based on a Model View Controller [MVC] architecture: the Model is an abstract description of the musical score, it includes all the properties of the elements which are organized in a tree, in a strictly similar way to their OSC address. The View is a graphical representation of the Model. The controller takes input messages, decodes them to modify the Model and when necessary, activates the refresh of the View at regular time intervals (every 10ms by default). This architecture was used to differentiate the method of handling the Model and the View in the Web implementation.

3.1 INScore Model as a WebAssembly Library

Mozilla developers have started the Emscripten compiler project [19] on the Internet using the LLVM technology. It initially allowed for the generation, from C/C++ sources, of a statically-compilable and garbage-collection-free typed subset of Javascript named asm.js. This first approach has demonstrated that near-native-code-performances could be achieved on the Web. Asm.js has been followed by WebAssembly [6] [WASM], a new efficient low-level programming language for in-browser client-side scripting, faster than the previous approach.

The existing INScore Model, developed in C++, was compiled with Emscripten to produce a WASM library. As a result, the native and the Web versions share the "main" of the code, which greatly minimises the maintenance of both platforms.

3.2 INScore View as DOM Based Javascript Library

INScore View has been developed using Typescript, [7] a language which builds on JavaScript, by adding static type definitions, allowing the TypeScript compiler to validate that code is working correctly. It is compiled as a Javascript library.

The View implementation is entirely based on the Document Object Model [DOM] as defined by the W3C. [8] It creates HTML elements on the fly and makes an extensive use of SVG. Most of the score objects properties are translated into style attributes (as defined by CSS).

[6] WebAssembly `https://webassembly.org/`
[7] Typescript `https://www.typescriptlang.org/`
[8] DOM Specification

3.3 INScore Controller

The controller lies in both the WASM and Javascript libraries as shown in Figure 2. Actually, the only input of the INScore engine are text messages (unlike the native version which also accepts OSC messages). These messages can result from user actions or come from the network (see section 5.3). They are first parsed and then passed on to the objects of the Model.

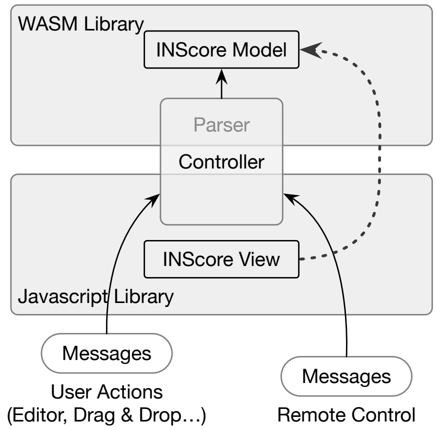


Figure 2. INScore Controller design. Input is collected from user action or received from the web to be passed to the WASM part of the controller. On changes, an update of the View is triggered and the View query the Model to synchronize.

4. SIGNAL PROCESSING EXTENSION

INScore Web can optionally embed the Faust compiler to provide signal processing objects within the score. Faust [10] is a functional, synchronous, domain-specific programming language working at the sample level, designed for real-time audio signal processing and synthesis. Faust programs can be efficiently compiled to a variety of target programming languages, from C++ to WebAssembly.

The Faust compiler is available as a WASM library [20] available as a NPM package [9] including a Javascript library providing a high level API to transform DSP code into a Web AudioNode. [10]

The type of a Faust object is `faust` and its `set` method (see Figure 3) takes DSP code as an argument. It is graphically represented by a browsable block diagram. Faust audio nodes can be instantiated as monophonic or polyphonic nodes, thus the `set` method takes an optional number of voices as illustrated below. When present (and even if equal to 1), a polyphonic Faust audio node is created.

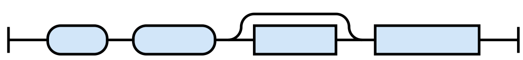


Figure 3. `set` method of Faust objects.

The following code creates a monophonic object named

[9] Faust NPM package
[10] The Web Audio API

karplus using the Faust physical modeling library, that is a ready-to-use, MIDI-enabled Karplus-Strong string with built-in UI.

```
/ITL/scene/karplus set faust
        'import("stdfaust.lib");
         process = pm.ks_ui_MIDI';
```

4.1 Faust Objects Methods

Faust objects carry all the properties common to INScore objects, including their temporal dimension. Their specific methods are the following:

- play: start or stop sound processing. Takes a boolean value ([01]) as an argument.

The next methods are only supported by polyphonic objects:

- keyOn, keyOff: similar to MIDI key on/off messages. Takes a MIDI channel, a pitch, and a velocity as arguments.
- allNotesOff: similar to MIDI all notes off message.

Faust objects support also specific query (get) methods, corresponding to read-only properties:

- in, out: gives the number of input and output signals of the Faust object.
- paths: gives the interface to the Faust object UI (as defined by the DSP code).

Paths returned by the path query are used internally to dynamically generate the address space of the Faust object (see section 4.2), providing control over the Faust node parameters.

4.2 Faust Objects Address Space

Faust provides user interface primitives allowing for an abstract description of a user interface within the Faust code. This description is independent from any GUI toolkits/frameworks and it's the *architecture files'* [21] responsibility to instantiate this abstract description. In INScore, this abstract description is instantiated in the address space of the Faust object. Let's consider the following query addressed to the karplus object as defined by the example in section 4.

```
/ITL/scene/karplus get paths;
```

The INScore engine returns a list of UI elements as follows:

```
/karplus/params/freq hslider freq 440.0 50.0 1000.0
0.01;
/karplus/params/bend hslider bend 0.0 -2.0 2.0 0.01;
/karplus/params/damping hslider damping 0.01 0.0 1.0
0.01;
etc.
```

where the general form of an element is a sequence of

address: the address of a Faust audio node parameter.

type: the type of the UI element (ignored by INScore).

name: the name of the UI element (ignored by INScore).

default: the default value of the parameter.

min: the minimum value of the parameter.

max: the maximum value of the parameter.

step: the step of values (ignored by INScore).

The address field is used to expand the address space of the Faust object so that for the object
/ITL/scene/karplus
the addresses
/ITL/scene/karplus/karplus/params/freq
/ITL/scene/karplus/karplus/params/bend
etc.
become valid addresses taking a float value as an argument, that is passed to the Faust audio node to set the corresponding parameter value.

5. WEB COMMUNICATION

5.1 Server Side

INScore provides a *fowarding* mechanism [16] that can be used to distribute scores over a local network. The general form of the forward message is illustrated in Figure 4. It can be addressed to the application or to the scene level. It takes a list of destinations as argument or can be used with no argument to stop forwarding. A destination host is specified similarly to a url, by IP number or by host name, followed by a port number. A filtering mechanism is also provided to select the messages to be forwarded. This mechanism was basically designed to transmit OSC messages on UPD sockets. It has been extended for the native INScore application, to support different protocols, namely Websockets and HTTP. The new form of the forward message is illustrated in Figure 5.

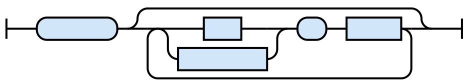


Figure 4. The forward message.

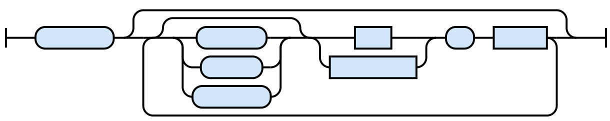


Figure 5. The extended forward message.

osc:// ws:// and http:// refer respectively to the OSC protocol, to Websockets and to HTTP. For compatibility reasons, the original form is preserved and implies the OSC protocol.

The OSC protocol runs over UDP and thus is connectionless, this is not the case for Websockets and HTTP that run over TCP. Currently and whether for Websockets or HTTP, the INScore server ignores the host name and accepts all incoming connections. This approach may be revised in the future to select authorized hosts.

5.2 Client Side

The OSC protocol is transparent on the client side: INScore has been natively designed to communicate via OSC. For Websockets and HTTP, an explicit connection must be initiated by the client and to this end, we have introduced the `connect` message whose form is similar to the `forward` message (see Figure 6). Used without argument, `connect` removes all the existing connections.

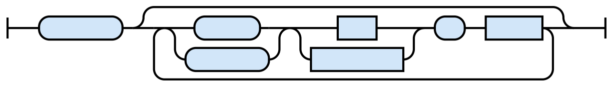


Figure 6. The `connect` message.

On client side, HTTP support is implemented over HTML Server-Sent-Events [SSE] API, a one way messaging system designed to allow a web page to get updates from a server. Websockets connections are bidirectional, but only communication from the server to the client is used for the time being.

Messages transmitted by the server are textual OSC messages that are parsed by the client upon receipt, in the same way as any input script.

5.3 Communication Scheme Overview

Figure 7 illustrates the overall communication scheme of INScore. Its extension to new protocols allows us - starting from a native version of INScore which then acts as a server - to control in parallel musical scores distributed over a local network and/or over the Internet.

The server receives as an input messages from user actions (e.g., drag & drop of scripts, interaction with the score, etc.) or generated by design according to the time flow of the score objects. Provided they are not filtered, these messages are automatically transmitted to all connected clients, making it possible to replicate and/or control a musical score on a whole set of targets.

6. CONCLUSIONS

The Web deployment of the INScore engine is an ongoing project started in 2017 in parallel to the native version development. The design issues encountered in the past were solved by the evolution of languages and compilation technologies, allowing us to support various platforms from the same source code.

The web version of INScore opens new perspectives, particularly relevant in the context of the current health crisis, where the web has taken a central place. The distribution of musical works involving electronic parts has never been straightforward. Being able to publish such works on the web, to make them available without prior installation, ready to be played from home, may constitue an essential tool for the dissemination of contemporary creation. Encoding of existing musical works in INScore and Faust is underway with this in perspective.

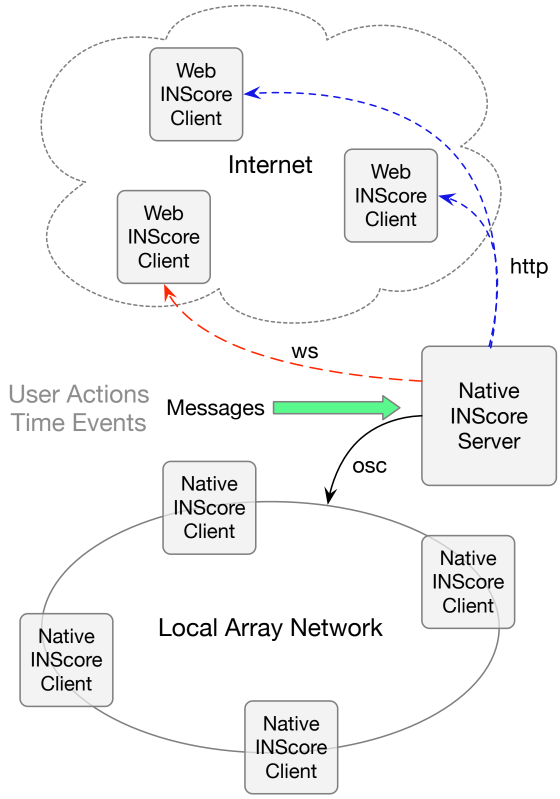


Figure 7. INScore communication scheme.

Remote control could also open up new prospects, both from an educational point of view and for the performance of the music: a composer will be able to perform his piece from home, interacting dynamically with the performance, while a set of connected listeners will be able to follow this performance, having at their disposal both the representation of the work and its sound rendering. Distributed performance was already possible on a local network, the extended communication scheme brings it to the Internet.

The presented work will be available as a library on NPM. An INScore editor is online at

`https://inscoreweb.grame.fr/`

7. REFERENCES

[1] K. Renz and H. Hoos, "A Web-based Approach to Music Notation Using GUIDO," in *Proceedings of the International Computer Music Conference*. ICMA, 1998, pp. 455–458.

[2] H. Hoos, K. Hamel, K. Renz, and J. Kilian, "The GUIDO Music Notation Format - a Novel Approach for Adequately Representing Score-level Music." in *Proceedings of the International Computer Music Conference*. ICMA, 1998, pp. 451–454.

[3] G. Hajdu, "Quintet.net: An environment for composing and performing music on the internet," *Leonardo*,

vol. 38, no. 1, pp. 23–30, 2005. [Online]. Available: https://doi.org/10.1162/leon.2005.38.1.23

[4] C. Hope, L. Vickery, A. Wyatt, and S. James, "The DECIBEL Scoreplayer - A Digital Tool for Reading Graphic Notation," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation*. Institut de Recherche en Musicologie, May 2015, pp. 58–69. [Online]. Available: https://doi.org/10.5281/zenodo.1289610

[5] J. Bell and B. Matuszewski, "SMARTVOX - A Web-Based Distributed Media Player as Notation Tool For Choral Practices," in *TENOR 2017*, Coruña, Spain, May 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01660184

[6] V. Goudard, "John, the semi-conductor : a tool for comprovisation," in *International Conference on Technologies for Music Notation and Representation (TENOR'18)*, ser. Proceedings of the 4th International Conference on Technologies for Music Notation and Representation, S. Bhagwati and J. Bresson, Eds., Montréal, Canada, May 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01923258

[7] 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*. Monash University, Jul. 2019, pp. 15–25. [Online]. Available: https://doi.org/10.5281/zenodo.3373369

[8] D. Fober, Y. Orlarey, and S. Letz, "INScore - An Environment for the Design of Live Music Scores," in *Linux Audio Conference*, Stanford, United States, 2012, pp. 47–54. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02158817

[9] D. Fober, G. Gouilloux, Y. Orlarey, and S. Letz, "Distributing Music Scores to Mobile Platforms and to the Internet using INScore," in *12th Sound and Music Computing Conference (SMC15)*, Maynooth, Ireland, Jul. 2015, pp. 229–233. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01348511

[10] Y. Orlarey, D. Fober, and S. Letz, "FAUST : an Efficient Functional Approach to DSP Programming," in *NEW COMPUTATIONAL PARADIGMS FOR COMPUTER MUSIC*, E. D. FRANCE, Ed., 2009, pp. 65–96. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02159014

[11] S. Ren, S. Letz, Y. Orlarey, R. Michon, D. Fober, M. Buffa, and J. Lebrun, "Using Faust DSL to Develop Custom, Sample Accurate DSP Code and Audio Plugins for the Web Browser," *Journal of the Audio Engineering Society*, vol. 68, no. 10, Nov. 2020. [Online]. Available: https://hal.inria.fr/hal-03087763

[12] D. Fober, Y. Orlarey, and S. Letz, "INScore - An Environment for the Design of Live Music Scores," in *Linux Audio Conference*, Stanford, United States, 2012, pp. 47–54. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02158817

[13] D. Fober, C. Daudin, Y. Orlarey, and S. Letz, "Time Synchronization in Graphic Domain - A new paradigm for Augmented Music Scores," in *International Computer Music Conference*, ICMA, Ed., New York, United States, 2010, pp. 458–461. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02158957

[14] D. Fober, S. Letz, Y. Orlarey, and F. Bevilacqua, "Programming Interactive Music Scores with INScore," in *Sound and Music Computing*, Stockholm, Sweden, Jul. 2013, pp. 185–190. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00851956

[15] D. Fober, F. Bevilacqua, and R. Assous, "Segments and Mapping for Scores and Signal Representations," GRAME, Technical Report, 2012. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02158968

[16] D. Fober, G. Gouilloux, Y. Orlarey, and S. Letz, "Distributing Music Scores to Mobile Platforms and to the Internet using INScore," in *12th Sound and Music Computing Conference (SMC15)*, Maynooth, Ireland, Jul. 2015, pp. 229–233. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01348511

[17] D. Fober, Y. Orlarey, S. Letz, and R. Michon, "A Tree Based Language for Music Score Description." in *International Symposium on Computer Music Multidisciplinary Research*, Marseille, France, Oct. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02368958

[18] S. Zagorac and P. Alessandrini, "ZScore: A Distributed System For Integrated Mixed Music Composition and Performance," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*. Concordia University, May 2018, pp. 62–70. [Online]. Available: https://doi.org/10.5281/zenodo.1289685

[19] A. Zakai, "Emscripten: An llvm-to-javascript compiler," in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, ser. OOPSLA '11. New York, NY, USA: Association for Computing Machinery, 2011, pp. 301–312. [Online]. Available: https://doi.org/10.1145/2048147.2048224

[20] S. Letz, Y. Orlarey, and D. Fober, "FAUST Domain Specific Audio DSP Language Compiled to WebAssembly," in *The Web Conference*. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 701–709. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02158925

[21] D. Fober, Y. Orlarey, and S. Letz, "FAUST Architectures Design and OSC Support." in *International Conference on Digital Audio Effects*, IRCAM, Ed., Paris, France, 2011, pp. 231–216. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02158816

INDRA: A VIRTUAL SCORE PLATFORM FOR NETWORKED MUSICAL PERFORMANCE

Drake Andersen
Vassar College
Poughkeepsie, NY, USA
dandersen@vassar.edu

ABSTRACT

This paper introduces Indra, a Max-based virtual score platform for networked musical performance. Indra allows a conductor to improvise with an ensemble over a local area network by determining the notation that appears on performers' screens in real time. Musical compositions for the Indra platform consist of short encoded or image-based notation clips that are tagged with customizable metadata corresponding to musical qualities. The conductor uses the metadata to filter clips, determining the general qualities of the musical texture while the software cycles through clips that meet the filter criteria. Indra is designed to be accessible and adaptable for musicians working in diverse styles and numerous performance contexts.

1. INTRODUCTION

Real-time music notation systems, also known as virtual scores, have been used by an increasing number of composers over the past two decades to explore novel computer-based performance paradigms. Virtual score software by creators like Nick Didkovsky, Jason Freeman, and Georg Hajdu invite composers, performers and listeners to engage with notation in new ways, whether improvising with animated graphic notations, modifying notation on the fly, or even undertaking "extreme sight reading" [1, 2].

Individual virtual score platforms exhibit diverse functionality, occupying a spectrum from software designed for the performance of an individual composition, such as Jason Freeman's *Glimmer* (2004) or Craig Vear's *On Junitaki Falls* (2017), to systems designed for a wide variety of uses with minimal stylistic or aesthetic constraints. One of the most important and influential general-purpose systems is Georg Hajdu's *Quintet.net*, first developed in 1999, which allows musicians at up to five different locations to perform together over the Internet [3]. More recent platforms for networked notation display include Decibel ScorePlayer [4], DrawSocket [5], and SmartVox [6]. In addition to their respective specialties, each of these platforms grapples effectively with the challenges of synchronizing notation across a network. However, virtual scores are also an opportunity to explore alternative, more flexible models for the selection and distribution of notation in performance.

Indra is a Max-based virtual score platform for networked musical performance that allows a conductor to improvise with an ensemble of any size by determining the notation that appears on performers' screens in real time.[1] Unlike many other virtual score platforms, Indra is intended to support a wide range of aesthetic and stylistic possibilities through its flexible interface, robust notation support, and customizable metadata and tagging systems.

Indra is designed to be accessible and adaptable so that musicians without previous experience can quickly learn to use the software and organize performances independently. You can download the latest version of the Indra software and documentation from this website: **https://creativeinteraction.org**

2. OVERVIEW

The Indra platform comprises three Max patches: one each for the composer, conductor, and performers. Since Indra is a platform and not a composition in and of itself, it can be used for the performance of many different compositions, as well as multiple performances of the same composition. Compositions in Indra are called *collections*, and can be generated individually or collectively. Existing musical works can also easily be imported and remixed using Indra. There is no functional limit to the number of performers.

The Indra workflow is deeply collaborative. The composer(s) create(s) a collection of short musical ideas or passages called *clips* that are passed along to a conductor, who interprets and arranges them during the performance alongside the musicians in the ensemble. Even though much of the material originates with the composer, the conductor and musicians shape the form, texture, and feel of each performance. Many performances with Indra incorporate improvisation, graphic notation, and indeterminacy, further distributing creative agency.

2.1 Performance Workflow

Collections are first created using the composer patch and saved as a database file that can be loaded into the other

[1] https://cycling74.com/

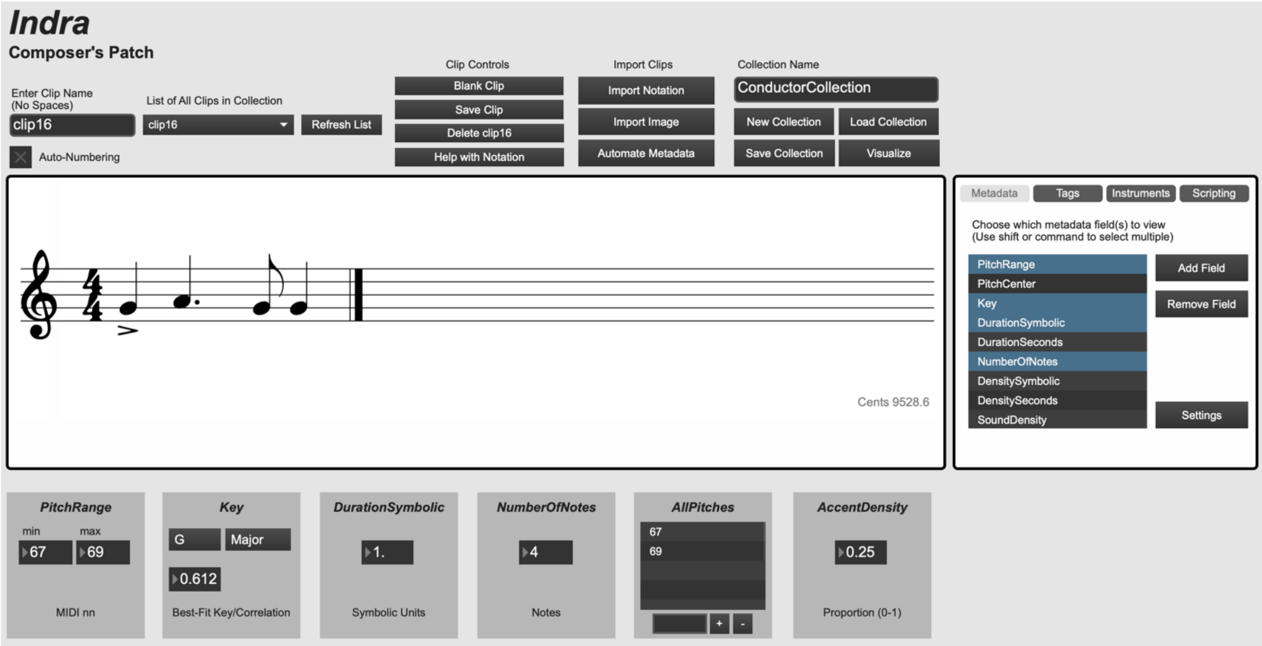


Figure 1. The composer patch during the creation of a collection. Selecting metadata fields from the list on the right dynamically populates the metadata field boxes below the notation display. The metadata field boxes are used by the composer to display and modify metadata corresponding to the clip currently being displayed.

two patches. Collections consist of short encoded or image-based notation clips that are tagged with customizable metadata corresponding to musical qualities. The conductor patch allows the conductor to connect to the performers over a local area network and send messages that determine the notation that appears on the performers' screens. The conductor filters the clips using their metadata to filter clips, determining the general qualities of the musical texture while the software continuously cycles through clips that meet the filter criteria. The primary function of the performer patch is to display notation to the performer in response to the conductor's messages.

Many existing virtual score systems are designed for expert users, and present significant obstacles to musicians without experience using specialized music software. While this is often mitigated through the direct intervention and participation of the creator, these obstacles sometimes discourage would-be users from engaging with the software. Consequently, a central design priority for Indra is ensuring a smooth workflow for musicians through a clear and consistent operational logic.

2.2 Aesthetic Background

Indra was conceived in dialogue with a variety of existing performance practices, from the open scores of Anthony Braxton and Earle Brown to the meta-language systems of John Zorn and Walter Thompson. Indra reflects a fundamentally nonlinear conception of composition, meaning that individual sounds and gestures are organized into short clips designed to be rearranged to shape musical meaning. Clips may use traditional or graphic notation, and may be divided by instrument or available to multiple performers. It is also possible to conduct nonlinear remixes of existing compositions, such as a Beethoven string quartet, or Stravinsky's *Symphonies of Wind Instruments*, by dividing them into clips.

2.2 Technical Description

Indra operates in the Max environment and makes use of the bach, cage, dada, odot, and zero packages, as well as several externals and abstractions distributed with the patch. Performances with Indra are conducted over a local-area network using UDP messages, employing the zero-configuration networking framework. Musical compositions for the Indra platform are stored as native bach *.llll files consisting of encoded notation and metadata, organized as a relational database. Compositions that employ image-based notation are distributed with an accompanying "images" folder that contains all image files.

3. NOTATION

Notation is created and modified using the composer patch. Figure 1 gives the composer patch during the process of creating a collection. Notation in Indra can be stored in a collection in one of two ways: (1) encoded as a bach *.llll file, or (2) an image file that is displayed to the performer. Indra uses the bach suite of objects [7] developed by Andrea Agostini and Daniele Ghisi to store and display encoded notation.[2] Notation for clips can be generated within Indra using the [*bach.score*] object graphic user interface, or by sending scripting messages through a text-based interface in the software. Among several Max-

[2]The official bach site: `https://www.bachproject.net/`

based notation frameworks, including MaxScore [8], bach was selected to be the basis for encoded notation in Indra due to the robust tools for analysis and visualization in the sibling cage [9] and dada [10] packages.

Users can also generate notation in the software of their choice and import it into Indra using the MusicXML format. Indra parses imported MusicXML files into clips by interpreting bars containing only whole rests as separators. Batch naming and tagging is supported. The use of encoded notation also allows for the automatic generation of metadata, as described below.

Indra supports the use of image files to represent clip notation as well. Batch importing, naming, and tagging of images is supported. Using images instead of encoded notation ensures the consistent appearance of the notation and allows for the use of graphic, colored, multi-staff, non-standard, or otherwise unsupported notational elements. One drawback, however, is that at present all metadata information for image-based clips must be entered manually.

4. METADATA AND TAGS

Indra uses a relational database structure to store three parallel layers of information about each clip: (1) metadata, (2) tags, and (3) instrument tags. This data is used by the conductor to filter clips during the performance, but remains invisible to performers.

4.1 Metadata

Metadata refers to the information stored with a clip that describes its musical qualities, including pitch content, key, duration, density, and many others. In the current version of Indra there are fifteen default metadata fields: PitchRange, PitchCenter, Key, DurationSymbolic, DurationSeconds, NumberOfNotes, DensitySymbolic, DensitySeconds, SoundDensity, AllPitches, AllPitchClasses, DurationType, DurationRange, StaccatoDensity, and AccentDensity.

Each of the default metadata fields is user-definable, but can also be generated automatically for programmatic notation using analytical tools, several of which rely on the cage and data sibling packages to bach. For example, the Key field is determined through an implementation of the Krumhansl-Schmuckler key-finding algorithm. Indra supports customizable major and minor profiles, and also outputs a correlation coefficient that can be filtered by the conductor for distinguishing between the perceived strength of a key between clips.

Several of the default metadata fields are considered *functional* metadata, meaning that they have an effect on the operation of the software in performance. For example, the DurationSymbolic and DurationSeconds fields are used to determine the length of time a clip is displayed on a performer's screen. Similarly, PitchRange data can be used to silently filter clips by instrumental range. If this setting is turned on in the conductor patch, only clips whose PitchRange values fall within the standard instrumental range of a given instrument can ever be assigned to that instrument.

In addition, there are fifteen slots for custom metadata fields whose values and formats are entirely user-definable. Custom metadata fields can take one of six formats: (1) a single integer, (2) a single floating-point number, (3) a minimum-maximum pair of integers, (4) a state from several composer-definable options, (5) a list of words and/or numbers, and (6) the special compound key format which can be used to define scales, modes, set classes, collections, and other key-like data types. Users can also specify the type of control surface the conductor will use to operate the filter in performance, including sliders, notation-based displays, keyboard-based displays, menus, and lists.

4.2 Tags and Instrument Tags

The tagging system functions in parallel with the metadata system, but independently and with somewhat different functionality. Tags are custom labels that are stored as part of a collection and can be applied to any clip. A clip can be associated with any number of tags. Figure 2 gives the tagging interface in the composer patch.

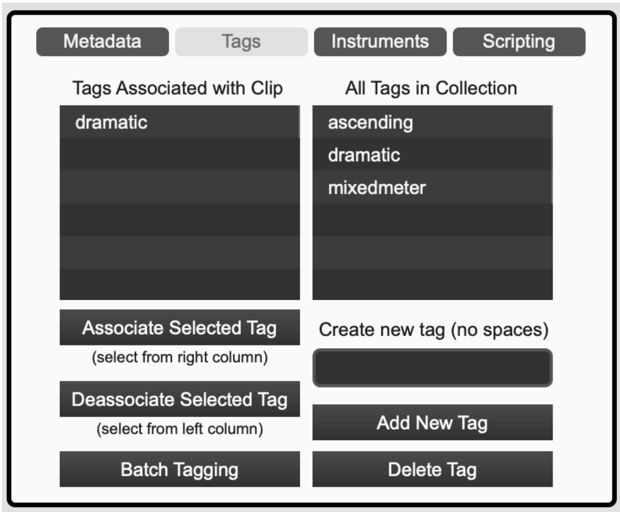


Figure 2. The composer patch tagging interface.

In addition to the standard tagging system, Indra also supports a separate tagging system specifically for instrumentation, which allows the conductor to automatically direct clips to particular instruments in performance. Instrument tags are similar to regular tags, except that they operate invisibly during performance. While performing a piece with Indra, the conductor must manually select which metadata and (regular) tags are active filters at any given moment. These settings generate a list of matching clips that are subsequently passed to the instruments. Instrument tags divide up this list so that only clips tagged for a particular instrument are actually sent to that instrument.

The conductor can utilize or ignore instrument tags as desired. Unlike regular tags, instrument tags are invisible: they do not appear in the main list of tags in the conductor patch, and once turned on operate automatically. Invisible operation speeds up the conductor's workflow by allowing the conductor to send instrument-specific clips to multiple

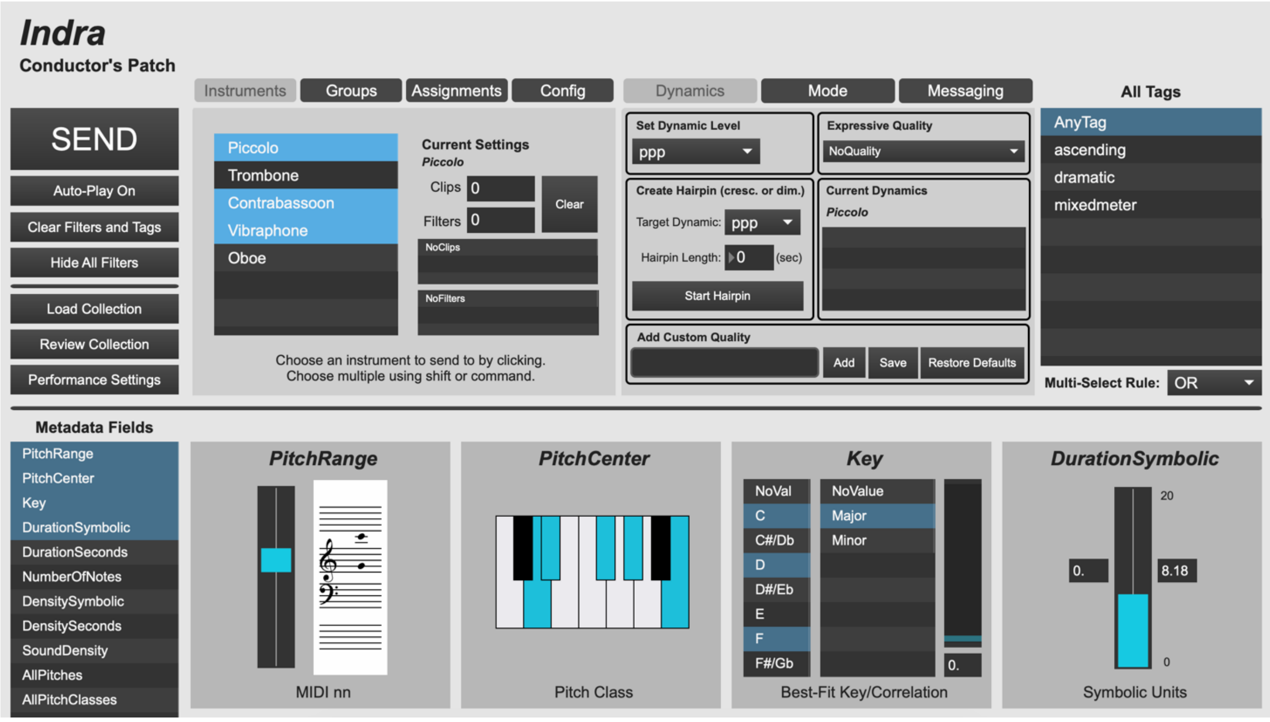


Figure 3. The conductor patch during a typical performance. As in the composer patch, the metadata field boxes appear dynamically based on the current selection in the Metadata Fields list on the left side of the screen. Unselected fields are automatically bypassed when sending new filter settings to performers.

instruments in a single operation. It also obviates the (otherwise redundant) need to select both a recipient and that recipient's instrument-specific tag. As with regular tags, a single clip can be associated with multiple instrument tags. Batch tagging is also available for instrument tags.

5. REHEARSAL AND PERFORMANCE

Indra is designed to make rehearsing and performance simple and straightforward for ensembles. Once the composer has completed the collection, the collection is distributed to the conductor and ensemble members for rehearsal and performance. The conductor leads the performance using their patch.

5.1 Filtering Clips

Figure 3 gives the conductor's interface as it might appear during a typical performance. The usual workflow is as follows. First, the conductor selects metadata fields from the Metadata Fields list in the lower-left corner. The corresponding metadata fields boxes will automatically appear in the lower half of the screen. The conductor then modifies filter settings using the interface elements in the metadata filter boxes. (The use of multiple filters selects for clips that meet all criteria.)

Next, the conductor may choose to restrict the clip selection to those with certain tags using the All Tags list in the upper right. The conductor can specify multiple tags using *AND* or *OR* logic. Finally, the conductor selects recipients by instrument or (user-definable) group from the multi-function box in the upper half of the patch on the left, and presses the large send button in the upper left corner of the patch.

This workflow order is flexible: recipients can be selected before or after modifying filter settings. Clicking the send button will send whatever tags and filter boxes are visible on screen to whichever recipients are selected.

5.2 Performance Modes and Dynamics

There are three modes to which a performer can be assigned during a performance: play, tacet, and improvise. Play mode is the usual mode of operation, in which clips are automatically cycled through the performer's display, while tacet and improvise clear the notation display. Conductors who wish to use the improvise message are encouraged to discuss how to interpret this instruction with performers during the rehearsal process.

The Dynamics tab allows the conductor to send dynamics and expression-related messages to performers. Conductors may send one of three types of dynamics or expression-related information: (1) a static dynamic level (from *ppp* to *fff*), (2) a transition from one dynamic level to another over a specified amount of time, or (3) an expression indication. Expression indications may be chosen from a list of common indications such as "rising and falling in waves" or "irregular sf," or may be customized by the user.

The messaging system allows the conductor and performers to send instant text messages to one another through the Indra system. Intended primarily for troubleshooting in real time, the messaging system supports communication from the conductor to one or more performers, or from one performer to the conductor.

6. DESIGN PRIORITIES

Indra prioritizes stability, adaptability, and a consistent operational logic so as to be as accessible as possible for musicians who are new to virtual score software. Supporting both encoded and image-based notation reflects a commitment to engaging with composers using traditional and graphic notation. The MusicXML import feature likewise allows composers to use the hardware or software of their choice to generate notation, rather than requiring them to learn and use Indra's built-in tools. This also facilitates collective and/or collaborative compositional practices.

Composers can take advantage of the customizable metadata and tagging systems to shape their performances. For example, besides strictly musical qualities, tags can also be used for theme groups, formal sections, and emotional affects. Finally, composers can use the [*dada.cartesian*] object to visualize their collection in progress by mapping default or custom metadata fields to different axes, or color, size or shape, and plotting clips as points.

The network configuration process—often the most tedious part of rehearsals and performances for musicians using networked notation systems—has been streamlined over several iterations into a fast and intuitive process through the use of the zero (Zero-configuration networking) package. This is especially true for the performer experience. Figure 4 gives the welcome screen that greets performers when they open the performer patch. Instead of having to deal with IP addresses or other technical information directly, performers simply enter a personal identifier (usually their first name and last initial) and their instrument, and then announce themselves over the network when prompted by the conductor. The zero package automatically resolves network addresses, populating a list in the conductor's patch with names and instruments.

Both the composer and conductor patches use dynamic interfaces for viewing and modifying metadata field values and filter settings, respectively. This keeps both interfaces as uncluttered as possible, while also making visually clear which filters are active and which are bypassed in the conductor patch during performance. In order to adapt to different conducting styles and performance contexts, the conductor patch settings can be saved separately from a given collection. For example, a conductor can save multiple rosters of performers in multiple settings files, but use the same collection for performances with each. Conductors can also assign and store custom groupings of performers, and select groups during performance using a single click, as demonstrated in the close-ups of the conductor patch given in Figures 5 and 6.

The simplicity of the performer patch, given in Figure 7, is intended to resemble traditional sheet music. At the same time, several features have been added as a direct result of previous performance experiences, including a preview window displaying the next clip to appear, an optional reference pitch for vocalists, a button to skip the current clip, a button to change clefs, and a flashing indicator to accompany the countdown as each clip advances. In addition, the performer patch includes a practice mode so that musicians can load a collection and cycle through clips on their own.

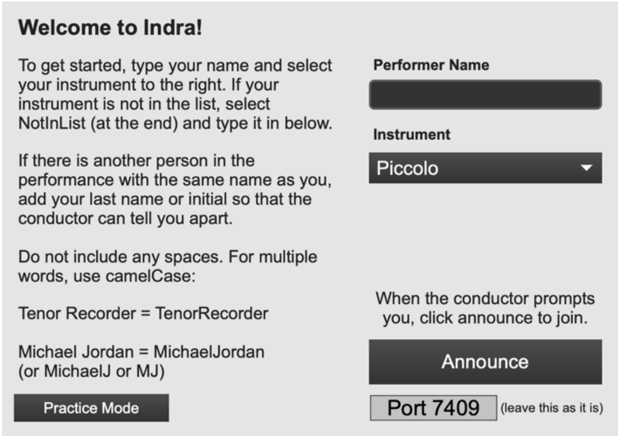


Figure 4. The welcome screen in the performer patch.

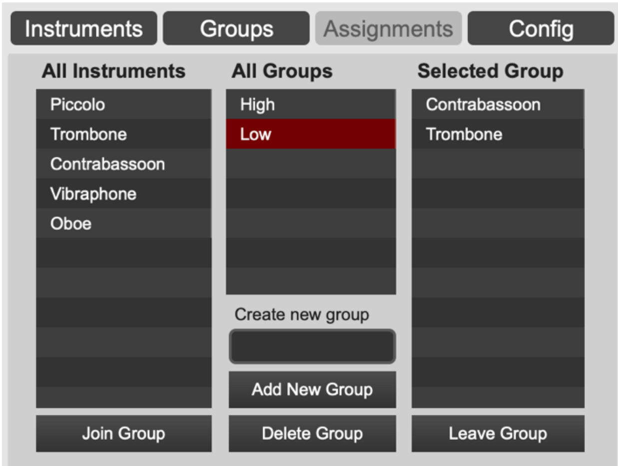


Figure 5. The Assignments tab in the conductor patch.

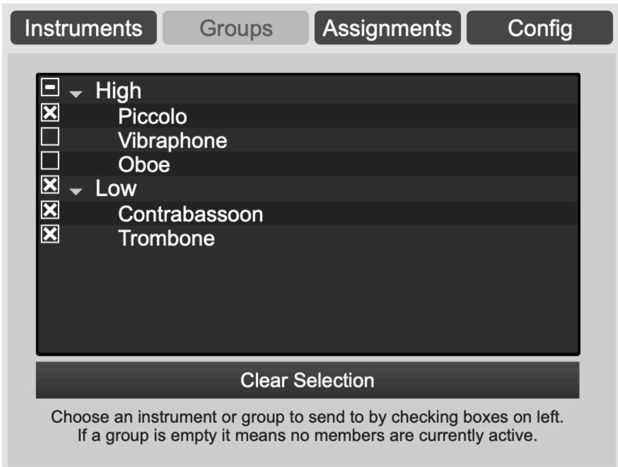


Figure 6. The Groups tab in the conductor patch. Clicking on the checkbox to the left of a group name selects all instruments within that group as recipients for the next message to be sent.

Figure 7. The performer patch during a typical performance.

7. CONCLUSIONS AND FUTURE WORKS

Indra is an accessible and adaptable virtual score platform for musicians working in diverse styles and numerous performance contexts. While Indra was originally conceived for concert settings, the composer Robert McClure recently used Indra to perform the musical accompaniment to a work for dance. Likewise, the author has organized performances using Indra that combine acoustic and electroacoustic forces, such as the chamber concerto for viola and ensemble *Spring Flow* premiered by Kallie Ciechomski. Upcoming workshops and performances focus on the creative exchange that takes place when composers conduct each other's works.

The next step in the development process will be incorporating gesture recognition technology into the conductor's interface for more intuitive and efficient control of the software. Additional areas for future development include using optical music recognition to automatically generate metadata from imported image files, and producing more extensive multimedia documentation and example collections.

Acknowledgments

Many thanks are due to Martha Cargo, Kallie Ciechomski, Jordan Dodson, Robert McClure, Charlotte Mundy, Mike Perdue, Yumi Tamashiro, and Jude Traxler for their participation and helpful feedback in testing and early performances. I am also grateful to Sam Wolk for his assistance in developing this project.

8. REFERENCES

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

[2] C. Vear. *The Digital Score: Musicianship, Creativity and Innovation.* Routledge, 2019.

[3] G. Hajdu, K. Niggemann, A. Siska, and A. Szigetvari. "Notation in the Context of *Quintet.net* Projects." *Contemporary Music Review,* vol. 29. no. 1, pp. 39–53, 2010.

[4] C. Hope, L. Vickery. "The Decibel Scoreplayer: A Digital Tool for Reading Graphic Notation." *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015.

[5] R. Gottfried, G. Hajdu. "DrawSocket: A Browser Based System for Networked Score Display." *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'19)*, Melbourne, Australia, 2019.

[6] J. Bell, B. Matuszewski. "SmartVox – A Web-Based Distributed Media Player as Notation Tool for Choral Practices." *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, A Coruña, Spain, 2017.

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

[8] G. Hajdu, N. Didkovsky. "MaxScore: Recent Developments." *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'18)*, Montreal, Canada, 2018.

[9] A. Agostini, E. Daubresse, and D. Ghisi, "*cage*: a high-level library for real-time computer-aided composition", *Proceedings of the International Computer Music Conference (ICMC'14)*, Athens, Greece, 2014.

[10] D. Ghisi, C. Agon, "Real-Time Corpus-Based Concatenative Synthesis for Symbolic Notation", *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'16)*, Cambridge, UK, 2016.

SCORE OBJECTS IN OM#

Jean Bresson
Ableton, Berlin
STMS lab: Ircam, CNRS, Sorbonne Université, Paris[1]
jean.bresson@ircam.fr

ABSTRACT

This paper is an overview of the new score objects and editors available in the OM# visual programming and computer-assisted composition environment.

1. INTRODUCTION

OM# [1] [2] is a visual programming and computer-assisted music composition environment derived from OpenMusic [2]. As a computer-assisted composition environment, its main purpose is to provide composers with programming tools allowing them to implement models for the generation, transformation, representation or synthesis of musical material. The visual language is a comprehensive and general-purpose graphical interface on Common Lisp, using the "patching" metaphor to assemble function calls and data structures in functional graphs.

The possibility to manipulate and visualize musical data structures within visual programs, using music notation in particular, is a key element to make such environment an effective compositional framework. Musical data containers and editors can be used as input, output, and for the storage, display and manipulation of intermediate material and results in compositional processes. They enable a specific workflow that contributes to set computer-assisted composition (CAC) beyond so-called "algorithmic composition" systems [3].

OM# was inially developed in the context of research projects aiming at extending the possibilities of OpenMusic in the domains of interaction, time structures and sound spatialization [4, 5, 6]. The early prototypes of the visual language – as presented for instance in [1] – did not yet include any support for scores and "traditional" music notation. We are now a few years later, and a fairly complete score object framework is available (see Figure 1). [3] This paper gives an overview of this framework.

[1] This work was partially carried out while the author was at Ircam STMS laboratory.
[2] https://cac-t-u-s.github.io/om-sharp/
[3] The project was also renamed OM# in the meantime since these early prototypes.

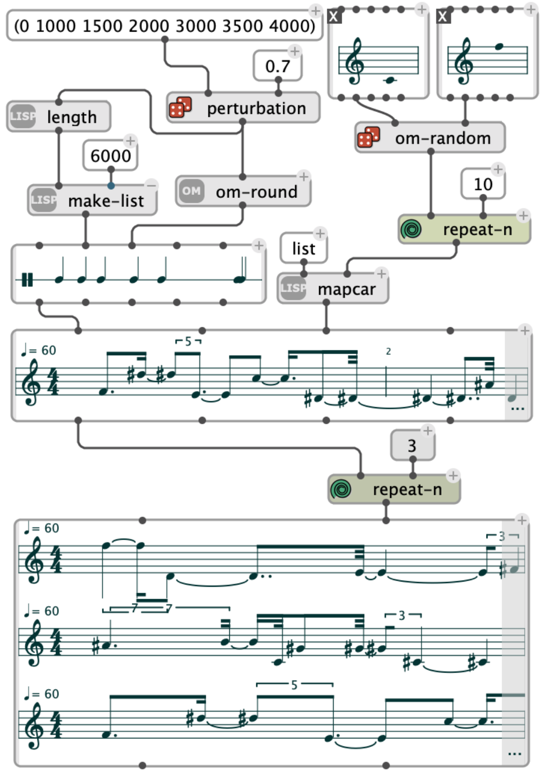


Figure 1. Score objects in an OM# visual program.

2. SCORE OBJECTS – BASICS

The updated score object framework in OM# inherits the core functionality from OpenMusic/Patchwork [7], and is structured hierarchically as follows:

A NOTE is defined by a pitch, a velocity and a duration, as well as complementary information about MIDI channel and port numbers.

A CHORD is a set of one or several NOTEs.

CHORD-SEQs and VOICEs are sequences of CHORDs, where time-positions are represented respectively as absolute onsets or as rhythmic proportions (see Section 3).

MULTI-SEQ and POLY are polyphonic objects which contain superimposed CHORD-SEQs or VOICEs, respectively.

Figure 2 shows an overview of these different objects. [4]

[4] From the user interface point of view, CHORD, CHORD-SEQ and VOICE objects' read/write accessors gather pitches, velocities, durations, etc. as separate lists of values, offering an orthogonal approach to this underlying hierarchical structure.

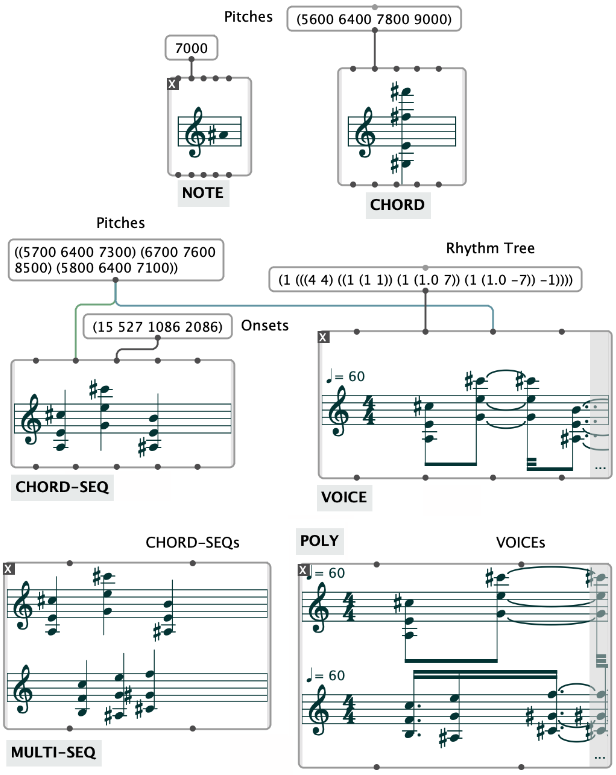


Figure 2. OM# main score objects.

3. RHYTHMIC STRUCTURES AND NOTATION

Each score object has an explicit or implicit *onset*, determining its time positioning relative to its container object. CHORDs are the actual basic element (also called "timed-item") in OM#'s time representation framework [8].

In order to give account for rhythmic notation, the VOICE object represents time structures using an additional layer of MEASUREs and GROUPs, overlaid on top of the sequence of chords (a GROUP contains nested sub-GROUPs, CHORDs, or RESTs).

The temporal/horizontal spacing algorithm handles graphical alignment between simultaneous events and symbols in polyphonic scores, taking into account the constraints of rhythmic structures (where horizontal space is not necessarily proportional to durations), beaming, artificial spacing introduced by bars, keys, alterations and other symbols. It offers a few different scaling options, and an alternative "proportional" representation of rhythmic structures (i.e. spacing proportionally to the actual duration of musical events) (see Figure 3).

From the user perspective, the rhythmic structure is expressed using a Rhythmic Tree (RT): a recursively nested (*duration* (*subdivisions*)) pair (where each element in *subdivisions* is another RT) representing relative durations and grouping in a compact textual format [9]. Rests are encoded by negative durations, and tied notes by float values. The RT is then internally converted to onsets of the chord sequence (considering a given tempo), and determines the rhythmic layer of measures and nested GROUPs, displayed with adequate beaming, tuplets, note heads and dots depending on the given metrics.

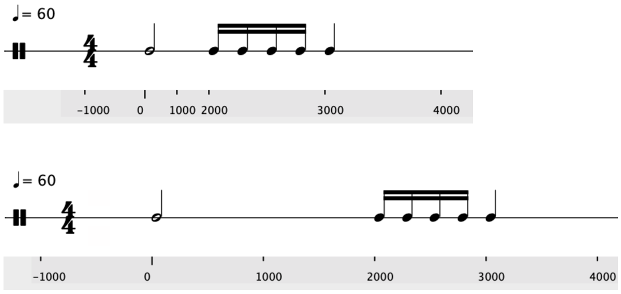


Figure 3. Options for time-spacing of a VOICE: rhythmic (top) vs. proportional (bottom).

OM# supports *grace notes*: a subdivision of 0 in a rhythmic tree group is interpreted as a note shifted by a small offset before or after the closest non-null subdivision in that group (depending on the relative position in the list, and on possible other grace notes aligned in a sequence before or after the "main" note of the group) – see Figure 4. Grace notes are displayed and editable in CHORD, VOICE and POLY editors.

Figure 4. Grace notes: Distribute chord pitches before and after the beat. The group of subdivisions (0 0 1 0) in the rhythm tree indicates that the first two notes are offset before, and the last one after the beat position of the chord.

4. DISPLAY AND EDITING

OM# score rendering follows the SMuFL (Standard Music Font Layout) specification [5] [6] [10], and so theoretically can adapt to any SMuFL-compliant musical font. [7]

[5] `https://www.smufl.org/`
[6] `https://w3c.github.io/smufl/gitbook/`
[7] OM# currently uses the *Bravura* font.

Smooth and continuous zooming in/out gesture and rendering is allowed by a precise positioning of musical glyphs on the staves following this layout specification.

Various options for displaying musical parameters like duration, velocity, MIDI channel and port using musical symbols, numeric values, note heads, color and opacity allow complementing the basic score information (see Figure 5). Picking and editing these parameters is enabled at the level of the NOTE, CHORD, and VOICEs.

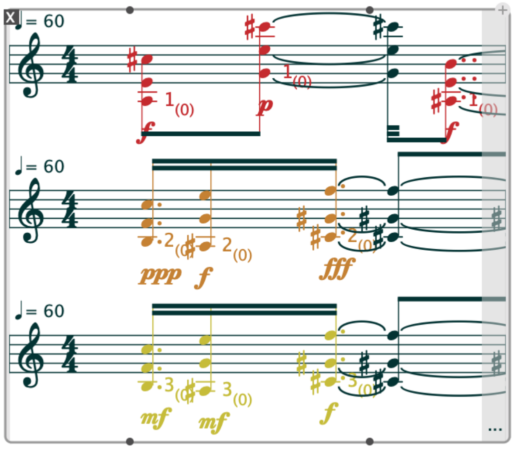


Figure 5. Coloured and extended score display including velocities and MIDI ports/channels.

5. EXTRAS

EXTRA objects are also inherited from the OpenMusic score framework: they represent additional elements that can be added to the score, although not used for actual (MIDI) rendering. The EXTRA objects currently available include texts, symbols (glyphs from the musical font), altered note heads (e.g. square heads, etc. – also with any glyph from the font), velocity symbols (marking the velocity for some specific chords or notes in the score), as well as labelled markers for score segmentation and annotation (see Section 6). They can be attached to any element of the score objects' internal hierarchy (NOTEs, CHORDs, GROUPs, etc.)

In OM#, EXTRA objects can be set and manipulated as lists (or lists of lists, etc.) directly through optional inputs of the score object boxes (see Figure 6).

6. GROUPING AND SEGMENTATION

The *time markers* in score editors can be used for extracting, processing, reordering or applying arbitrary functions to delimited score segments, implementing a simple and versatile version of the "segmentation framework" introduced in [11]. Figure 7 illustrates an application of the *map-segments* function used along with *omquantify* to perform the piecewise rhythmic quantification [8] of a segmented CHORD-SEQ object.

[8] Conversion to VOICE by a translation of sequences of durations into Rhytmic Trees.

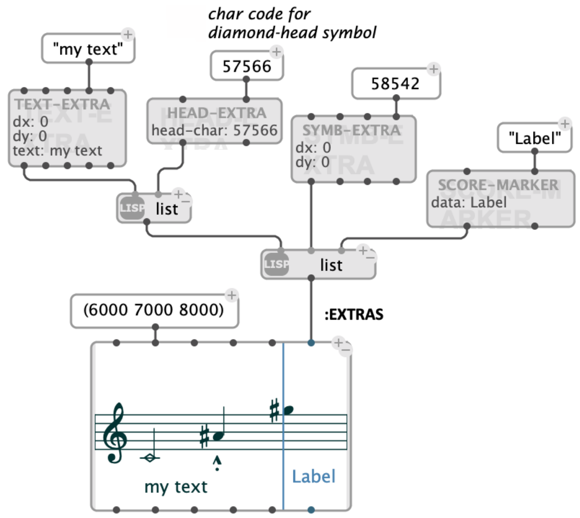


Figure 6. EXTRAS: setting additional score components. The first chord has an attached text and special note-heads, the second chord has another attached symbol, and the third chord has a labelled marker.

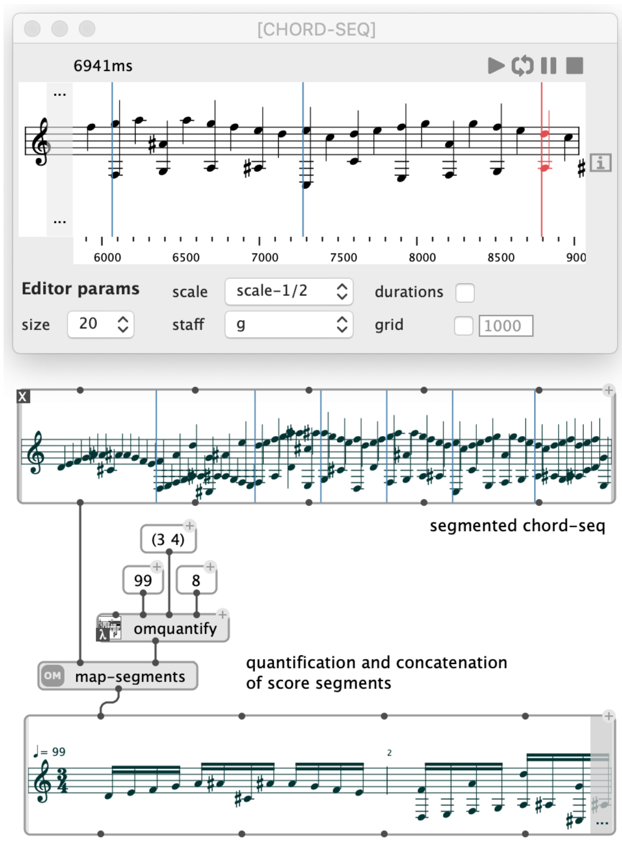


Figure 7. Segmentation and piecewise rhythmic quantification using score markers and *map-segments*. Markers can be added algorithmoically as "score extras" or manually in the editor. Score segments delimited with markers are quantified one by one, and the results concatenated in *map-segments*.

The score editor features additional utilities for grouping score elements: selected groups identified by unique IDs can be displayed or processed either internally (in the editor) or externally (in visual programs). They can also constitute the basis for score analysis models: a basic pitch-class set analysis comes inbuilt in the chord-seq editor as a simple example, using the N-CERCLE object representation [12] for selected pitch sets (see Figure 8). [9]

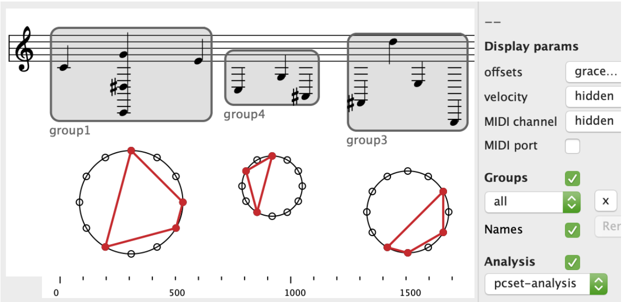


Figure 8. Grouping and analysis inside the CHORD-SEQ editor: pitch-class set analysis.

7. PROGRAMMABLE EDITOR PARAMETERS

OM# score object boxes include optional inputs (and outputs) making it possible to set (and respectively, to read) attributes which do not belong to the contained / generated score object, but to the box and editor, and determine how the score will be rendered in it. Such attributes include the staves (*G*, *F*, *GF*, etc.), the scale (diatonic, 1/4th tones, etc.), or the musical font size.

Figure 9 shows these parameters manipulated in an OM# visual program.

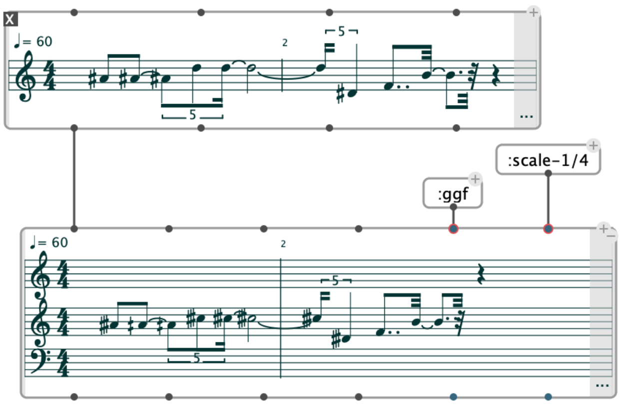


Figure 9. Setting score display parameters in OM# visual programs: staff configuration and (quarter-tone) scale.

8. SCORES AS REACTIVE INTERFACES

OM# features an embedded "reactive" extension, enabling programs to run and update the score and other data containers and editors as a response to user interaction and external incoming events [13].

Score objects in this context, in addition to dynamically displaying updated states and result of visual programs, can act as interactive controllers, propagating user inputs in downstream data processing.

The NOTE object box is implemented as a *slider* UI box (also called *Interface Box*) so that clicking and dragging on it dynamically updates the pitch of the stored NOTE value (according to the mouse position on the staves), as well as any downstream-connected parts of the visual programs, if adequate box connections are set reactive (see Figure 10).

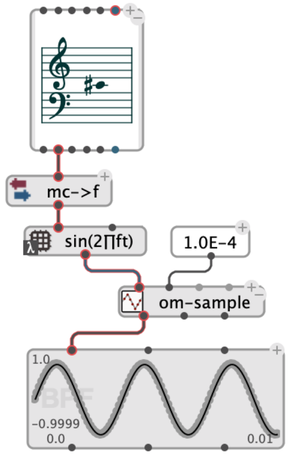


Figure 10. Using the NOTE box as input controller (*slider*) to interactively parametrize and fine-tune a visual program (here, a sine-wave generation algorithm). Reactive inlets/outlets and patch-cords are highlighted in red.

The interaction with score objects enabled with this reactive model is close to the one experienced by users in Max [14] (in particluar using the *bach* framework [15]), with the main differences that: (1) Reactive data-flow is simulated – while evens and notifications are "pushed" in the data-flow graph, the execution implements a pull-based model; and (2) Graphs are only *locally* reactive (where connections between boxes are explicitly set so), which allows the user to control the computation load and frequency in response to changes, and to mix reactive data flow with pull-based evaluation of the visual programs.

9. CONCLUSION

The score representation and editing features presented in this paper contribute to make of OM# an operational and effective framework for computer-assisted music composition today.

The more tangible improvements of this framework are at the level of display and interaction with score elements, facilitated by a renewed rendering framework. The inclusion of editor parameters in visual programming is also a new concept permitted in OM#, and the representation of grace notes is another notable increment.

However, some of these features are still incomplete and constitute the object of future work: at the time of this writing, rhythmic structure editing is still limited (as compared to OpenMusic, for instance), as well as support for tempo changes and variations or micro-tonality beyond 1/4 and 1/8th tones.

[9] Grouping and segmentation features presented in this section are available since OM# v1.3.

10. REFERENCES

[1] J. Bresson, D. Bouche, T. Carpentier, D. Schwarz, and J. Garcia, "Next-generation Computer-aided Composition Environment: A New Implementation of OpenMusic," in *Proceedings of the International Computer Music Conference (ICMC)*, Shanghai, China, 2017.

[2] J. Bresson, C. Agon, and G. Assayag, "OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research," in *ACM MultiMedia'11 (OpenSource Software Competition)*, Scottsdale, USA, 2011.

[3] G. Assayag, "Computer Assisted Composition Today," in *1st Symposium on Music and Computers*, Corfu, Greece, 1998.

[4] D. Bouche, J. Nika, A. Chechile, and J. Bresson, "Computer-aided Composition of Musical Processes," *Journal of New Music Research*, vol. 46, no. 1, 2017.

[5] J. Bresson, J. MacCallum, and A. Freed, "o.OM: Structured-Functional Communication between Computer Music Systems using OSC and Odot," in *FARM: Workshop on Functional Art, Music, Modeling & Design – Proceedings of the ACM SIGPLAN International Conference on Functional Programming (ICFP)*, Nara, Japan, 2016.

[6] J. Garcia, T. Carpentier, and J. Bresson, "Interactive-Compositional Authoring of Sound Spatialization," *Journal of New Music Research*, vol. 46, no. 1, 2017.

[7] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue, "Computer Assisted Composition at IRCAM: From PatchWork to OpenMusic," *Computer Music Journal*, vol. 23, no. 3, 1999.

[8] J. Garcia, D. Bouche, and J. Bresson, "Timed Sequences: A Framework for Computer-Aided Composition with Temporal Structures," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*, A Coruña, Spain, 2017.

[9] C. Agon, K. Haddad, and G. Assayag, "Representation and Rendering of Rhythmic Structures," in *Second International Conference on Web Delivering of Music*, Darmstadt, Germany, 2002.

[10] D. Spreadbury and R. Piéchaud, "Standard Music Font Layout (SMuFL)," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*, Paris, France, 2015.

[11] J. Bresson and C. Pérez-Sancho, "New Framework for Score Segmentation and Analysis in OpenMusic," in *Proceedings of the Sound and Music Computing conference (SMC)*, Copenhagen, Denmark, 2012.

[12] M. Andreatta and C. Agon, "Implementing Algebraic Methods in OpenMusic," in *Proceedings of the International Computer Music Conference (ICMC)*, Singapore, 2003.

[13] J. Bresson and J.-L. Giavitto, "A Reactive Extension of the OpenMusic Visual Programming Language," *Journal of Visual Languages and Computing*, vol. 25, no. 4, 2014.

[14] M. Puckette, "Combining Event and Signal Processing in the MAX Graphical Programming Environment," *Computer Music Journal*, vol. 15, no. 3, 1991.

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

MULTI-SCALE ORACLE AND AUTOMATED REPRESENTATION OF FORMAL DIAGRAMS BASED ON THE COGNITIVE ALGORITHM

Joséphine Calandra
IReMus, LaBRI
josephine.calandra@labri.fr

Jean-Marc Chouvel
IReMus
jeanmarc.chouvel@free.fr

Myriam Desainte-Catherine
LaBRI
myriam@labri.fr

ABSTRACT

This article deals with the automatic representation of formal diagrams, which corresponds to a paradigmatic analysis of the musical work which is being listened to. These diagrams represent musical materials as a function of time and are initially obtained from the audio signal, applying a *Cognitive Algorithm*. In this article, we focus on the second step of the algorithm, as such we assume that the first step, analyzing and labeling the audio signal, has been done. Thus, we propose to analyze predefined materials given as a string. Then we develop the automatic creation of all the formal diagrams of higher levels that result from it. The structuration of the sequences of materials of the lower level constructs the formal diagrams of higher levels. The structured characters which are gathered then represent a higher-level material. Therefore, we present the Multi-scale Oracle: a data structure that stores and connects the different levels and materials. Thus, a character string given as input of the system produces a superposition of formal diagrams as a function of various structuring parameters. As the hierarchical formal diagrams offer a new representation of music, we suggest the musicologists could use these diagrams for analysis.

1. INTRODUCTION

Our research consists of performing a music analysis as a cognitive process accurately related to this specific music as a phenomenon in time. Therefore, we seek to model the representation of the temporal evolution of a piece of music according to the different cognitive imperatives of people who are listening to music. As it has been demonstrated by François Delalande in [1], cognitive imperatives induce the way people listen to music and therefore structure music. By offering in this method of analysis a different representation, thus a different than commonly used reading, we want to understand how music creations through their audio representation are segmented by the listeners and therefore procure emotions to them (G. Brelet [2], J. Sloboda [3]). Listening systems and representations over time, such as OSSIA Score [4] and Antescofo [5], exist but do not offer a structured representation. On the other hand, the structure estimation of pieces of music by multi-criteria analysis and regularity constraint developed by G. Sargent in his thesis [6] gives hierarchical representations that do not consider the temporal phenomenon.

In previous work, one of the authors [7] developed the *Cognitive Algorithm*: a listening model which aims at modelizing the cognitive processes that are set up while listening to music. Nevertheless, this modelization is a hypothetical one that structures the music listener's audio representation to be as faithful as possible to the *Gestalt Theory*. By *structure*, we mean the action of segmenting in a way that gathers objects together to constitute a unique object of higher level. The output is a representation of formal diagrams at different structuring levels of distinct time scales. Each formal diagram gives a view of the music constituents' temporal evolution, called *objects* that are the elements analyzed as instantiated paradigms. We draw a line between the *object* and the *material* as the latter is the actual paradigm. For instance, the deployment of new materials is highlighted by the discovery front: the line representing the most recent new materials at each instant t. Formal diagrams also highlight behaviors such as pattern reiteration, reversion, or speed. From a given-structural-level diagram, and in parallel with the construction of this diagram, we can create the higher-level one by using segmentation and similarity criteria and, therefore, grouping lower-level materials to create a higher-level material. Moreover, the analysis and then the formal diagram representation is not specific to the note scale but applicable to different temporal scales and different sound dimensions. These are determined by the similarity and segmentation criteria provided as a parameter of the system and the constitution of lower-level diagrams.

Besides, this representation is different from scores or piano-rolls as it changes the ordinate axis and highlights the temporal appearance of the materials and not the pitches, which offers another reading for pieces of music. A form of compression is also obtained as the material memory increases as we listen to music. Then, to get an accurate representation, we do not need to know all the materials when processing the music representation. Therefore, there is a need to determine the similarity and segmentation criteria that will be put in place to run a configurable algorithm as generic as possible to obtain as many representations as possible perceptions of music. In this article, we want to develop an algorithm that can offer representations that a musicologist would not have been able to create from the angle of his own experience and analysis by automating the *Cognitive Algorithm*. Even if this algorithm has already been tested manually by Jean-Marc Chouvel, our goal here is to automate this analysis in order not only to simplify the task of the musicologist but also to remove one's pre-established knowledge to better understand the composition and auditory processes.

In this article, we focus on the second part of the algorithm, assuming that the audio signal is already labeled into characters. Then we will deal with the analysis of character strings representing the music. F. Lerdahl and R. Jackendoff [8] proposed a theory to structure tonal music with different grouping rules that would bring some tools for the elaboration of segmentation rules in our algorithm.

Memory is where the information is stored and the question of access to the stored elements and memory optimization arises: this is why there is a need to augment the *Cognitive Algorithm* with an accurate representation of the memory. For that, we propose to use the *Factor Oracle* as implemented by C. Wang and S. Dubnov [9] and first presented by C. Allauzen, M. Crochemore, and M. Raffinot in [10].

We seek to create a multi-scale computed representation that gathers different superposed reading levels. Moreover, we aim to do so by systemizing the paradigmatic and syntagmatic analysis optimally. However, we are not interested in the semiotic analysis of these results: we leave this work to the musicologists wishing to analyze the composition put in the system.

In Section 2, we explain how the *Multi-scale Oracle* works, and what data structures we chose to organize the memory of the modelized listening subject. In Section 3 we will explicit the different rules implemented to structure hierarchically the various character strings that are input. Section 4 explains the structuring function implementation and the memory hierarchization, and Section 5 presents applications of the *Cognitive Algorithm* to Wolgang Amadeus Mozart's K545 *Rondo* and Claude Debussy's *Hommage à Rameau* reduced as label strings. Then, we will discuss the results in Section 6.

2. MULTI-SCALE ORACLE

A description of the *Multi-scale Oracle*, the combination of the *Cognitive Algorithm* and the *Factor Oracle*, is provided. We also describe the chosen data structures for memory implementation, where the materials are stored.

2.1 Presentation of the Cognitive Algorithm

The *Cognitive Algorithm* (Figure 1) was developed and applied by one of the authors [11] to propose a methodological algorithm for musical analysis. This algorithm is the process that constructs the different layers of formal diagrams associated with the music supplied as input. This is mainly composed of two tests. A first test is called *Paradigmatic Recognition Test*. It is a comparison between the object currently being listened to and the previously listened to objects. The second test is the *Syntagmatic Recognition Test*. This test validates whether the concatenation

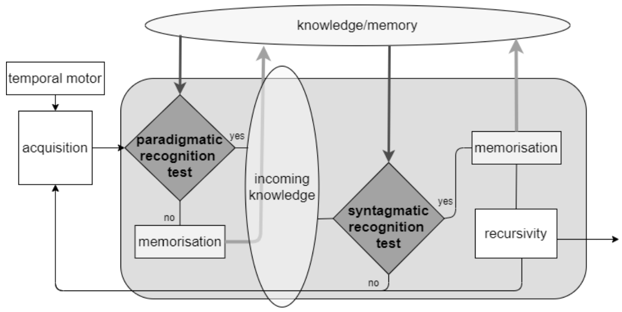


Figure 1: General Scheme of the *Cognitive Algorithm*. For more information, a complete version can be found in [12].

of the objects obtained since the last segmentation constitutes a higher-level object.

The algorithm at a level of structure i is the following. All the objects are processed one by one. At the instant t, the object $obj(i,t)$ is assimilated. The first test is then carried out: if the object which has just been heard is similar to an object previously heard of the same level, then it is recorded in the *formal memory*. The *formal memory* corresponds to the memory where the objects in the time dimension are stored. If this object has never been heard before, it is written in the *formal memory* and also in the *material memory*: the memory that stores every new material which appears over time. Then, the current object $obj(i,t)$ is concatenated in the *incoming knowledge* with previously heard objects of the same level. The assumption is made that a higher level object is being created and the next objects at this level can be guessed according to what has already been processed before. Then comes the second test: if there is at this moment a structuring criterion, then the test is validated and this hypothetical object becomes a higher-level object. The structuring criteria will be clarified in part 3.2. Then, we iterate the same algorithm with this new higher-level object $obj(i+1, t-k)$, where k is the number of lower-level concatenated objects multiplied by their duration, as input. If, conversely, no segmentation criterion is detected, the test is invalidated and the algorithm is reiterated with the next object of the same level.

The difficulty of the computer implementation is that the functions described in this algorithm are general. There is a need to make them mathematically explicit while allowing the system to remain as inclusive as possible because we want to analyze any type of music.

In a previous article [13], we focused on a single level of construction, the first one which is the sampling window scale. We also focused on the first part of the algorithm: the *Paradigmatic Recognition Test* at the signal level. In this article, we leave the signal analysis aside and focus on analyzing from a higher level where materials are already represented by symbols. Therefore the first test corresponds to a character string comparison and the second test corresponds to the structure by segmentation criteria explained in part 3.2.

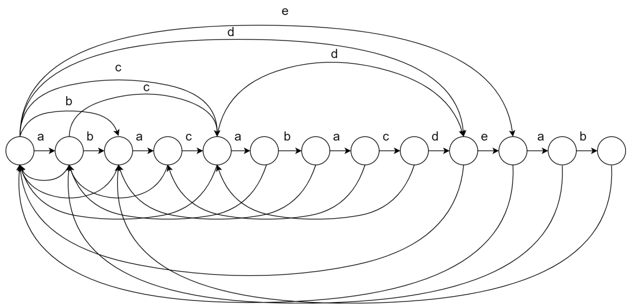


Figure 2: *Factor Oracle* based on the string *"abacabacdeab"*.

2.2 Organization of the memory

The audio is represented by a character string where each character corresponds to a given material. A material can be a declination of any specific musical criterion such as a specific pitch, tone, or even rhythm at different hierarchical levels such as notes scale, grouping notes scale, phrases, or music parts. Therefore the input object for each level of the system is at time t a single character. The memory organization is essential for the system performances because the algorithm performs a constant back and forth, especially with the previously heard objects of the different levels.

2.2.1 Factor Oracle

Objects acquired in the system are represented in memory by an oracle as designed by C. Wang and Shlomo Dubnov [9]. The *Factor Oracle* is a data structure of the form $Q = q_1 q_2 ... q_t ... q_T$ with T states. This oracle enables, thanks to *links*, to quickly find the longest character string previously heard in the system, which is similar to the currently heard character string. There are two types of *links*: *suffix links* and *forward links*. *Suffix links* are created from state $t + k$ to state t with $k > 0$ when the element corresponding to the link q_{t+k} going from state $t + k - 1$ to state $t + k$ and the element corresponding to the link q_t going from state $t - 1$ to t are equals. *Forward links* are of two types: the *internal forward links* correspond to the *links* going from the previous state $t - 1$ to the state t and correspond to the temporal unfolding of the musical work. *External forward links* are created from state t to state $t+k$ with $k > 0$ when the most recent *internal forward link* q_{t+k} is preceded by a q_{t+k-1} *link* such that $q_{t+k-1} = q_t$ and q_{t+k} have never followed q_t before. Each *forward link* is named after the index corresponding to the listened material at that moment, the *suffix links* are not labeled and the states correspond to each temporal moment t of the input. We find an example of a *Factor Oracle* created from the string *"abacabacdeab"* in Figure 2.

For the implementation of the *Factor Oracle*, we use the code developed by C. Wang and S. Dubnov [14].

2.2.2 Relation between the Oracle and the Cognitive Algorithm: the Multi-Scale Oracle

The benefit of using such a memory organization is that there is no need to go through the entire string to find previous similar elements. Searching and comparing elements in the entire string has a $nlog(n)$ complexity. With the *Factor Oracle*, accessing the last similar item is constant once the new state is created. Nevertheless, the creation of the oracle data structure still requires computation time.

Also, the oracle models the *incoming knowledge memory*: indeed, thanks to the *forward links* of the currently heard object suffix, we can make one or more hypotheses about what the oracle expects to follow this object. These hypotheses will be validated or not by *Syntagmatic Recognition Test*. Whether or not these hypotheses are validated provides information on the auditory cognitive behavior by consolidating the music structure or, conversely, by creating a surprise effect.

Nevertheless, the oracle does not compute any information about segmentation, so we need to create rules to justify segmentation and we need to create data structures to store the materials. The *incoming knowledge memory* and the structuring rules need to access the higher-level objects, such as the oracle anticipates the incoming objects at the actual level according to the existing ones in higher levels that contain these actual-level objects. For that, we create different Factor Oracles at each necessary structure level, and we complete those with other data structures that store the information connecting every level. This is what we call the *Multi-Scale Oracle*.

2.2.3 Multi-Scale Oracle data structures description

Besides the oracle, various data structures are added to memorize the materials and to make links between the different hierarchical levels of the formal music analysis. The *Multi-Scale Oracle* is the concatenation of the *Factor Oracle* and the following structures, instantiated as many times as there are hierarchical levels of structure.

The *concatenated object* corresponds to the word currently created in the *incoming knowledge memory* for the higher level. This character string corresponds to the concatenation of all the characters read since the last structure in this level. Each time the string is structured, this character string is reinitialized to the empty word, then each new character corresponding to the material heard is concatenated until it is structured again.

The *links table* associates each current state with the index of the corresponding state of the higher-level oracle. This table is updated at each structure. When structuring, all the characters contained in the *concatenated object*, which correspond to different states at the level of the current oracle, correspond to a single character at the higher level. Therefore as many indices as there are characters contained in the *concatenated object* are added in the *links table*. These indices have the same value, which is the maximum index of the *links table* incremented by 1. This value corresponds to the index of the higher-level state associated with the new structured object.

The *history next table* corresponds to all the higher-level materials obtained until instant *t*. This is an array of pairs (i_label, i_str) where i_label corresponds to the higher-level label and i_str is the corresponding current level structured string. This data structure is optimized because it is updated in constant time and browsed in maximum linear

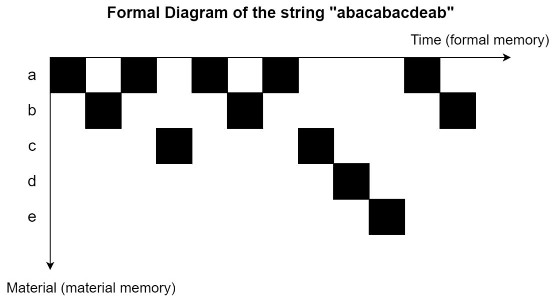


Figure 3: Formal diagram of the string *"abacabacdeab"*.

time meanwhile it contains at most as many elements as the higher level.

The *formal diagram* corresponds to the formal representation of the current state of the oracle at the current hierarchical level. This diagram is a matrix of size $n * t$ where n is the number of materials and t is the current time. For each index $M_{i,j}$, with $0 \leq i < n$ and $0 \leq j < t$ we have $M_{i,j} = 1$ if at instant j the material i is parsed, $M_{i,j} = 0$ otherwise. An example of a formal diagram created from the string *"abacabacdeab"* is shown in Figure 3.

3. CREATION AND IMPLEMENTATION OF THE TWO PARADIGMATIC AND SYNTAGMATIC RECOGNITION TESTS

In this section we present how the different rules of the *Paradigmatic* and *Syntagmatic Recognition Tests* were determined and implemented.

3.1 Paradigmatic Recognition Test

The first test compares the current object as strings with the previously heard objects. At the moment, this is a module verifying the strict equality between two strings. Of course, we intend to improve this part in the future because strict equality between two objects considered to be similar in a musical piece is rare. Two patterns considered as similar can, in reality, have a variation of note, rhythm, tempo, nuance, or even transposition, which can be transcribed as a different character at a lower level and thus also impact the recognition of the pattern at higher levels.

For example, the patterns *C-C-G-G-E-E-C-C* might be considered similar to *C-G-E-C* for the musicologist while the strict comparison of the associated strings *aabbccaa* and *abca* (with *a* associated to *C*, *b* to *G*, and *c* to *E*) gives a negative result for sure. The same problem will arise with time expansion or reduction of the same pattern, or *nota cambiata* that might appear in some patterns. The question of computing similarity will then arise in later studies.

3.2 Syntagmatic Recognition Test

We set up different structuring rules for the second test. Each rule was developed according to the constraints imposed by the previously implemented ones. We searched the most basics rules considering the more complex F. Lerdhal and R. Jackendoff generative theory [8], based on the Gestalt theory. When the second test is raised, these rules are applied or not according to the current context: if the rule is applied, the test is validated, otherwise, it is not.

We suggest five rules. Nevertheless, more advanced rules can be developed in the future.

3.2.1 Rule 1

Rule 1: when an already known object at level n is read again, the level n is structured just before this object. The structured *concatenated object* constitutes a higher-level object.

Here is an example of Rule 1 on the string *"abacabacdeabfgabachijklmhinopqabacrsrsttu"*. ***Str. 0-1*** stands for the structure of level 0 giving a corresponding object at level 1. Level 1 contains the labeling of such structure with strict string equality. Opening parenthesis represents the beginning of a new object at a higher level while a closing parenthesis represents the structuring operation.

Level 0 : abacabacdeabfgabachijklmhinopqabacrsrsttu
Str. 0-1 : (ab)(ac)(a)(b)(a)(cde)(a)(bfg)(a)(b)(a)(chijklm)(h)(inopq)(a)(b)(a)(crs)(r)(st)(tu)
Level 1 : ABCDCECFCDCGHICDCJKLM
Str. 1-2 : (ABCD)(CE)(CF)(C)(D)(CGHI)(C)(D)(CJKLM)
Level 2 : ABCDEFDEG
Str. 2-3 : (ABCDEF)(D)(EG)
Level 3 : ABC
Str. 3-4 : (ABC)
Level 4 : A
Correspondance between level 3 and level 0 gives A= (abacabacdeabfgabachijklmhinopq), B= (a) and C= (bacrsrsttu)

However, we see that this rule alone can give disproportionately sized objects at high levels. It is not consistent in a perspective where one would want objects of the same level to have a similar duration (Grouping preference rules 5 in [8]).

Besides, this structuring system also raises concerns because this rule essentially destroys the principle of hypothesis. If we have, for example, the sequence *"abacab"*, there should be segmentation before the second and the third *"a"* because an already known *"a"* appears again and there is also segmentation of the second *"b"* for the same reasons. This gives the structure *"(ab)(ac)(a)(b)"*. This structure destroys the hypothesis expecting a *"b"* after the arrival of the second *"a"*. Therefore, there is no possibility that the word *"ab"* will come up again because this is what happened at the first occurrence of *"a"*.

When we speak about structuring while an already known object is back, it is for the initial segmentation, in the absence of additional rules, such as those of the Gestalt theory. Once there is a grouping, the structuring must indeed be done in real-time with all the activated levels considering the higher-level object and not layer by layer. Therefore, there is no question of segmenting between *"a"* and *"b"* if there has been a group *"(ab)"* recorded (this can be referred to the Grouping Well Formedness Rule number 4 in [8]).

3.2.2 *Rule 2*

Rule 2 aims at resolving the hypothesis problem: level n is structured only if the object constituted of the *concatenated object* plus the character that has just been read does not exist at the higher level.

Therefore, in the previous example, we do not structure before the arrival of the second b. We then have the structure *"(ab)(ac)(ab)"*.

We must specify then that the segmentation is not done one level after the other but all levels at once. First, the only string we know *"abacabacdeabfgabachijklmhinopq abacrsrsttu"* is read in real-time :
step 1 - **Level 0 :** a
step 2 - **Level 0 :** ab
step 3 - **Level 0 :** aba
There is a *"a"* again, so *"ab"* is structured and a new level is created, containing the symbol *"A"* that corresponds to the label of the lower-level group *"(ab)"*. That gives at step 4:
Level 0 : aba
Str. 0-1 : (ab)a
Level 1 : A
Then the first string is read again until the next structuration. We know *"(ab)"* corresponds to a material of level 1 so we will not structure in the middle of this material at level 0. If there is a structuring criterion at level 1 (for example if A is read again), a level 2 is created and computed in parallel.
Step 12 gives the results:
Level 0 : abacaba
Str. 0-1 : (ab)(ac)(ab)a
Level 1 : ABA
Str. 1-2 : (AB)A
Level 2 : A

The string *"abacabacdeabfgabachijklmhinopqabacrsrsttu"* structured with the rules 1 and 2 gives the results :
Level 0 : abacabacdeabfgabachijklmhinopqabacrsrsttu
Str. 0-1 : (ab)(ac)(ab)(acde)(abfg)(ab)(achijklm) (hinopq)(ab)(acrs)(rst)(tu).
Level 1 : ABACDAEFAGHI
Str. 1-2 : (AB)(ACD)(AEF)(AGHI)
Level 2 : ABCD
Str. 2-3 : (ABCD)
Level 3 : A

Nevertheless, in the string *"abacabacde"*, the return of the pattern *"ac"* represented by *"B"* at level one is hidden in a larger pattern *"acde"* because the new object *"d"* does not allow to structure after the second occurrence of *"ac"*: the structure is then *"(ab)(ac)(ab) (acde)"*. Then, we propose Rule 3

3.2.3 *Rule 3*

As soon as the *concatenated object* at level n is an object already seen at the upper level, the string is structured (Grouping Well Formedness Rule number 5 in [8]).

This rule complements Rule 4, which, however, requires additional computations.

3.2.4 *Rule 4*

There is a structuring operation when the *concatenated object* is a string that has already been heard even included in a larger higher-level object. It is then also necessary to structure the element when it was seen for the first time and to modify the objects of higher levels accordingly.

The string *"abacabacdeabfgabachijklmhinopqabacrsrsttu"* structured with the four rules gives the results :
Level 0 : abacabacdeabfgabachijklmhinopqabacrsrsttu
Str. 0-1 : (ab)(ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hi)(jklm) (hi)(nopq)(ab)(ac)(rs)(rs)(t)(t)(u).
Level 1 : ABABCADABEFEGABHHIIJ
Str. 1-2 : (AB)(AB)(C)(AD)(AB)(EF)(EG)(AB)(H)(H) (I)(I)(J)
Level 2 : AABCADEAFFGGH
Str. 2-3 : (A)(A)(BC)(A)(DE)(A)(F)(F)(G)(G)(H)
Level 3 : AABACADDEEF
Str. 3-4 : (A)(A)(B)(A)(C)(A)(D)(D)(E)(E)(F)
Level 4 : AABACADDEEF

The difference between Rule 3 and Rule 4 in this example is the structure of *"(hi)"* with Rule 4 while Rule 3 would have given the two objects *"(hijklm)"* and *"(hinopq)"*. However, immediately repeated objects are immediately structured as higher-level materials with this rule. This induces isolation of these materials which can no longer be included in a larger material. This also implies that all the higher levels will contain exactly these same materials of this given length.

Also, the structuring algorithm ends, not because we can no longer structure, but because there is a loop and the same structures are obtained at the highest levels.

3.2.5 *Rule 5*

To avoid this material isolation, Rule 5 is therefore proposed: an isolated object is not structured (Grouping Preference Rule 1 in [8]).

The string *"abacabacdeabfgabachijklmhinopqabacrsrsttu"* structured with the five rules gives the results :
Level 0 : abacabacdeabfgabachijklmhinopqabacrsrsttu
Str. 0-1 : (ab)(ac)(ab)(ac)(de)(ab)(fg)(ab)(ac)(hi)(jklm) (hi)(nopq)(ab)(ac)(rs)(rs)(ttu).
Level 1 : ABABCADABEFEGABHHI
Str. 1-2 : (AB)(AB)(CAD)(AB)(EF)(EG)(AB)(HHI)
Level 2 : AABACDAE
Str. 2-3 : (AAB)(ACD)(AE)
Level 3 : ABC
Str. 3-4 : (ABC)
Level 4 : A

In this case, structured elements of roughly equivalent length are obtained at each level. However, this rule might not be efficient on the one hand because the isolated elements are integrated into the following grouping whereas it might be more relevant to integrate them into the previous grouping

Algorithm 1: Structuring_function(level n=0, marker m=0)

Result: Every hierarchical level structured with the associated formal diagrams

```
if level n does not exist and marker m == 0 then
    oracle_n ← initialization(oracle_n);
    links_n ← [0];
    history_next_n ← tab(empty);
    formal_diagram_n ← matrix(empty);
    concat_obj_n ← string(empty);
end
while parsing string s do
    c ← parsed(s);
    oracle_n ← add_state(oracle_n, c);
    formal_diagram_n(c, t) ← 1;
    display(formal_diagram_n);
    if (RULES are passed and m == 0) or
    (m == 1 and len(concat_obj_n) > 0) then
        segmentation(n + 1, m);
    end
    concat_obj_n ← append(concat_obj_n, c);
    if level == 0 and EOS then
        m ← 1;
    end
    if m == 1 then
        segmentation(n + 1, m);
        concat_obj_n ← append(concat_obj_n, c);
    end
end
```

or to keep this object isolated, for example, if it is strongly isolated by a silence.

4. MAIN ALGORITHM AND HIERARCHICAL MEMORY

The *Cognitive Algorithm* is implemented in such a way as to update the *Multi-scale Oracle* and obtain the formal diagrams displayed and updated in real-time.

4.1 Structuring function

The main algorithm consists of calling the *structuring_function(level n, string s, marker m)*. This function is initialized at *level n* (usually set at 0), with the initial *string* s, and marker set at 0. A pseudo-algorithm of the structuring function is provided in the Algorithm 1.

Algorithm 2: segmentation(level n, string s, marker m)

```
t_{n+1} ← max(links_n) + 1;
for i in range(length(concat_obj_n)) do
    links_n ← append(links_n, t_{n+1});
end
s ← label(concat_obj_n);
concat_obj_n ← string(empty);
structuring_function(n + 1, s, m);
```

The *structuring_function* is a recursive function divided into three parts. The first part corresponds to the initialization of the various data structures at the current level if they do not exist. The *oracle* at level n $oracle_n$ is initialized with the initializing function provided in the *Variable Markov Oracle module* developed by C. Wang and S. Dubnov [14]. The *links table* $links_n$ is initialized with a node 0 which correspond to the index of the initial state of the oracle. The *history next table* and the *formal diagram* at level n $history\_next_n$ and $formal\_diagram_n$ are initialized with an empty table and an empty matrix and the *concatenated object* at level n $concat\_obj_n$ is initialized with an empty string. Then the function starts a loop which parses the character string input to the function. It adds in the data structures the new object acquired at this level. First, a state corresponding to the character *c* parsed on the string *s* is added to $oracle_n$ with the adequate function provided in the *Variable Markove Oracle module* [14]. Then the $formal\_diagram_n$ is updated, meaning that the material corresponding to the character *c* is added to the formal diagram if it does not exist yet, and the value of the matrix at material *c* and time *t* is set to 1. Then the updated $formal\_diagram_n$ is displayed.

In a second part, there is a structuring operation if the previous rules are validated for the character *c* at time t (Algorithm 2). The *links table* $links_n$ is updated with the corresponding index of the higher-level node. There is then a comparison of the $concat\_obj_n$ with the already existing top-level objects contained in $history\_next_n$ and *s* become the label at the next level corresponding to the $concat\_obj_n$.

If the object does not exist, $history\_next_n$ is updated with the couple $(s, concat\_obj_n)$ with s as a new label. The *concatenated object* is reset to the empty character string and $structuring\_function(n + 1, s, m)$ is called at the next level with the new labeled object as the input string. At the upper levels, the character string corresponds to a single character *s*, and the function exit and goes back down to the lower level when there is no structuring. If the function is called at the first level of structure, the character string is read as long as there is no structuring. Finally, $concat\_obj_n$ is updated by concatenating the parsed character *c*.

A question is to know when to end the algorithm at higher levels without creating an infinite number of oracles. For this, we use a *marker*. This leads us to the third part: when the last character of the initial character string is read, the marker changes from zero to one and remains at one for each entry in the function at higher levels. Structuring at each level is then enforced. The function $segmentation$ is called again for the last characters of each level so that a new state is not created for this level because it corresponds to the last structuration. Moreover, if the upper level does not exist, it is not created and the algorithm goes directly to the lower level. As all the needed information is provided to the system at this point, the reading and structuring of the remaining characters in the intermediate levels are completed a few steps later.

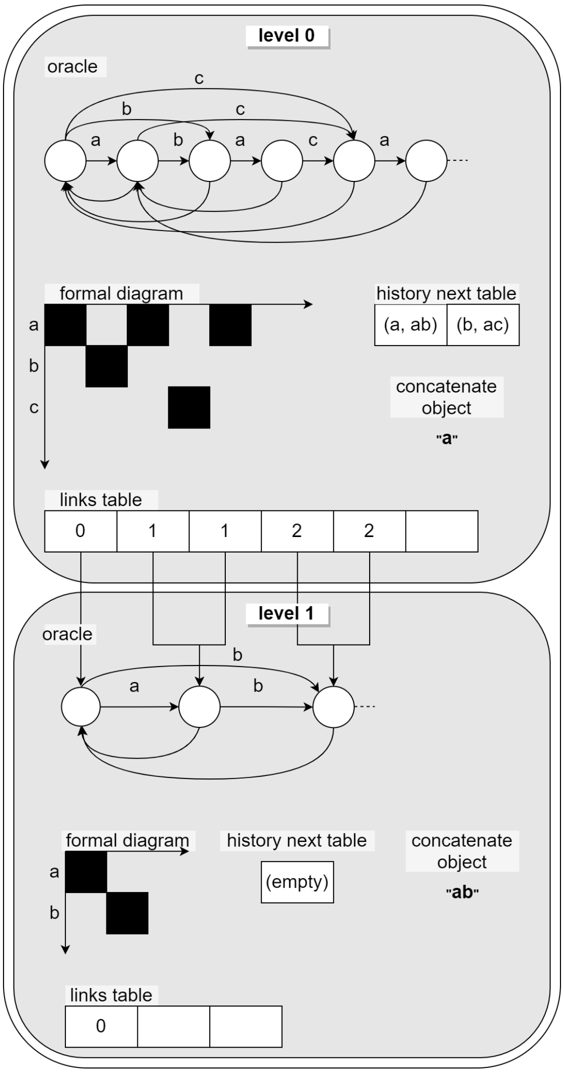


Figure 4: Representation of the *Multi-scale Oracle*.

4.2 Hierarchization of the memory

The organization of the *Multi-scale Oracle* and the back and forth in the memory between hierarchical levels is obtained as follows: the *Multi-scale Oracle* lists for every level *i* a substructure gathering the five data structures described above: a *factor oracle*, a *formal diagram*, a *history next table*, a *concatenated object* and a *links table*. These data structures are updated at any time *t* when new characters are read. Information from higher and lower-level oracles can be accessed through the *links tables*. Relations between the lower-level and higher-level materials are recensed in each *history next* structure. A representation of the *Multi-scale Oracle* is presented in Figure 4.

Each time the lower level is structured, the higher level is created if it does not exist yet and a new state at this higher level is created. The same process is repeated if there is a structuring operation following the creation of this new state. If there is no structuring, the algorithm returns to the lower level where a new state is added, which will lead or not to new structuring operations. If the algorithm is at the lowest level, then it keeps on acquiring new elements of the musical piece, meaning characters from the string.

Figure 5 illustrates schematically the steps of updates and structuration in the *Multi-scale Oracle* according to the five rules with the string *"abacabacdeab"*. First, there is the creation of an *initial state* for the data structures of level 0, meaning the data structures are created and initialized. This is the step (0), the *step of the creation of a new level*. Then, the first character *"a"* is read. A state corresponding to character *"a"* is created in the oracle of level 0 and the *formal diagram* is updated with the material corresponding to *"a"* : this is the step (1), the *step of creation of a state*. There are no structure criteria so anything else happens and *"a"* is appended to the *concatenated object* which was until then the empty word. The same thing is done with the letter *"b"* (2). The third letter *"a"* is read, so a state is added to the according oracle and the *formal diagram* is updated (3). Then, as this letter has already been seen before, there is *structuration* of the *concatenated object* which is *"ab"*. This is the step (4), the *step of structuration*. The next level does not exist yet, so there is the *creation of a new state at the next level*: adequated data structures are created and initialized (5). A new state corresponding to lower-level *concatenated object "ab"*, labelled *"a"*, is created (6) and there are no structure criteria so the loop for level 1 of structure ends, and another character *"c"* is parsed at level 0. The algorithm goes on this way until the end of the string of level 0. The *concatenated object* is structured at states (29), (31), and (33) because the marker is set at 1 since the end of step (28).

5. RESULTS

Examples on W.A. Mozart's *Rondo K.545* and C. Debussy's *Hommage à Rameau* associated strings are provided to illustrate the *Multi-Scale Oracle* analysis based on strings.

5.1 W.A. Mozart's *Rondo K.545*

We take as an example of study the character string *"abacabacdeabfgabachijklmhinopqabacrsrsttu"* which is a reduction of the *Rondo K.545* by W.A. Mozart. We find in Figure 6 the segmentation from the score at the scale of the musical phrase that leads to such a reduction. This reduction was operated manually by the authors and allows us to start on a basis for hierarchical structures automatically computed with the *Cognitive Algorithm*. We did this first analysis from the score to simplify our problem. In the future, the first step will not be made by a musicologist but by a configurable automated audio analyzer giving labels such as what we propose in [13].

We see that the musical phrases labeled with the same letters are simplifications of different variations. What we consider as strict equality during the automatic analysis is then not a real one. The impact of this simplification will be studied when the connection between the signal analysis and the multi-scaled analysis from a character string will be implemented.

As studied in part 4, different results are obtained depending on the rules implemented to structure the piece of music. When we structure according to the five rules mentioned, we find the formal diagrams in Figure 7.

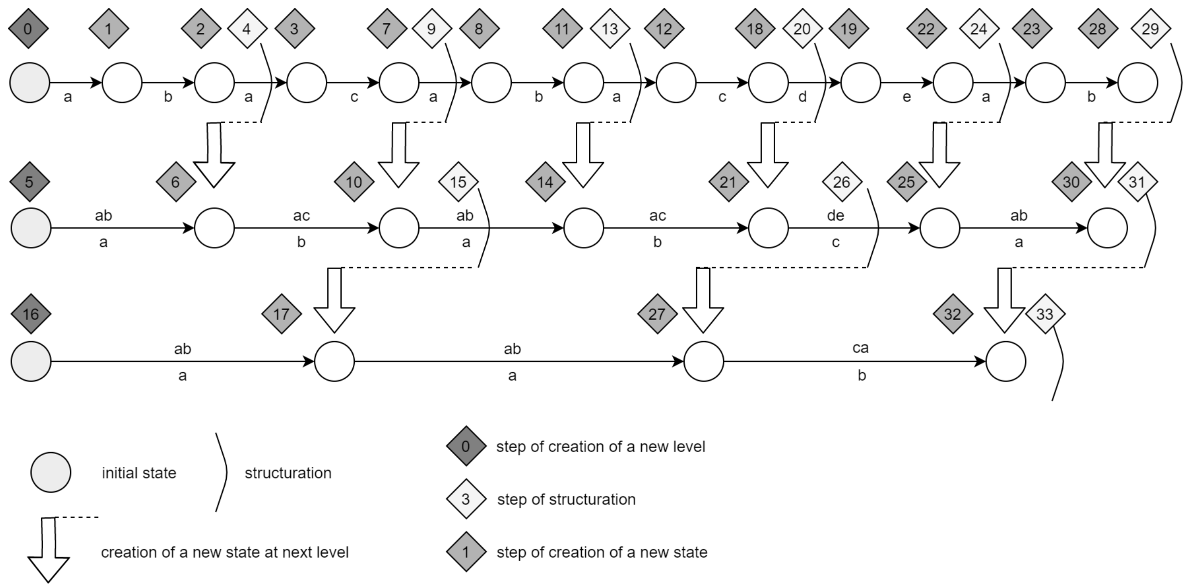


Figure 5: Hierarchization of the memory.

The analysis of these diagrams from a compositional point of view is left to musicologists. However, we can make a few observations about the obtained results. We can compare these diagrams with those obtained when we remove one or more of the stated rules.

If we remove Rule 4 and Rule 5 (see Figure 8) , we end up in the configuration where the diagrams created from a certain level are identical (see part 3.2). So we have as many new diagrams produced as there are oracles created until the end-of-string marker is activated. The termination of the algorithm is therefore ensured, but we have a finite number of identical diagrams and the same number of associated windows for each process that is opened. For example in Figure 4, there are four hierarchical levels, so there are four open windows on the computer's screen that make the results harder to read.

5.2 C. Debussy's *Hommage à Rameau*

We can run the algorithm with any other string as an entry. From an analysis of Claude Debussy's piece of music *Hommage à Rameau* obtained in the same way than the previous example, the associated labels obtained are *(1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1, 2, 3, 4, 9, 20, 21, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 28, 29, 32, 33, 34, 35, 1, 2, 3, 4, 5, 6, 36, 37, 9, 38, 39, 9, 39, 40, 1, 39, 40, 41, 42)*. As there are more than 26 labels, we use numbers instead of letters for easier readability. The associated hierarchical diagrams obtained with the rules 1, 2, 3, 4, and 5 are represented in Figure 9. The *Paradigmatic Recognition Test* computed with strict equality of the strings gives some rigidity to the formal diagrams which might be more accurate with a more flexible similarity test.

6. DISCUSSIONS

We wonder whether the different rules proposed in part 3.2 are necessary and relevant: the study of W.A. Mozart's *Rondo K.545* gives rather efficient results because it has a relatively conventional form, even if we can already see some ambiguities with structures that play between binary and ternary form. However, some pieces of music can lead to more complex and ambiguous structures. For example, in some configurations, the combined rules 4 and 5 induce a significant restructuration of the character string. Therefore there are additional cognitive operations (in the sense of our modelization, meaning higher processing complexity) with the combination of some rules. Indeed, the extract "abacabachijklmabach" is structured in *"(ab)(ac)(ab)(ac)(hijklm)(ab)(ac)(h"* or *"ABABCAB"* which itself is structured *"(AB)(AB)(CAB"*. The parenthesis is not closed at the end of the strings because the analysis is in progress: the last characters correspond to the *concatenated object* and structuring criteria are required for the characters to be structured. When the *"inopq"* objects are concatenated to the previous string there are changes in the structure. In fact, the *"abacabachijklmabachinopq"* character string becomes *"(ab)(ac)(ab)(ac)(hi)(jklm)(ab)(ac)(hi)(nopq"* that is labeled at higher level *"ABABCDABC"* which itself is structured *"(AB)(AB)(CD) (AB)(C"*. Rule 5, which prevented at first glance from structuring before the third *"A"* in level two of the first example, was finally not used because a new character *"D"* appears with the use of Rule 4 in the second example. Moreover, contrary to the example of W.A. Mozart's *Rondo K.545*, the use of Rule 4 sometimes might not involve only the direct higher level but also other levels. In the example presented before, the combination of several rules induces a deeper modification of several higher levels.

However, these additional computations might not be a problem as maybe the cognitive processes themselves consist of recomputing at any time and restructuring the incoming information.

Another question is to know whether these rules are sufficient. For example, we do not take the time signature into account while it can be significant. The character string

Figure 6: Structuration of the W.A. Mozart's *Rondo K.545* from the sheet.

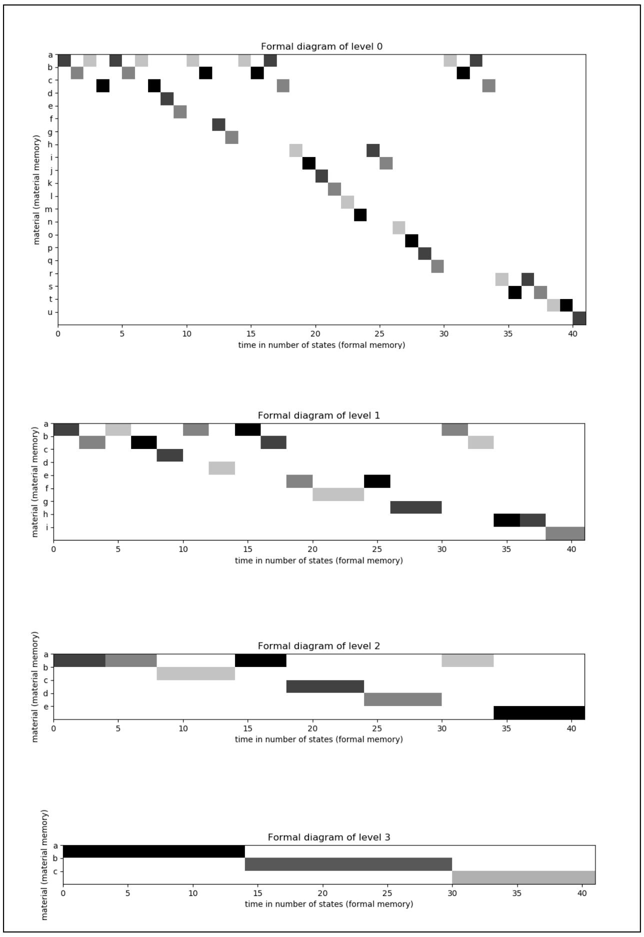


Figure 7: Formal diagrams of W.A. Mozart's *Rondo K.545* at hierarchy levels 0, 1, 2, and 3 with the five structuring rules. The objects have different colors to differentiate two different objects of the same material that are one after the other.

"abacac" may be structured *"(ab)(ac)(ac)"* (binary structuration obtained with the rules we propose in this article) or *"(aba)(cac)"* (ternary slicing). There may even be beat or time signature changes within the same song.

Finally, it is necessary to measure to what extent the implemented rules remain neutral and are not inferred from the knowledge of the musicologist. Adding a time signature, for example, could orient the analysis of a piece where the composer was trying to override its existence in his work.

The computation of similarity between two objects also has to be deepened. Indeed, we want to set up a similarity threshold with a computing distance between two character strings rather than strict equality.

Also, we would like to spell out different functions, such as changes in tempo, nuances, transposition, inversion, and their associated coefficients, to clarify the interconnections between two elements considered to be of the same class. Therefore all the information necessary for the reconstruction of the initial music from a high hierarchical level would be obtained. Moreover, we could represent the presential rate of a previously heard object. For example, if the currently heard object is 80% similar to a previous one, the

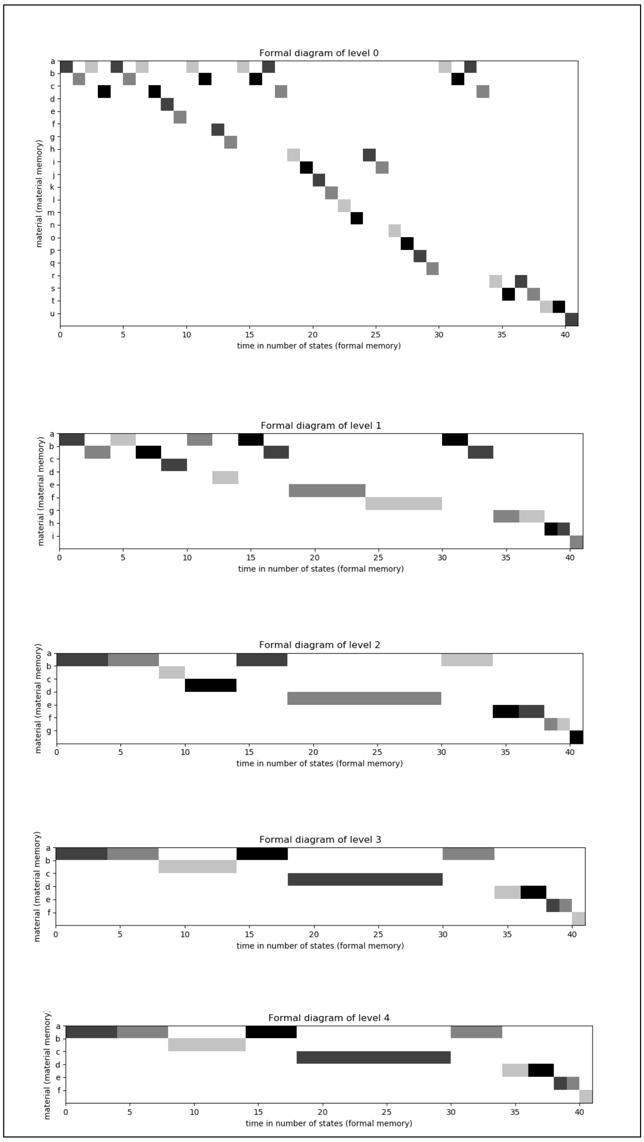


Figure 8: Formal diagrams of W.A. Mozart's *Rondo K.545* at hierarchy levels 0, 1, 2, 3, and 4 with the structuring rules 1, 2, and 3. Six other levels (levels 5, 6, 7, 8, 9, and 10) are created but not shown here as they are identical to levels 3 and 4.

adequate material will be represented accordingly.

Currently, a display is presented in such a way as a window corresponds to a diagram. However, this display is not optimal because there can be a wide range of hierarchical levels meaning a significant number of windows. Furthermore, this display does not align the origins of the different diagrams, while this would visually highlight even more the relations between the different hierarchical levels. In the future, we would like to represent all of these diagrams on a single three-dimensional graph, where the first two axes would be the time axis and the material axis, and the third axis would correspond to the time scale of the structures. It would enable a continuous representation of the structures on different levels. However, we would have to modify our multi-scale oracle, such as it would be less linear for each structuring level. Instead of having one

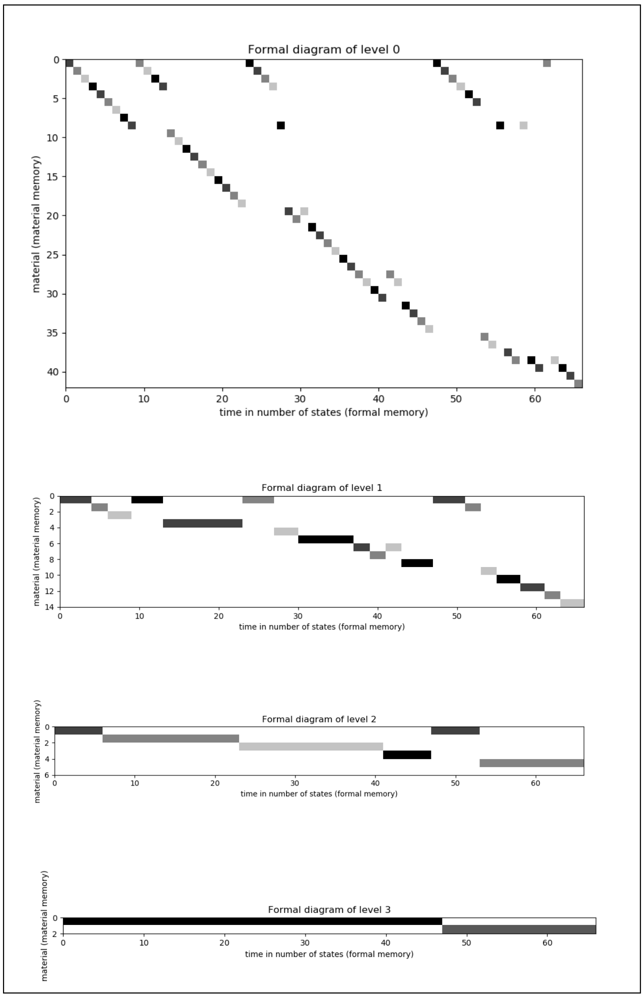


Figure 9: Formal diagrams obtained with the rules 1, 2, 3, 4, and 5 of C. Debussy's *Hommage à Rameau* at hierarchy levels 0, 1, 2 and 3.

augmented *Factor Oracle* at each level, we would have an oracle with multi-scale connexions.

Moreover, at the first level of structuring (from the signal) a representation in gray level for the dynamics of the different objects is implemented, but we still have to integrate it for the different hierarchical levels as there is so far no information on the dynamics with only character strings as input.

7. CONCLUSIONS

In this article, we propose an automatic implementation of a new musical representation. This representation is computed based on the temporal evolution of the piece on different time scales. Then, we suggest an algorithm allowing from a character string, where each character corresponds to a defined musical material, to produce formal diagrams, meaning representations of musical materials as a function of time. These diagrams are structured on different hierarchical temporal levels. Each level is structured in real-time when listening to music, the auditory process being modeled by the parsing of the initial character chain. This automated production of the representation was implemented by the creation of elementary rules defined in this article. We also leave the possibility to choose the *Syntagmatic Recognition rules* to obtain different possible diagrams. Moreover, this article also describes the memory arrangement and the relations between the different levels of memory with the *Multi-Scale Oracle*. Also, we describe and explain the main loop of the algorithm and the trajectory between the different structural levels.

Next, we need to connect the signal analysis of the first level and the entire hierarchical analysis based on string characters. In the first situation, the similarity between objects is computed based on the sampling widows, and the objects are segmented when there are changes. On the opposite, the similarity between strings is calculated based on string comparison, and these are segmented when they are similar. The whole point will now be to create consistency between the two different analyses.

8. REFERENCES

[1] F. Delalande, *La musique au-delà des notes*. Presse Universitaires de Rennes, 2019.

[2] G. Brelet, *le temps musical: essai d'une esthétique nouvelle de la musique, 2 Volumes*. Presses Universitaires de France, Paris, 1949.

[3] J. Sloboda, "Music structure and emotional response: Some empirical findings," *Psychology of Music*, vol. 19, pp. 110–120, 1991.

[4] J.-M. Celerier, P. Baltazar, C. Bossut, N. Vuaille, J.-M. Couturier, and al., "Ossia: Towards a unified interface for scoring time and interaction," in *Proc. Int. Conf. on Technologies for Music Notation and Representation - TENOR 2015*, Paris, France, 2015.

[5] A. Cont, "Antescofo: Anticipatory synchronization and control of interactive parameters in computer music." in *International Computer Music Conference (ICMC)*, Belfast, Ireland, 2008, pp. p.33–40.

[6] G. Sargent, "Estimation de la structure de morceaux de musique par analyse multi-critères et contrainte de régularité," *PhD Thesis, Rennes University*, 2013.

[7] J.-M. Chouvel, "Musical form, from a model of hearing to an analytic procedure," *Interface*, vol. 22, pp. 99–117, 1993.

[8] F. Lerdhal and R. Jackendoff, *A generative theory of tonal music*. Cambridge, Mass. : MIT Press, 1983.

[9] C. Wang and S. Dubnov, "The variable markov oracle: Algorithms for human gesture applications," in *IEEE Multimedia, vol. 22, no. 4*, 2015, doi: 10.1109/MMUL.2015.76., pp. 52–67.

[10] C. Allauzen, M. Crochemore, and M. Raffinot, "Factor oracle: a new structure for pattern matching; ; theory and practice of informatics." in *Proceedings of SOFSEM'99*, Paris, France, 1999.

[11] J.-M. Chouvel, *Analyse Musicale, sémiologie et cognition des formes temporelles*. L'Harmattan, Paris, 2006.

[12] ——, "Categories and representation in cognitive musical analysis," *Sonus*, vol. 35, pp. 17–35, 2014.

[13] J. Calandra, J.-M. Chouvel, M. Desainte-Catherine, and al., "Génération automatique de diagrammes formels par un algorithme cognitif modulaire: étude préliminaire," in *Journées d'Informatique Musicales (JIM)*, Strasbourg, France, 2020.

[14] C. Wang and S. Dubnov, "Guided music synthesis with variable markov oracle," in *3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, Raleigh, North Carolina, 2014.

DISTRIBUTED NOTATION IN THE BROWSER, AN OVERVIEW

Jonathan Bell
Aix Marseille Univ, CNRS, PRISM
Perception, Representations, Image, Sound, Music
Marseille, France
belljonathan50@gmail.com

ABSTRACT

This paper endeavours to discuss a few available technologies for musical notation, from a user's point of view, focusing on "distributed notation": musical notation rendered in real time to multiple devices simultaneously. The rapid evolution of browsers and the cheaper cost of heterogeneous (cross-platform) systems inclines us to narrow down the scope to browser-based solutions. The overview recalls a few key concepts of the JavaScript ecosystem, typically addressed to composers with little background in web development. The survey then compares three frameworks: INScore (INScoreWeb), DRAWSOCKET, and SmartVox, focusing on their respective approach to software architecture/implementation, as well as their "higher" user level. After highlighting the convergence points between INScore and DRAWSOCKET - but also their own specificity, such as INScore's time model, or DRAWSOCKET's API, the paper concludes with a case study: the Tenor 22 choral concert.

1. INTRODUCTION: MUSICAL NOTATION IN THE DIGITAL AGE

1.1 Animated Notation

Numerous Australian composers/academics have composed with and written about digital forms of notation: Vickery sees in animated screen-based notation '*an important solution to visualising a range of musical phenomena and techniques including continuous parametrical changes, synchronisation with prerecorded audio or live processing, and nonlinear formal organisation*' [1]. Hope, citing Winkler, sees animated notation as '*a third way between improvisation and fixed scores*' [2]. In such settings, Wyatt questions the role of the conductor [3] when for instance the timing is dictated by the advancement of a cursor on the page. Kim-Boyle, finally, pioneers research at the intersection between notation and immersive *augmented reality* technologies [4] (see section 4.1.1).

These approaches to notation present a drastic change in the organisation of musical time, which otherwise still follows today principles inherited from the *ars nova*. [1] To paraphrase Vickery, animated notation simplifies the synchronization of human performers to multimedia, as well as the display of generative or interactive scores in which the work can render a different content each time [5], or adapt to the performance of the performer [6]. Among these technologies, three frameworks allow for real-time score display in the browser.

- INScore, which was operational as early as 2012 [7], releases in 2020 its web version (INScoreWeb).
- DRAWSOCKET [8, 9], which elaborates on its authors' former (albeit still actively developed) notational projects [10, 11, 12, 13]
- SmartVox, best described as a web-based distributed media player [14].

Whilst SmartVox is dedicated to a very specific task (the display and synchronisation of mp4 video files), INScore and DRAWSOCKET - which also both supports videos - share with the Decibel Score Player (see [15], 2. The Canvas scoring mode) the ability to render in real time SVG [2] drawing commands via Open Sound Control (OSC) [16] messaging in distributed setups (i.e. on multiple devices).

1.2 Web-Based cases

The present study strictly observes browser based solutions which is why the landmark Decibel Score Player [15], a native iOS application, is not discussed here. Network communication is evidently simplified by web/browser-based technologies, however the Decibel Score Player and a few other projects fully support this key feature of distributed notation, such as: Pedro Louzeiro's *Comprovisador*, which distributes notation in real-time to several clients using *bach* [17] in Max/MSP, communicating through UDP (see section 2.1.3), or Slavko Zagorac's ZScore [18], which is built on the top of the native version of INScore [7].

After a brief overview of the web technologies most commonly used by these notational systems, the paper introduces an non-exhaustive list of recent frameworks/pieces involving browser-based musical notation (Section 3.1), in order to compare a few of their general characteristics (Section 3.2). The paper then mainly focuses on DRAWSOCKET, INScore and SmartVox, which are best known by the author.

[1] The division of large units (*tempus*) into smaller one (*prolatio*), roughly corresponds to today's time signatures, see for instance: https://en.wikipedia.org/wiki/Prolation

[2] https://en.wikipedia.org/wiki/Scalable_Vector_Graphics

2. OVERVIEW OF ASSOCIATED TECHNOLOGIES

2.1 Communication Protocols

2.1.1 HTTP

Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. Riding on the top of TCP, it was designed for communication between web browsers and web servers.

2.1.2 WebSocket

WebSocket is distinct from HTTP. WebSocket is a communications protocol that provides full-duplex communication channels over a single TCP connection. It has become the *de facto* standard for real-time interaction with the browser.

2.1.3 TCP-UDP

TCP is a connection oriented protocol. UDP is a connection less protocol. As TCP provides error checking support and also guarantees delivery of data to the destination router this make it more reliable as compared to UDP. On other hand UDP is faster and more efficient than TCP.

2.1.4 OSC

OSC [16] is a content format developed at CNMAT by Adrian Freed and Matt Wright. Typically intended for sharing music performance data (gestures, parameters and note sequences) between musical instruments either via UDP (most common scenario) or TCP.

2.1.5 odot

OSC developments at CNMAT later led to odot [19] which allows to label data with human readable text (associative arrays). [3] Although discussed in greater detail in later sections, Figure 2 shows the translation operated by DRAWSOCKET, from an OSC bundle (odot) into JSON.

2.2 JavaScript

Javascript is commonly referred to as the language of the web. Originally client-side, Javascript engines are now embedded in some servers, usually via node.js. The most comprehensive documentation about the language can be found at developer.mozilla.org.

2.2.1 JSON

JavaScript's native format, JSON - JavaScript Object Notation - has eclipsed XML and thus became the most ubiquitous data exchange format for any publicly available web service. JSON supports associative arrays as well as ordered lists of values (arrays).

[3] An associative array, map, symbol table, or dictionary is an abstract data type composed of a collection of (key, value) pairs, such that each possible key appears at most once in the collection.

2.2.2 Node.js

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js owes its success to its module exchange platform NPM [4], as well as to the convenience of writing the client and the server in the same language.

2.2.3 Node for Max

The release of Max 8 in 2018 featured a new Javascript support with Node for Max. The Max API module helps interaction and communication with Max from within the Node context. [5]

2.2.4 Frameworks

Node.js is low level, which is why most applications use frameworks to deal with common server features. Routing, most importantly, is used to divert users to different parts of the web applications based on the request made. The Express framework has emerged as one of the most popular framework among Electron, koa, meteor and vue.js to name a few.

2.2.5 WebAssembly

WebAssembly (abbreviated Wasm) is a new type of code that can be run in modern web browsers. Providing languages such as C/C++, C# and Rust with a compilation target so that they can run on the web, the main goal of WebAssembly is to enable high-performance applications on web pages designed to run alongside JavaScript.

3. AN OVERVIEW OF WEB TECHNOLOGIES FOR DISTRIBUTED NOTATION

'*Distributing Music Scores to Mobile Platforms and to the Internet*' [20], an idea idea expressed by Fober in 2015, enjoys a growing interest today, as exemplified by two major frameworks (see table 1) as well as multiple independent initiatives by individual composers/developers (see table 2), all briefly described in the list below.

3.1 Description

3.1.1 INScore

INScore [7] is an environment for the design of augmented interactive music scores, open to unconventional uses of music notation and representation, including real-time symbolic notation capabilities. It can be controlled in real-time using Open Sound Control [OSC] messages as well as using an OSC based scripting language, that allows designing scores in a modular and incremental way. INScore supports extended music scores, combining symbolic notation with arbitrary graphic objects. All the elements of a score (including purely graphical elements) have a temporal dimension (date, duration and tempo) and can be manipulated both in the graphic and time space. The INScore engine is based on a MVC architecture. The abstract

[4] https://www.npmjs.com/

[5] https://docs.cycling74.com/nodeformax/api/module-max-api.html

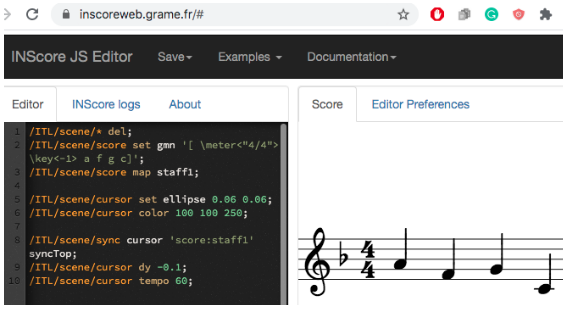


Figure 1. The INScore IDE is available online at : https://inscoreweb.grame.fr/.

model is designed in C++ and can be deployed on any platform (Windows, MacOS, iOs, Android) including the Web where it is compiled as a WebAssembly module (see Section 2.2.5. INScore Web now has an IDE (Integrated Development Environment) available online (see Figure 1, and the INScore Web package will be available on NPM in 2021.

3.1.2 *DRAWSOCKET*

DRAWSOCKET is best described in [8, 9], which is why only references tracing its genealogy are provided here. The roots of the DRAWSOCKET project can be traced back in Hajdu's quintet.net [11], already a landmark piece of software in the realms of distributed notation and networked music performances when the internet was still a young technology. More recent works by Gottfried on 'SVG to OSC transcoding' [13] also reveal some of its premises.

Odot (OSC) Bundle :	JSON object (or Python dictionary)

/violin : {
/key : "svg",
/val : {
/new : "circle",
/id : "bob",
/cx : 100,
/cy : 100,
/r : 20,
/fill : "green"
}
}
	{
	"violin" : {
	"key" : "svg",
	"val" : {
	"new" :"circle,
	"id" : "bob",
	"cx" : 100,
	"cy" : 100,
	"r" : 20,
	"fill" : "green"
	}
	}
	}

Figure 2. Comparison between an odot bundle (OSC), and a JSON object, following DRAWSOCKET's API syntax.

The transcoding of OSC notational commands to JSON (see Figure 2) describes it well, and highlights its potential for versatile real-time notational purposes in the browser. Taking advantage of the odot [19] library's ability to dynamically format and label OSC bundles, the node.js framework is embedded in Max (via Node for Max, see section 2.2.3), but can also run independently from the Max environment [6]. DRAWSOCKET provides the user with an API [7] dedicated to real-time communication with connected clients. As shown in Figure 2, the top level router sends the bundle to a specific client (here the violin). Replacing /violon by /* would send the bundle to all. Then each keyword (/key or "key" in Figure 2) obeys a slightly different - although consistent - syntax : "svg" will draw an svg on the page, "tween" will animate the svg according to its target /id value (in figure 2 the target svg's id is "bob"), "pdf", "sound" and "file" will load the corresponding files (e.g. pdf, mp3 or JSON). The Elbe Tunnel's scores repository provides an apt example for observing DRAWSOCKET in action. [8]

3.1.3 *SmartVox*

Developed at IRCAM in 2016 in the SoundWorks framework (Robaszkiewicz 2015) of the CoSiMa project (ANR-13-CORD-0010), SmartVox [14] consists of distributing and synchronizing mp4 audiovisual scores on the browsers of the performers' smartphones (typically on a network heterogeneous local i.e. on different OS or cross-platform), to help them sing in polyphony, in site specific settings (e.g. moving around the audience), and in a spectral language (i.e. microtonal). Written in Javascript, SmartVox uses the native JSON format to assign each singer a unique file (see Figure the script example below).

```
const score = {
  duration: 20 * 60, // seconds

  // define the different parts
  parts: {
    'soprano-1': {
      file: 'videos/soprano-1.mp4',
    },
    'soprano-2': {
      file: 'videos/soprano-2.mp4',
    },
    // ...
  },

  // define the different sections
  sections: {
    alpha: {
      time: 0,
      label: 'First_section',
    },
    beta: {
      time: 117,
      label: 'Second_section',
    },
    // ...
  },
};
```

3.1.4 *Prolonged Into the Latent (PITL)*

Amongst an ever-increasing number of solutions for web-based synchronised scores, Justin Yang's *Prolonged into the latent* [9] provides a good entry point for considering how two types of clients (the conductor [10] and one of the singers [11]) can communicate via WebSockets (see Section 2.1.2). Yang's notational interface is reminiscent of the *Guitar Hero* video game.

3.1.5 *John*

Goudard's *John, the semi-conductor: A tool for comprovisation* [12] [21], is a distributed notation software '*designed*

[6] See https://drawsocket.github.io/index.html
[7] See: https://drawsocket.github.io/api.html

[8] https://quintetnet.hfmt-hamburg.de/tunnel_webviewer/index.html
[9] The code is available at: https://github.com/elosine/prolonged_into_the_latent
[10] The conductor's interface can be accessed via the following url: https://pitl.justinyang.net/?parts=0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15 &controls=yes
[11] The singer's interface can be accessed via the following url: https://pitl.justinyang.net/?parts=0, pressing 'start' on the conductor will cause the singer's part to start playing. Hitting the ready button on the singer's interface will
[12] Code available at: https://github.com/vincentgoudard/ReactiveJohn

to help collective free improvisation'. The term *Comprovisation* refers to the composer and scholar Sandeep Bhagwati [22].

3.1.6 Anna und Marie

Pirchner [13] [23] developed a real-time score-system for the composition *Anna & Marie* by Marko Ciciliani. SuperCollider sends OSC messages to the dedicated score system generating symbols and playing instructions for each performer, rendering them on tablet displays.

3.2 Comparison

The six aforementioned frameworks will now be comparerd according to the following categories : 1/ Which type of server does the setup rely on. 2/ Client side, does the framework target the web or a specific operating system 3/ Does it have WebSocket support (see Section 2.1.2). 4/ Does it use a framework such as Express (see Section 2.2.4). 5/ Is it a framework, or was the architecture build for a single piece 6/ Does it support sound. 7/ Does it support tradition notation (abbreviated CMN for Common Music Notation) 8/ Does the framework rely on a shared clock/which library does it use from this 9/ Does it support Scalable Vector Graphics. 10/ Does it use pre-existing graphic libraries.

	INScore	Drawsocket
Server	python	node.js
Web/Native	Web, Android, OSX...	Web
Websockets	ws	ws
uses a framework	no	express
is a framework	yes	yes
Sound	faust	tone.js
CMN	Guido	MaxScore Bravura
Sync	native	timesync
SVG support	yes	yes
Graph. lib., animation	qt	tween Three.js

Table 1. Comparison of architectural characteristics of two major frameworks for distributed notation.

3.2.1 node.js/WebSocket

Most importantly, the first two rows of the tables shed light on the strong presence of node.js (see Section 2.2.2) on the server side, together with WebSockets (see Section 2.1.2), today well-known technologies for implementing real-time communications in the modern web. The case of INScore is different in the sense that it privileges self-containedness over server/client architecture: the page can therefore be delivered by servers of any kind such as python, Apache, or node.js.

The second row of Table 2 shows that three technologies use socket.io, a JavaScript library built on top of WebSocket. DRAWSOCKET and some parts of PITL use the express.js framework, *A&M* uses electron (row No 3, discussed in Section 2.2.4).

[13] The code is available here: https://github.com/asa-nerd/Anna-und-Marie

3.2.2 Sound

Regarding their sonic capacities, INScore has the advantage to embed *faust* [24], and therefore promises interesting deployments of DSP and sound synthesis in the realm of *musique mixte*. *PITL* uses the web audio API and DRAWSOCKET uses tone.js, a simple library built on the top of it. For traditional notation (CMN), INScore natively support (and maintains) GUIDO [14] . DRAWSOCKET and MaxScore now share approximately the same API

3.2.3 Synchronisation

DRAWSOCKET and PITL show that *timesync* is today the most commonly used library to, for instance, make sure a message arrives to all clients exactly at the same time, according to a shared clock (Smartvox uses an IRCAM library [25], based on a similar concept).

3.2.4 Graphics and Animation

Whilst most applications use SVG (Scalable Vector Graphics being the prominent solution to define graphics for the web), only INScore, DRAWSOCKET, support SVG rendering in real time. Whilst DRAWSOCKET, as its names states, *draws* SVG on a page, without an intermediary library, it uses GSAP-tween for animation [15] , which can be conceived as 'high-performance property setter'. Regarding the use of external libraries for graphic renderings, no clear convergeance can be found across the six frameworks considered here.

	PITL	A&M	John	SmartVox
Server	node.js	node.js	node.js	node.js
Web/Native	Web	Web	Web	Web
Websockets	socket.io	socket.io	∅	socket.io
uses a fr.	express	electron	Meteor	SoundWorks
is a fr.	no	no	yes	yes
Sound	webaudio	∅	∅	(mp4)
CMN	∅	∅	∅	∅
Sync	timesync	∅	∅	@ircam
SVG support	yes	yes	yes	no
Graph. lib.	Three.js	snap.svg	d3js	∅

Table 2. Comparison of architectural characteristics of a few browser-based frameworks for distributed notation.

After this rapid technical overview of the landscape of real-time/distributed notation, technical considerations as well as user experiences with INScoreWeb, SmartVox and DRAWSOCKET will be discussed in greater detail, in order to highlight what brings them closer and where they diverge.

4. COMPARATIVE STUDY

When compared two by two, the three aforementioned frameworks reveal a few striking similarities (see Section 4.3/Figure 4), but also raise questions when they approach the same domains differently.

[14] https://guidodoc.grame.fr/#welcome-to-guido
[15] Available at: https://greensock.com/docs/v3/GSAP/Tween

4.1 INSCORE and SmartVox

4.1.1 Augmented Reality

The author has documented some of his own artistic works in the realm of AR notation with SmartVox [26, 27, 28]. Emerging works such as [4, 29] let us envisage rapid developments in this field, in which the prescription of a gesture in space should reveal representations far more intuitive (often called prescriptive, or tablature-like, as opposed to descriptive i.e. relying on an abstract system such as the 5-line staff) than those used on paper for centuries, or those developed on animated screens since a few decades.

Although achieved by [30], in which four performers were guided by Hololens, the rapid evolution and ephemerality of such technology lets us favour cheaper smartphone solutions. So far the straightforward method used by the author consisted a simple display above the head of the performer. With such cheap setups, however, holographic display - which requires a different image for each eye - often failed to provide a comfortable display due to calibration issues linked to the size of the performer's phone and length of his/her inter-pupillary distance.

INScore can run as a native Android application, which has given evidence of very promising results on EPSON bt-350 glasses in this domain. INScore's forwarding mechanism [16] allows for the real-time transfer from one INScore instance (on a laptop) to another (on glasses), by writing the following script, in which '$glasses' corresponds to the ip address of the target device.

```
glasses = "192.168.0.26:7000";
/ITL forward $glasses;
```

This proved convenient for debugging purposes, and revealed a comfortable display for the user. The native Android version of INScore enjoyed stable results. The downside of EPSON bt-350 glasses, for browser-based tests, is that it currently fail to load html pages served by DRAWSOCKET, INScoreWeb, or SmartVox.

4.1.2 Sound

Sound may not seem a central feature for softwares dedicated to musical notation. However, it has been demonstrated in [31, 32, 33] that many fruitful artistic works rely on the use of sounds or auditory signals as a score - e.g. a guide track for a singer rather a sound file to be played through loudspeakers. SmartVox, essentially a distributed mp4 player, takes advantage of the HTML video tag, whose embedded media supports audio as well as video.

INScore's browser implementation [34], thanks to the recent release of the Faust NPM package [17] (Faust compiled as a WASM library), currently investigates the possibilities of exploiting the DSP capacities of the web page rendering the score. The feature will therefore allow (among other possibilities) a precise synchronisation between electronic transformations and the score, e.g. the opening of the browser's microphone when the cursor hits a certain note.

[16] Described here: https://inscoredoc.grame.fr/refs/10-forwarding/

[17] https://www.npmjs.com/package/@grame/libfaust

4.2 DRAWSOCKET and SmartVox: cache memory and delay management

4.2.1 Join the performance at any time

Compared to native applications, web pages are fragile: the page can load improperly, a user interaction causes it to misbehave... In such cases, reloading if often required. Then problems may arise if something goes wrong in the middle of a musical performance, and if the current state of the application is not stored in the cache (e.g. which bar in the composition timeline are we in 'now'?). In DRAWSOCKET, the caching system stores automatically all the drawing commands, so that refreshing the page client side keeps trace of all the drawing commands since the server started, or since the page was cleared. When a client is late, a mechanism allows it to catch up delay when a message is received too late by a client [18]. More precisely, for tween animations specifically (see footnote No 15), with the 'cmd' function, DRAWSOCKET will check to see the difference between the start time and the current time (according to the shared clock discussed in Section 3.2.3) and jump ahead if it is late. [19]

One of the strength of SmartVox consists of its ability to update the 'currentTime' (the instant currently displayed in the video) on each server tick: when clients periodically query the shared clock to check whether the drift is not too important [20]. This feature proved to be very robust in several performances [21], if for instance, in the middle of a piece, a problem occurs, and a performer needs to refresh his/her page, or if he/she was not ready when the piece started.

4.2.2 Scheduler

The default behaviour of DRAWSOCKET privileges instant responsiveness over synchronisation, if for instance a sound is triggered to all clients. However, each incoming message is automatically time-tagged according to a shared clock. While writing this article, two different messages ('del' and 'schedule') are being tested, both locally and in remote setup. 'schedule' delays a message of a given value according to the clock sync offset, while 'del' simply delays it. [22] Such features let the user elaborate complex temporal score architectures.

4.3 INSCORE - DRAWSOCKET, similarities

In order to show how, fundamentally, INScore and DRAWSOCKET obey the same kind of OSC-driven drawing commands, both of the following scripts draw a line or rectangle named "cursor", at the top left of the violin part. In INScore the origin is in the center which is why x and y are

[18] See the source code :https://github.com/HfMT-ZM4/drawsocket/blob/master/code/node/lib/drawsocket-client.js, line 947-1069 and 1567-1613)

[19] Reloading the page in the middle of a tween animation: https://youtu.be/2ahjjbS5s2U

[20] See the source code here: https://github.com/belljonathan50/SmartVox0.1/blob/master/src/client/player/PlayerExperience.js, line 153).

[21] e.g. Deliciae, Common Ground, Le temps des nuages, SmartVox... All described in former TENOR publications by the same author.

[22] See the API, in the 'event' keyword section : https://drawsocket.github.io/api.html#event

negative (scaled between -1. and 1.). In DRAWSOCKET the origin is at the top left, and counts in pixels (see Figure 3, the first 3 line in the INScore script, middle column in DRAWSOCKET). To animate the cursor object and thus move it across the score, however, INScore and DRAWSOCKET use different methods, (see Figure 3, line 4 until the end in the INScore script, right column in DRAWSOCKET), which will be detailed in more details in the following section.

INScore & INScore WEB	DRAWSOCKET	

## création du curseur: 	
 	
/ITL/scene/cursor set rect 0.01 0.2;
/ITL/scene/cursor color black;
 	
## création d'une ligne (trajectoire): 	
/ITL/scene/line set line wa 1 0;
/ITL/scene/line x -0.5; ## vers la gauche
/ITL/scene/line y -0.7; ## vers le haut
/ITL/scene/line duration 1; ## 4 secondes
 	
## synchronisation du curseur à la ligne: 	
/ITL/scene/cursor tempo 60 ;
/ITL/scene/sync cursor line syncFrame;
	"// création du curseur"
	
	/violin : {
	/key : "svg",
	/val : {
	/new : "line",
	/id : "cursor",
	/x1 : 200,
	/x2 : 200,
	/y1 : 40,
	/y2 : 80,
	/style : {
	/stroke : "black"}
	}
	}
		"// définition d'un déplacement de 200 pixels vers la droite"
		
		/violin : {
		/key : "tween",
		/val : {
		/id : "score-tween",
		/target : "#cursor",
		/dur : 5,
		/vars : {
		/x : 200,
		/paused : "true"
		}
		}
		}

Figure 3. Animating a cursor in INScore and DRAWSOCKET.

4.4 INSCORE - DRAWSOCKET, a significant difference in animation design

4.4.1 INScore Time Model

The description of time in INScore [35] shares with two other French projects (Antescofo [36, 37] and Iscore [38]) similar concerns about technology's ability to handle both continuous and event-driven time. INScore objects therefore have a duration and a date in order to be then synchronized graphically according to their temporal relation, which makes it possible for example to "monitor" an event to trigger another (e.g. trigger a page turn after a cursor the cursor finished crossing the staff. . .). In the example above (see Figure 3) the cursor is synchronised (according to a given tempo - 60) to a line (a segment of a certain length in the graphic space, and of a given duration).

In INScore, synchronising an object (a) to another (b) has a very peculiar meaning, which may be understood as 'making x adopt the spatial properties of y, function of time'. Although this is only one way to understand them, but as useful entry point example, INScore graphical objects may be interpreted as belonging to two different categories : cursors (a) and trajectories (b), or, in other terms players/pointers (a) and score (b). Typically, the cursor is 'enslaved' (or 'synchronised') to a given trajectory (master). Just like a traditional (fixed) score, 'b' can be executed at a slightly different speed each time by its interpreter (a), we therefore attribute a fixed date and duration the score b, and a tempo to the cursor a (an example in *absolute time* is provided in section 5.5, *absolute time* constrats with *musical time* which is tempo-relative).

We find interesting to recall here how various improvements of INScore eventually led to the new specification of a 'tempo' attribute (see [34], end of the introduction), giving evidence of the complex demands of musical time. In *Perspective Temporelles* [23] , observing how the cursor's speed changes at the very beginning might be an apt exemple for illustrating how the time to graphic relationship often needs refined adjustments ([3], section 2.1, time to graphic relation). Indeed, in the above example, the horizontal trajectory of the cursor is divided in two segments [470 , 540[and [540, 2880[, each being given a corresponding duration, hence the perceived effect of an acceleration of the cursor.

[23] The piece is available at http://berio.grame.fr/perspectives/

4.4.2 Animation in DRAWSOCKET

In contrast, DRAWSOCKET offers a simple animation solution, the "tween", from the GSAP library - a standard for javascript animation in HTML5. The animation is defined according to the time and the graphic space to be traversed. The tween approach is easier to handle that INScore for the simple cases, but may on the other hand encounter limitations when the temporal unfolding is not linear with respect to the graphic space. To handle such cases, DRAWSOCKET's tween implementation supports 'multi-segment tween timelines' (accessed from DRAWSOCKET's helpfile, tween animation tab).

5. CASE STUDY: TENOR 21 CHOIR CONCERT

This article is written during a time when the world is profoundly impacted by the aftermath of the Covid-19. Some constituted vocal ensembles have had to endure a whole year without any rehearsals. Contemporary musical practice is suddenly forced to operate massively within a field that Hajdu [11] had investigated since the early days of the internet, and which, at the time, was anything but easy to realise : 'Devising a network performance environment, such as my Quintet.net, is probably among the most demanding tasks a composer or visual artist can face today' [11]. Since realtime audio transmission over the internet was unthinkable at the turn of the millenium, Hajdu had envisaged a system (quintet.net) in which the music played by the instrumentalists were recorded and subsequently encoded to midi, for distribution over the network. As formulated in the title of his article 'Embodiment and disembodiment in networked music performance' [39], Hajdu has also anticipated from an early stage the many possible shortcomings (technical as well as artistic) of performances in which the players are located at a long distance from between each other. Another well-know pre-Covid limitation of network music performance was induced by the fact that network delay prevented musicians from rhythmically responding to each other [40].

As a response to the crisis, and under Hajdu's initiative, a remote choral concert was organised at the Hambourg Horschule for Tenor 2021, in which each singer was provided which a low latency audio kit (Rasbperry Pi 4 + microphone + Soundcard), thanks to the effort of Jacob Sello configuring twenty Raspberry Pi-embedded JackTrip clients [41], together with 2 iPads (one for zoom sessions, and the other for DRAWSOCKET distributed notation).

Five pieces were rehearsed for the concert, all using DRAWSOCKET in various ways, together with low latency audio. This presented the obvious advantage that the singers could all rehearsed together from home, and the composers located in four different countries could also attend the rehearsals.

The DRAWSOCKET feature which proved most helpful here was its ability to dynamically load different pieces of the concert, so that the singers (clients) didn't need to do anything else than join their allocated url at the start of the concert.

5.1 Anders Lind: The Max Maestro

The interface for this piece is originally a max Patch, and was entirely re-written in DRAWSOCKET, so that the composer could remote control the live-generated notation from Sweden while the choir was rehearsing in Hamburg. [24]

5.2 Justin Yang: Prolonged into the latent

As seen earlier in Table 2, *Prolonged into the latent* has its own environment, and DRAWSOCKET was therefore used here only to encapsulate Justin Yang's website within an iframe.

```
/bas4 : {
  /key : "html",
  /val : {
    /new : "iframe",
    /parent : "forms",
    /id : "iframe_ex",
    /style : {
      /position : "absolute",
      /top : "0px",
      /left : "0px",
      /width : "100vw",
      /height : "100vh"
    },
    /src :
"https://pitl.justinyang.net/?parts=0"
  }
}
```

DRAWSOCKET redirects here to the lowest voice of the composition (bass 4, accessed via the DRAWSOCKET url concatenated with '/bas4') to the corresponding part of Justin Yang's website. [25]

5.3 Richard Hoadley: Unthinking Things

Unthinking Things was originally written in INScore. The algorithmic work was composed by Hoadley using SuperCollider sending control messages to INScore [42]. The port to DRAWSOCKET for the concert consisted of a video of the INScore-generated piece, served and synchronised by DRAWSOCKET. [26]

5.4 Jonathan Bell: Common Ground

Common Ground [27] is originally written in bach [17], and most specifically in its most recent *bell* textual extension [43, 44]. The original performance involved singers dancing in an immersive space [27], with scores distributed and synchronised by SmartVox embarqued on a Raspberry Pi. Like for Hoadley, the final performance consisted of videos synchronised via DRAWSOCKET. [27]

5.5 Palestrina: O crux Ave

The Palestrina work was the only one whose language is based on a regular - albeit very slow - pulse, which raised questions mentioned earlier [3] about the role of the conductor when the pulse can be conveyed via animated notation means. The numerous rehearsals allowed iterative attempts with pulse-based animations [28] and contrasting approaches with scrolling cursors realised with I the very beginnin NScore, with which the mapping between time and the pixel-accurate cursor position on the screen can be notated with great precision: in the following example for instance, the cursor travels the distance between x1 (208) and x2 (249) in one second (t2 - t1), then the distance between x2 (251) and x3 (305) in 1 second (t3 - t2).

```
 # x1   x2   y1   y2     t1        t2
([208, 249[ [93, 394[) ([0:0:0, 0:1:00[)
 # x2   x3   y2   y3     t2        t3
([251, 305[ [97, 394[) ([0:1:00, 0:2:00[)
([309, 373[ [95, 396[) ([0:2:00, 0:3:00[)
([375, 437[ [92, 397[) ([0:3:00, 0:4:00[)
```

The perspective envisaged for future attempts will consist of capturing the conductors gesture with gesture follower technology [45] to overcome the limitations caused by the current lag of today's video conferencing platforms (such as zoom), which currently makes conducting impractical.

6. CONCLUSION

With the present survey, the author hopes to have shed light on the emerging field of distributed notation in the browser. With the DRAWSOCKET API and the synchronisation capabilities of INScore, if we think of the scaling capacities of such technologies (the ebb Tunnel performance involved a 144 musicians [29], Le temps des nuages was premiere with 80 singers [30]), or the unforeseable perfoming situations these may lead to when combined with increasingly accessible AR technologies, we hope the cases discussed here will prompt more composers to investigate this exciting field.

Acknowledgments

I am would like to thank Dominique Fober, for his generous time ans support, as well as the Hamburg team for the Tenor 21 choral concert experience.

[24] Original work by Anders Lind: https://youtu.be/4iePLi5uQzU. Version ported to DRAWSOCKET by Jonathan Bell : https://youtu.be/sdSyHIbK5FY

[25] Available at: https://pitl.justinyang.net/?parts=0&controls=yes

[26] The score is available at : https://youtu.be/gLWvjR8vPHw

[27] A recording is available at : https://youtu.be/ZrLgbBw4xfU

[28] A version of the piece animated via the GSAP-tween library in DRAWSOCKET is available here: https://youtu.be/3SS9Cb0AtU0

[29] Score: https://quintetnet.hfmt-hamburg.de/tunnel_webviewer/index.html Performance: https://youtu.be/cdnA_ZijYUI

[30] Recording with the score: https://youtu.be/SyFdR2HiF00 Performance: https://youtu.be/7j2_D-nQAHY?t=6424

7. REFERENCES

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

[2] C. Hope, "Electronic scores for music: The possibilities of animated notation," *Computer Music Journal*, vol. 41, no. 3, pp. 21–35, 2017.

[3] 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*, Hamburg, Germany, 2020, pp. 169–174.

[4] D. Kim-Boyle, "3d notations and the immersive score," *Leonardo Music Journal*, vol. 29, pp. 39–41, 2019.

[5] J. Freeman, "Extreme sight-reading, mediated expression, and audience participation: Real-time music notation in live performance," *Computer Music Journal*, vol. 32, pp. 25–41, 09 2008.

[6] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." in *International Computer Music Conference (ICMC)*, Belfast, Ireland, Aug. 2008, pp. 33–40.

[7] D. Fober, Y. Orlarey, and S. Letz, "INScore - An Environment for the Design of Live Music Scores," in *Linux Audio Conference*, Stanford, United States, 2012, pp. 47–54.

[8] 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'19*, Melbourne, Australia, 2019, pp. 15–25.

[9] G. Hajdu and R. Gottfried, "Networked music performance in the old elbe tunnel," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*. Melbourne, Australia: Monash University, 2019, pp. 55–60.

[10] G. Hajdu and N. Didkovsky, "Maxscore: Recent developments," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, Montreal, Canada, 2018, pp. 138–146.

[11] G. Hajdu, "Quintet.net: An environment for composing and performing music on the internet," *Leonardo*, vol. 38, no. 1, pp. 23–30, 2005.

[12] R. Gottfried and J. Bresson, "Symbolist: An open authoring environment for user-defined symbolic notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, Montreal, Canada, 2018, pp. 111–118.

[13] R. Gottfried, "Svg to osc transcoding as a platform for notational praxis and electronic performance," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation – TENOR'15*, Paris, France, 2015, pp. 154–161.

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

[15] A. Wyatt, L. Vickery, and S. James, "Unlocking the decibel scoreplayer," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*, Melbourne, Australia, 2019, pp. 61–68.

[16] M. Wright, "Open sound control: An enabling technology for musical networking," *Org. Sound*, vol. 10, no. 3, pp. 193–200, Dec. 2005.

[17] A. Agostini and D. Ghisi, "A max library for musical notation and computer-aided composition," in *Computer Music Journal*, 2015.

[18] S. Zagorac and M. Zbyszynski, "Networked comprovisation stratagies with zscore," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'20*, Hamburg, Germany, 2020, pp. 133–140.

[19] J. MacCallum, R. Gottfried, I. Rostovtsev, J. Bresson, and A. Freed, "Dynamic Message-Oriented Middleware with Open Sound Control and Odot," in *International Computer Music Conference*, ICMA, Ed. Denton, United States: University of North Texas, 2015.

[20] D. Fober, G. Gouilloux, Y. Orlarey, and S. Letz, "Distributing music scores to mobile platforms and to the internet using inscore," 2015.

[21] V. Goudard, "John, the semi-conductor: A tool for comprovisation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, Montreal, Canada, 2018, pp. 43–49.

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

[23] A. Pirchner, "Ergodic and emergent qualities of real-time scores. anna and marie and gamified audiovisual compositions," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'20*, Hamburg, Germany, 2020, pp. 189–197.

[24] Y. Orlarey, D. Fober, and S. Letz, "FAUST : an Efficient Functional Approach to DSP Programming," in *New computational paradigms for computer music*, E. D. FRANCE, Ed., 2009, pp. 65–96.

[25] J.-P. Lambert, S. Robaszkiewicz, and N. Schnell, "Synchronisation for Distributed Audio Rendering over Heterogeneous Devices, in HTML5," in *2nd Web Audio Conference*, ser. Proceedings of the 2nd Web Audio Conference (WAC-2016), Atlanta, GA, United States, Apr. 2016.

[26] J. Bell, "Networked Head-Mounted Displays for Animated Notation and Audio-Scores with SmartVox," NIME - New Interfaces for Musical Expression, Jun. 2019.

[27] J. Bell and A. Wyatt, "Common ground, music and movement directed by a raspberry pi," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'20*, Hamburg, Germany, 2020, pp. 198–204.

[28] J. Bell and B. Carey, "Animated notation, score distribution and ar-vr environments for spectral mimetic transfer in music composition," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*, Melbourne, Australia, 2019, pp. 7–14.

[29] G. Santini, "Action scores and gesture-based notation in augmented reality," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'20*, Hamburg, Germany, 2020, pp. 84–90.

[30] D. Kim-Boyle and B. Carey, "Immersive scores on the hololens," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*, Melbourne, Australia, 2019, pp. 1–6.

[31] J. Bell, "Audio-scores, a resource for composition and computer-aided performance," Ph.D. dissertation, Guildhall School of Music and Drama, 2016.

[32] S. Bhagwati, "Elaborate audio scores: Concepts, affordances and tools," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, Montreal, Canada, 2018, pp. 24–32.

[33] C. Sdraulig and C. Lortie, "Recent audio scores: Affordances and limitations," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*, Melbourne, Australia, 2019, pp. 38–45.

[34] D. Fober, "A web based environment embedding signal processing in musical scores," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'21*. Hamburg, Germany: Hamburg University for Music and Theater, 2021.

[35] D. Fober, Y. Orlarey, and S. Letz, "INScore Time Model," in *International Computer Music Conference*, Shanghai, China, 2017, pp. 64–68.

[36] J.-L. Giavitto, J.-M. Echeveste, A. Cont, and P. Cuvillier, "Time, Timelines and Temporal Scopes in the Antescofo DSL v1.0," in *International Computer Music Conference (ICMC)*, ser. Prooceedings of the 2017 ICMC / EMS conférence : Hearing the self, vol. ISBN 978-0-0845274-6-5. Shanghai, China: ICMA, Oct. 2017.

[37] J.-L. Giavitto and J. Echeveste, "Real-Time Matching of Antescofo Temporal Patterns," in *PPDP 2014 - 16th International Symposium on Principles and Practice of Declarative Programming*. Canterbury, United Kingdom: ACM, Sep. 2014, pp. 93–104.

[38] A. Allombert, M. Desainte-Catherine, and G. Assayag, "Iscore: A system for writing interaction," in *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*, ser. DIMEA '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 360–367.

[39] G. Hajdu, "Embodiment and disembodiment in networked music performance," in *Body, Sound and Space in Music and Beyond: Multimodal Explorations*. Taylor & Francis, 2017.

[40] C. Chafe, J.-P. Cáceres, and M. Gurevich, "Effect of temporal separation on synchronization in rhythmic performance," *Perception*, vol. 39, no. 7, pp. 982–992, 2010.

[41] C. Chafe, "JACKTRIP ON RASPBERRY PI," in *LAC - Linux Audio Conference*, 2019.

[42] R. Hoadley, "Homenaje a cervantes," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'17*, H. L. Palma, M. Solomon, E. Tucci, and C. Lage, Eds. A Coruña, Spain: Universidade da Coruña, 2017, pp. 229–238.

[43] J.-L. Giavitto and A. Agostini, "Bell, a textual language for the bach library," in *ICMC 2019 - International Computer Music Conference*, New York, United States, Jun. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02348176

[44] J. Bell, "Improvements in bach 0.8.1, a User's Perspective," in *SMC - Sound and Music Computing*, Turin, France, 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02773883

[45] F. Bevilacqua, F. Guédy, E. Fléty, N. Leroy, and N. Schnell, "Wireless sensor interface and gesture-follower for music pedagogy," in *International Conference on New Interfaces for Musical Expression*, New York, United States, 2007, pp. 1–1.

AUTOMATIC FOR THE PEOPLE: CROWD-DRIVEN GENERATIVE SCORES USING MACHINE VISION AND MANHATTAN

Chris Nash
Department of Computer Science and Creative Technology, University of the West of England,
Frenchay Campus, Coldharbour Lane, Bristol, BS16 1QY, UK
chris.nash@uwe.ac.uk

ABSTRACT

This paper details a workshop and optional public installation based on the development of situational scores that combine music notation, AI, and code to create dynamic interactive art driven by the realtime movements of objects and people in a live scene, such as crowds on a public concourse. The approach presented here uses machine vision to process a video feed from a scene, from which detected objects and people are input to the Manhattan digital music notation [1], which integrates music editing and programming practices to support the creation of sophisticated musical scores that combine static, algorithmic, or reactive musical parts.

This half- or full-day workshop begins with a short description and demonstration of the approach, showcasing previous public art installations, before moving on to practical explorations and applications by participants. Following a primer in the basics of the tools and concepts, attendees will work with Manhattan and a selection of pre-captured scenes to develop and explore techniques for dynamically mapping patterns, events, structure, or activity from different situations and environments to music. For the workshop, scenes are pre-processed so as to support any Windows or Mac machine. Practical activities will support discussions on technical, aesthetic, and ontological issues arising from the identification and mapping of structure and meaning in non-musical domains to analogous concepts in musical expression.

The workshop could additionally supplement or support a performance or installation based on the technology, either showcasing work developed by participants, or presenting a more sophisticated, semi-permanent live exhibit for visitors to the conference or Elbphilharmonie, developing on previous installations.

1. INTRODUCTION

In *The Open Work* [2], Umberto Eco presents a vision of Interactive Art as “works in movement”, based on “openness”, in which audiences become agents in the completion of open-ended artworks, left unfinished by their author and completed during the performance (which may never reach a ‘close’) through a partnership of performers and audience. These works give rise to a multiplicity of meaning, and democratisation of the creative process, but are also realised within the possibilities and constraints afforded by the artist and their tools.

This paper describes a technology-based platform for such works, in the form of a crowd-driven music system, where a notated musical work can make use of detailed visual scene analysis of a physical space, to map dynamic motions of objects and people in a scene (e.g. a crowd, audience, or public space) to the music in realtime, in order to create, manipulate a live performance.

The proposed workshop will practically explore musical opportunities and challenges presented to artists and audiences, and issues of agency, aesthetic, attribution, and adaptability. Participants will be invited to develop crowd-driven musical works, using a platform coupling object-detection (via machine vision) with Manhattan [1] (Section 2.5) – an accessible, yet scalable digital music notation for composition and programming, which flexibly enables users to edit musical phrases and patterns manually, but also insert formulas and code fragments that dynamically manipulate a piece during performance. The notation supports a wide gamut of generative applications, from simple expressions for individual notes to sophisticated algorithmic processes that generate entire works, and event handling that enables interactive works, responding to live input of musical or non-musical data.

This paper begins with a definition and brief discussion of crowd-driven music, followed by a description of the technology and its previous use in public installations, before outlining the particulars of the proposed workshop, installation, and performance.

2. CROWD-DRIVEN MUSIC SYSTEMS

Crowd-driven music is defined here as a situational score in which the live performance or playback of a notated work in some way responds to the realtime dynamic processes or changing structures in an external chaotic system, such as a crowd of people.

The specific technology and techniques featured in this workshop support a closed-loop system (Figure 1) wherein a live video feed of a public space (e.g. a train platform) is analysed in realtime using machine vision (machine learning used to detect and classify objects and people), with the results streamed to a generative music environment (Manhattan, Section 3) that automatically composes (and renders) a score, using a mixture of static (fixed) and dynamic (processed) elements, the acoustic result of which is played back live into the space.

This system of interaction, and specifically the use of machine learning to conduct detailed and comprehensive visual scene analysis, extends previous artworks that use manual input from the audience or public (e.g. [3], [4]), to affect an autonomous, persistent live performance.

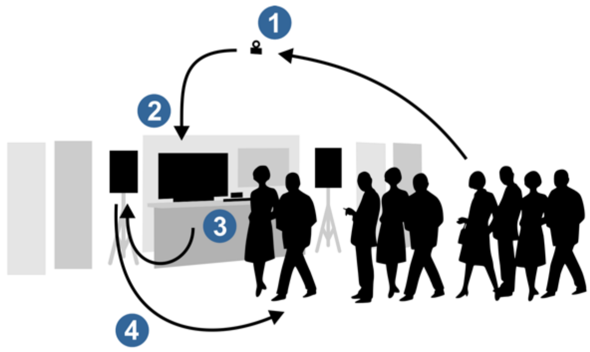


Figure 1 Crowd-Driven Music System

The feedback loop in the system makes it possible for human subjects to take both active and passive roles in a piece; that is, any performance is the product of natural patterns and processes in the crowd, but can also influence crowd behaviour, and be itself subject to influence from individuals and groups within the crowd. This creates both challenges and opportunities for artists to explore new aesthetics and experiences; the agency afforded the crowd versus the integrity of the composer's voice, dictated by the processes and functions employed (and available) to map events and structures between the physical (visual) and musical domains.

2.1 Technical Overview

The system used for the workshop uses machine learning to analyse a scene and detect objects in realtime, using the Darknet *convolution neural network* (CNN) framework [5] and an OpenCL port of the YOLO ("You Only Look Once") v2 realtime object detection system [6]. In common with other machine learning practices, objects are detected using a classifier previously trained on a known set of labelled images, which is then be applied to new images (or frames of a video), detecting objects based their similarity to previously seen examples. YOLO is a performance-optimised detection system that enables the process to run in realtime on modest hardware by reducing the number of convolution processes required per image. In realtime use, using as the graphics processing unit (GPU) of a laptop (MacBook Pro with AMD Radeon 455), the system is able to analyse footage at roughly 7 fps, easily enough to track changes in a scene for the purposes of controlling music.

2.2 Mapping Strategies

The adopted model for mapping scene data to the musical performance is to maintain the state of objects in the scene and allow the playing music to read (pull) the data on-demand. This ensures that changes in the scene are introduced to the performance at musically coherent points or intervals, as decided by the composer – the start of a note, bar, phrase, or section – supporting both fast reactions (events and triggers) and slower gradual processes and context shifts.

While crowd activity can be characterised by noise-like (seemingly random) behaviour, any environment contains hidden patterns and order – structures and processes that can be exposed and exploited in music. Such crowd patterns are not intrinsically musical, placing the onus on composer and code to manipulate them to a given aesthetic. Pre-processing detected objects using techniques such as clustering (grouping similar objects based on proximity) can be useful to extract gestalts that facilitate the mapping process. For example, a naïve 1:1 mapping of individuals to separate musical pitches will overwhelm most systems of harmony or counterpoint, creating a cacophony for anything other than sparse scenes.

Carefully calculated clustering techniques, however, can reduce large crowds to a more manageable finite sets of groups, while also preserving an individual's agency within the music through their influence over the make-up of a group or movement between groups. Such grouping mechanisms allow pieces to respond to higher-level structure in a crowd, which accordingly can be mapped to more abstract musical processes (e.g. tempo, form, structure, harmonic progression). Other notable techniques include linear mapping (e.g. number of people ~ tempo), parameter smoothing (preserving gradual changes without responding to noise), constraints (e.g. quantising pitch to scales or harmonic pitch sets), and condition (i.e. if-the-else; to detect events or categorise data into ranges).

2.3 Performance Platforms

The presented system was originally developed to support a public installation for *BBC Music Day 2018* [7,8], in collaboration with the BBC and Great Western Railways, whereby two crowd-driven pieces were commissioned and installed on the main platform of Bristol Temple Meads station for the day. BBC Music Day is an annual event (covered by BBC TV and Radio) to get the public more involved in music through a series of performances and activities across the country. In Bristol, the crowd-driven music installation was developed to transform travellers passing through the main railway station into composers.

The following objectives were developed to guide the development of the technology and experience:

- ***Agency*** – Members of the public should be able to appreciate their influence on the music.
- ***Accessible*** – The music should have broad appeal to an audience composed of the general public.
- ***Aesthetic*** – The technology should support a coherent musical voice, idiomatic of the artist.
- ***Adaptive*** – The piece should vary over time, responding to context and avoiding repetitiveness.

Each piece tracks platform activity with respect to four objects: people, trains, luggage, and bicycles, using their positions, numbers, or grouping to create and control musical patterns or processes, such as melody, harmony, dynamics, tempo, or instrumentation. Objects were chosen to provide elements that support constantly changing contexts over time, at different rates (constant, regular, occasional, rare). Figure 2 shows an example scene with object detections, as displayed to onlookers on the platform; also showing detected clusters (groups of people) and their relative sizes as blocks along the lower edge.

Installations have since featured in events such as the *Sofia Science Festival*, with new public art commissions for UWE's new Engineering building and local museums.

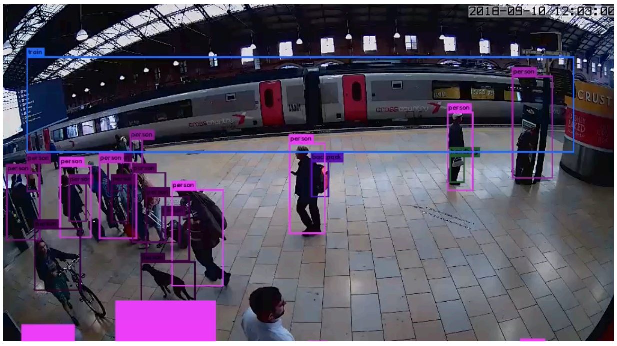


Figure 2 Realtime Object Detection on the Platform

2.3.1 Not So Different Trains

The first piece (Supporting Video 1) used scene data in the live composition of a minimalist piece for piano and virtual choir (singing the suburbs of Bristol using concatenative synthesis), where: the distribution of the crowd determined harmonic pitches and chord voicing; its size and density (quiet vs. busy) determined tempo, instrumentation, and dynamics; and specific events (such as bicycles) varied chord quality. An algorithmic process was used to calculate pitch sets and control semi-randomised progression between tonic, sub-dominant and dominant functional chord families, to provide a tonal palette for crowd-mapped processes, based on an extended *Tintinnabuli* technique, inspired by Arvo Pärt.

2.3.2 Massive Railtrack

The second piece (Supporting Video 2) applies similar processes in development of a "trip-hop" musical style, in homage of Bristol-based band, *Massive Attack*. The harmonic language follows a simplified I-IV-V process derived from the previous piece, synthesised using a multi-sampled local community choir (*Rising Voices*, Bristol Drugs Project's recovery choir) with a synth bass part tracks the largest group of people within the crowd and appropriately selects from the tonal palette. Further drum programming uses untuned samples recorded by the choir (as well as loop), layered according to the relative busyness of the platform (number of people, trains, etc.).

2.4 Supporting Videos

The following videos are available online [8] from:

http://nash.audio/manhattan/tenor2020

Performance Video 1 *(Classical style; ~12mins, HD/MPEG4)*
Screen capture of live music generation with inset platform footage, classical style using piano and synthetic choir (programmed to sing the neighbourhoods of Bristol).

Performance Video 2 *(Trip Hop style; ~12mins, HD/MPEG4)*
Screen capture of live music generation with inset platform footage, trip-hop style using synthesisers and sampled *Rising Voices* choir (for both voice and drum sounds).

Early Technical Demo *(annotated initial experiments; 2:34)*
Annotated screen capture of early experiments with basic music mappings of people locations to notes (pitch and rhythm) and percussion density to crowd density. Annotations are provided to explain the system and process.

2.5 The Manhattan Software

Manhattan [1] (pictured in Figure 3) is a digital music platform developed for learning and creativity in both music and programming – designed to extend traditional computer music practices (such as sequencing) through code fragments situated in the music notation, to support algorithmic, reactive, and dynamic pieces.

Manhattan offers an accessible, yet scalable introduction to music programming for computer musicians, and is used in teaching to develop computational thinking skills in non-coders, part of a wider initiative to support digital literacy and widen participation in coding. The environment exploits the grid-based pattern sequencer style of *soundtrackers*, made of cells specifying notes or other musical events, and applies a spreadsheet metaphor to introduce formulas to musical playback, inheriting many of benefits that have made spreadsheets one of the more successful models of end-user programming. [9]

Unlike other programming languages, the visibility of the data (music), rather than code (formulas), is prioritized, enabling a traditional sequencing/editing workflow, but where the effect of code on the music is apparent. [10] Users can balance manually edited music sequences (static patterns) with varying levels of engagement with programming abstractions (dynamic processes), from simple expressions for isolated dynamic behaviour at specific moments (e.g. conditional repeats, random elements) to formulas for generating entire pieces (e.g. algorithmic music, minimalism, aleatoric music). The environment similarly supports event handling of realtime user and data input to support reactive and interactive applications for live performance, live composition, improvisation, interactive installations, or other sonic art. As such, Manhattan supports continuum of musical practices and aesthetics, from conventional popular and classical styles, to more experimental art and contemporary forms.

Manhattan is being developed as part of a research project involving artists, universities, and schools that is looking at tools to support and extend creative and pedagogical practices in both music and programming. The software is free to download[1] for MacOS and Window, and includes everything required to start writing and programming music (including built-in sounds and synthesisers, plus extensive interactive tutorials).

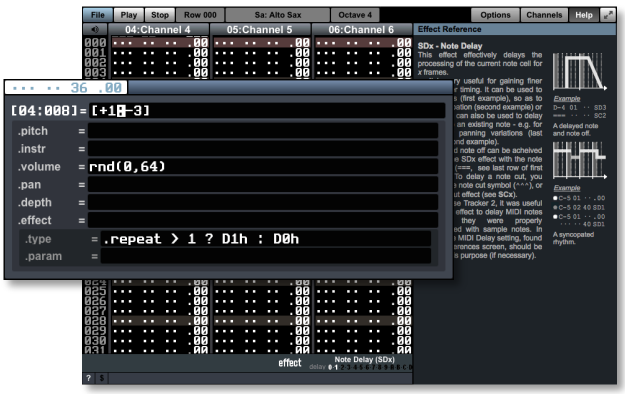


Figure 3 Manhattan music software

[1] http://nash.audio/manhattan

3. WORKSHOP

The workshop is designed with the following objectives:

- Discuss and explore practical applications and expressive opportunities in mapping complex or chaotic systems (e.g. public crowds) to both live and prepared musical works.
- Experimentally develop works of crowd-driven music, using AI (machine learning) and generative techniques, supported by the Manhattan software.
- Identify effective musical analogies and ontologies for structures and processes in non-musical systems, and the aesthetics they engender.
- Establish directions and collaborations for future research or artistic projects.

3.1 Proposed Schedule

A representative draft itinerary for a half-day (3-4hr) workshop is presented below, adaptable to the conference programme. A full-day (5-6hr) alternative is also outlined (changes marked *), specifically allocating more time for practical development and discussion. The extended full-day programme is recommended if conference organisers select the option to exhibit participants' works.

00:00 – Welcome and Introductions *(15m, all)*
Host and delegates introduce themselves, briefly describing their background and respective areas of interest or expertise.

00:15 – Opening Presentation *(30m, organisers)*
Audio/visual presentation introducing relevant concepts and technologies (i.e. *Manhattan* and machine learning concepts), with demonstrations of previous artistic works, basic syntax and expressive mapping techniques.

00:45 – Manhattan Primer *(30m or 60m*, all with support)*
Simple practical exercises using prepared materials, designed to introduce delegates to the fundamentals of the Manhattan tool and syntax, demonstrating core coding concepts (e.g. variables, data, iteration, functions, conditionals) using musical examples. This group exercise will encourage open discussion of techniques and participant's interests and backgrounds, used to help frame and guide subsequent sessions.

[15m BREAK or LUNCH*]

01:30 – Music for One *(30m, all with support)*
Initially, mapping concepts are explored through event handling of simple live inputs (using provided MIDI controllers), applied to affect change in repeating musical patterns, in preparation for more complex scenes and scenarios.

02:00 – Music for Many *(60m or longer*, all with support)*
Participants choose from a selection of pre-captured footage featuring various public settings (e.g. foyers, stations, streets), applying what they have learnt, to develop musical mappings and generative processes that respond to live stream of data about detected objects, such as their position, number, groupings, etc. Within each example setting, a diverse range of objects are tracked (e.g. people, vehicles, animals, luggage).

03:00 – Closing Discussion *(15mins, all)*
Review of issues and findings (or research goals) that have emerged, and call for interest in further research / collaboration. Participants are invited to continue developing their pieces for future exhibition as an installation or performance, possibly as part of the TENOR conference programme.

3.2 Intended Audience

This workshop is suitable for all TENOR delegates, especially those with an interest in interactive or generative music. The technology is designed to be accessible to any musician or computer user, and requires no specific expertise in programming. However, the workshop would particularly suit those with backgrounds, research interests or experience in: notation, composition (modern or common-practice), sequencing, programming (usage and semantics), live installations or performance, and artistic applications of AI/machine learning.

3.3 Required Resources

The workshop has no special requirements. Depending on attendance, it will require a single room with a capacity of 20-30, with table space for each delegate (boardroom, u-shaped, cabaret or classroom layout). There should be a projector/screen with VGA/HDMI connection and a high-quality stereo system for computer audio.

The workshop can be hosted using delegates' own machines or in a onsite computer room, running either Mac (preferred) or Windows, so long as third-party software is supported. Footage is pre-computed, obviating the need to apply processor-intensive machine learning in realtime, such that participants should be able to use any Windows/Mac machine.

Manhattan is freely available and has no specific (e.g. hardware or admin) requirements. Participants can download the software in advance, and retain it after the event. Workshop registration should indicate their preferred OS (Mac or Windows, including version and language). Tablets (e.g. iPads) are not supported. Access to WiFi (e.g. eduroam) is desirable.

Where possible, tea and coffee facilities are desirable. No fee is required for participation in the workshop, unless deemed appropriate by the conference organizers.

3.4 Exhibition of Works

For exhibition of workshop outputs, in the form of a video installation, demo, or presentation; a large TV or projected screen with USB input and loudspeakers would be required, to host a playlist of rendered videos showcasing the participants' works for their chosen scene. The video will be prepared at the conclusion of the workshop.

4. INSTALLATION / PERFORMANCE

Separate from the workshop, a public installation based on the live crowd-driven system is also proposed, showcasing an original interactive work designed for a specific open space, such as a public area of the Elbphilharmonie.

The installation would be adapted from the form used for *BBC Music Day* [7,8] (discussed in Section 2.3/4), with relatively modest technical requirements (TV or projected screen, loudspeakers – see Figure 4), but will require some consultation to identify an effective space and live scene to support an expressive performance. For example, the original installation, a train platform, offered a constant symphony of motion, including proximal and lateral movement of people, groups, and objects, at varying paces, sometimes stationary, while fluctuating in density over time (from very quiet to very busy). The system provides considerable flexibility with respect to mapping options, where different composition strategies can accommodate a wide range of settings, but expressive possibilities are ultimately tied to the entropy of the scene itself, and will suffer from too quiet or uniform a scene.

The work(s) presented can adapt previous material (e.g. adapting Bristol themes for Hamburg) or be newly commissioned with respect to a defined musical brief (with respect to the hosting organisation or intended audience) or a specific interactive role (with respect to the space or activity therein). An experimental aesthetic, for example, might suit an audience of TENOR delegates more than the general public – and, in a public or civic space, works must also be considerate of the space's function; music cannot negatively interfere with normal affairs or inhabitants, being either intrusive, distracting, or annoying. On the train platform, for example, the priority was the smooth and safe running of the station.

The installation should be able to run for several days (or longer) unattended (using a dedicated machine designed to support a permanent installation, currently in development) and can connect to its live video feed remotely (through an ad-hoc wireless network), enabling flexible and discreet location of the camera module. Options for online or local (smartphone) streaming are also possible.

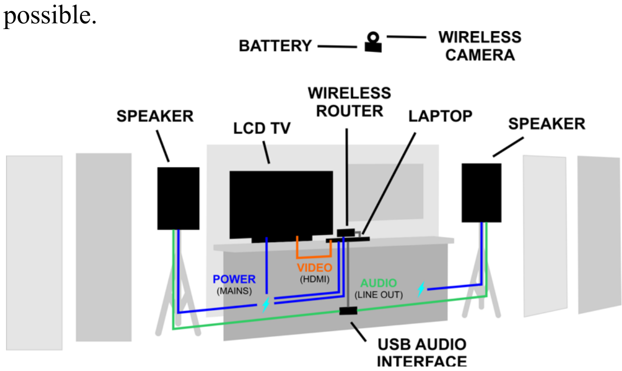


Figure 4 System Overview for Public Installation

4.1 A Note on Privacy

For live or recorded works, this installation is compliant with EU laws concerning privacy and the processing of personal data, such as the GPDR. In live use, captured video footage and data detections are not stored, but processed exclusively in realtime before being discarded. The footage and all data derived from it (i.e. object detections) contain no personally-identifiable information (individuals are identified as "person" only), which is transparently displayed at time of performance. Unless otherwise arranged, only public spaces and scenes of crowds where individuals have no reasonable expectation of privacy are captured. Previous public installations and performances have been ethically reviewed and approved by the BBC, National Rail, and UWE Bristol. Provided previously-captured footage (used in closed workshops or academic settings) has similarly been collected and curated to ensure ethical use of data and protection of privacy.

5. REFERENCES

[1] C. Nash, "Manhattan: End-User Programming for Music," Proceedings of New Interfaces for Musical Expression (NIME) 2014, 2014, pp. 28-33.

[2] U. Eco. *The Open Work.* Cambridge, MA: Harvard University Press, 1989.

[3] Y. Wu, L. Zhang, and N. Bryan-Kinns. "Open Symphony: Creative Participation for Audiences of Live Music Performances" in *IEEE Multimedia*, January, 2017.

[4] S. Bhagwati. *Making Music*. Ulm: April, 2000.

[5] J. Redmon. *Darknet: Open Source Neural Networks in C*. Available: http://pjreddie.com/darknet, 2013.

[6] J. Redmon and A. Farhadi. *YOLO9000: Bigger, Better, Faster.* arXiv:1612.08242 [cs.CV] Cornell University, 2016.

[7] C. Nash. *Track by track: Generative music installation for BBC Music Day*. Technical Report: http://uwe-repository.worktribe.com/output/860147, Bristol: UWE Bristol, 2018.

[8] C. Nash. *Manhattan software, videos and materials.* Available: http://nash.audio/manhattan/shared, 2018.

[9] J.F. Pane and B.A. Myers. *Usability Issues in the Design of Novice Programming Systems*. Carnegie Mellon University, Technical Report CMU-CS-96-132, 1995.

[10] C. Nash, "The Cognitive Dimensions of Music Notations," in *Proceedings of the Second Internation Conference on Technologies for Notation and Representation of Music (TENOR)*, Paris, 2015.

About the Organizers / Speakers

Dr Chris Nash (principal organizer) is a professional programmer and composer – currently Senior Lecturer in Music Technology at UWE Bristol, teaching software development for audio, sound, and music (DSP, C/C++, Max/MSP). His research focuses on digital notations, HCI in music, virtuosity, end-user computing, systematic musicology, and pedagogies for music and programming.

Corey Ford (technical support, recorder) is a postgraduate researcher at UWE Bristol, completing an MRes Data Science studying technology for learning music notation and programming, supervised by Dr. Nash.

Hamburg and London-based music software developers, *Steinberg Media Technologies GmbH*, creators of the *Cubase*, *Nuendo,* and *Dorico* music editors, have also been approached to participate and support the workshop, based on a previous working relationship with the organisers. The extent of the company's involvement will be confirmed closer to the event.

AUTOCONDUCTOR – SYNCHRONIZING GRAPHIC SCORES, MULTI-CHANNEL SOUND AND FULL-HD PROJECTION

Lothar Felten
Freelance programmer
lothar.felten@gmail.com

Christian Klinkenberg
Composer
klinkenbergcomposition@gmail.com

ABSTRACT

In this paper, a networked score display system which synchronizes graphical notation, precise pitch notation and video with multi-channel audio is presented.
The software was purpose built for the project *The Glacier - Opera 2.0* composed by Christian Klinkenberg.

1. INTRODUCTION

Since 2016 the composer Christian Klinkenberg has been using a graphic video score for his compositions. This was the case in *Das Kreuz der Verlobten* and *The Leaves That Hung But Never Grew*. Unfortunately, the synchronization of the video score was insufficient in both cases. This was the trigger to develop a software to meet the special needs.

1.1 The previous projects

Concerning the project, The Glacier - Opera 2.0 [1] , the composer Christian Klinkenberg and the programmer Lothar Felten have jointly developed the software.
Since the first opera *Das Kreuz der Verlobten*, Christian Klinkenberg has worked with video notation. All musicians played an individual video file on their computer. The scrolling of the frames of the graphic score from right to left was performed with the transition Slide-Push in the software Adobe Premiere[1]. The conductor gave the signal and everyone pressed Start or Play on their video player. Real synchronicity was therefore by no means provided.
For the composition *The Leaves That Hung but Never Grew* [2] , a public domain software called *syncplay* [3] was used. Lothar installed the software on a server, and the musicians had to install a client version on their device. There were clients for all standard operating systems. The devices brought by the musicians where running iOS, Android and Windows, while the syncplay-server was running Linux. The advantage of syncplay was that the rehearsal work was much more comfortable than without this synchronization. As soon as a client jumped at a particular scene, all computers jumped with it. The disadvantage of this software was that the videos are only synced at start, no intermediate synchronization is applied. After the length of the entire piece, the videoscore was offset by up to 2 seconds between the slowest and the fastest device.
So, the project autoconductor was started.

1.2 Available solutions

There are multiple solutions available that provide means for scrolling through an image-based score in sync over the networks, such as *Decibel ScorePlayer* [4] [2][3] or *Quintet.net* [5] [4]. Two hard requirements for the composer's works are synchronization with a continuous video stream for the audience and the possibility to show a common graphical notation and a distinct precise notation per instrument on one screen for each musician.
Most available solutions also require a significant amount of configuration on the part of each musician. A notable exception is the *SmartVox* [6] [5] project that uses a web-based approach and hence supports a wide variety of platforms.

1.3 Goals for the new project

Parallel to the video scores, a video in mp4 with 1080p and 7.1 sound should be projected for the audience. The multi-channel sound was important because, among other things, a click-track was used, which of course should not be heard on the soundtrack of the audience.
The scores for the musicians consisted of 2 parts:

- The upper part consists of the graphic score, which remains the same for all musicians.
- The lower quarter consists of a pitch notation on staves[6].

Because of the microtonal compositional aspects, in addition to the standard notation with cent deviations[7], tablature notation and other alternative notation systems such as for the Bohlen-Pierce clarinets, the "*Mller-Hajdu notation* [7] [8][9] was used. This score was notated in four colors in the software Finale and later converted into a png-picture. The four colors are needed to link the notes to the graphic. When selecting the colors, care always had to be taken to ensure that they remained easily distinguishable from the

[1] http://www.operatheglacier.com
[2] http://www.kl-ex.com/en/projects/the-leaves-that-hung-but-never-grew/
[3] http:https://syncplay.pl

[4] http://decibelnewmusic.com/decibel-scoreplayer/
[5] http://quintet.net/
[6] https://www.bachproject.net/2016/09/07/the-smartvox-project/
[7] http://www.noralouisemuller.de/Muller_BP-Notation.pdf

background, which was also colored[10].
A precise time positioning should simplify the rehearsal work. So, it is possible to jump from one position to another.

2. TECHNICAL REQUIREMENTS

The technical requirements for *The Glacier Opera 2.0* can be summarized as a simple, lightweight tool to provide an audiovisual stream for the audience in synchronicity with the complete scores for all musicians.
Looking at the details, the complexity of the task becomes clear: The video does not only provide a stream with stereo sound for the audience but also includes multiple click-tracks for the drummer and the conductor. Also, supporting a wide variety of devices to show all the different scores in sync with the video over a wired or wireless network with an accuracy high enough to play music together is another aspect which had a significant impact on the design decisions.

2.1 Prior experience

The first collaboration was a year earlier for an interactive musical narrative in the microtonal system *Bohlen-Pierce* [8] with graphical scores. In the successful project *The Leaves That Hung but Never Grew*, we gathered initial experience with a setup where the audience was invited to play along with the musicians was gathered: At the start of the performance, everyone was invited to use their smartphone to connect to a wireless network and to use the browser to produce their own sounds. A large screen showed the images illustrating the fairy tale and included a superimposed score in the form of graphical notation. Symbols in 7 different colors, which should represent a certain musical content[11] of the graphical score were also used on the user interface shown on the smartphones of the participating listeners. This first project showed that its feasible to provide a responsive, low latency interface using browser-based technology over a wireless network. The same techniques were used as the foundation for the *autoconductor* project.

2.2 Client side

From the start it was clear that no dedicated hardware would be used the musicians would bring in their own devices and with it a wide range of hardware and operating systems. The lowest common denominator for the entire range of mobile and portable devices is probably a web browser.
An *app* would have been an option, but it would have required a cross-platform framework to support at least Windows, Android, iOS and Linux and needs to be installed beforehand.
A web-based application on the other hand, should run on every platform supporting a modern browser without any installation. Ideally, it also behaves in the same way on all systems. Also, there is little to no training required as the users know how to connect to a network and navigate to a website.

[8] `http://www.huygens-fokker.org/bpsite/`

To keep the setup time for the musicians as low as possible, a single web page provides the entire content. Except for an initial drop-down box to select the score for the instrument at hand, there is no further control element on the page.
The content is provided and controlled by the server, the client-side technique used to display the scores is JavaScript and the communication is done via WebSockets.

2.3 Server side

The main design goals for the server-side were portability, ease-of-use, low latency and low cost. The opera was performed in multiple venues in Belgium and New York without a dedicated staff member for the server or network setup, hence the requirement for a system that once configured can be setup in a short amount of time. Also, the server and network gear consisted of a re-purposed laptop, a second-hand internet router and some cables – parts that can be quickly taken onto a flight. The sole purpose of the router was to provide a wifi access point and a private network, there is no internet access required. A mixed operation of wired and wireless operation is possible. The preferred method is a wired network, but recent improvements of wireless network technology make it a viable option.

Figure 1. Performance in New York without a human conductor.

The laptop runs the Linux operating system, the *nginx* [9] webserver and the main application written in C++/*Qt5* [10].
If equipped with a second screen or a projector, the video output can be shown in full-screen mode on the second screen while the first screen shows the window with control elements.

2.4 Multichannel audio

In contrast to *Das Kreuz der Verlobten*, bars and rhythm should also be possible. The visual information was not sufficient. So, in addition to the stereo sound (Gabriel's voice, radio and effects) for the audience, we needed at least one additional channel for the click. Preliminary experiments with the ensemble confirmed the assumption that the tempo is often unstable with purely visual information.
The possibility of working with multi-channel sound for

[9] `https://nginx.org/`
[10] `https://www.qt.io/`

future productions should also remain.
The first rehearsals with the software still contained some bugs, which were fixed step by step. For example, in the beginning, there was still very strong stutter. Reducing the update rate was the solution to this issue. Besides, it turned out that the files of the graphics in *png* [11] format were very large and caused significant load on the wifi network. In the *jpeg* [12] format, however, the pictures painted by Marc Kirschvink in 1080p with small file sizes could achieve an acceptable quality despite lossy compression.

3. REALISATION

The main concept is based on the video shown on a large screen and the scores of all musicians will be synchronized with the video. If the video is paused or positioned to a different time code, the displayed scores will scroll accordingly. The graphical score consists of a long series of images, which, when placed side by side would make up a single long score for the entire musical piece. These images scroll by at a constant speed and the currently shown position is synced to the video time code.
The video playback is controlled by an application with graphical user interface that runs on the server, see Fig. 2.

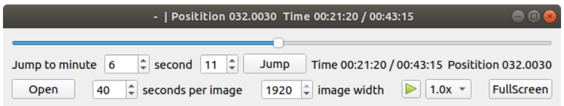


Figure 2. Video controls.

In addition to the normal controls a video player offers, there are fields labeled *image width* and *seconds per image*. Those fields specify the width of a single image of the score and the amount of time it will take to scroll by.
The total amount of images needed depends on the length of the video track, the width of the images and the number of seconds per image. Next to the time code labels, the *Position* label shows the current image and pixel associated with the current time code.
For convenience the *Jump* button was added, which repeatedly allows a quick jump back to a given time code, a useful tool for rehearsals.
For the musicians this is hidden, they only see a full screen browser and after the initial selection of the instrument a score, see Fig. 3.
Here the score consists of two parts, the top is the graphical notation and the bottom quarter is used for supplemental information, either scores or text.

4. IMPLEMENTATION DETAILS

The entire solution is implemented as a classic client server architecture. All communication is done via http and WebSockets. In contrast to classic AJAX which uses *XMLHttpRequests* [13], Websockets allow bidirectional communication between the client and the server. In our case, we make extensive use of the server to client direction.
Other research by *Augusto Ciuffoletti* [14] has shown that latencies below one second are possible over the internet. Our implementation proves that in a local, dedicated wireless network the attained latency is low enough to synchronize scores for musicians.

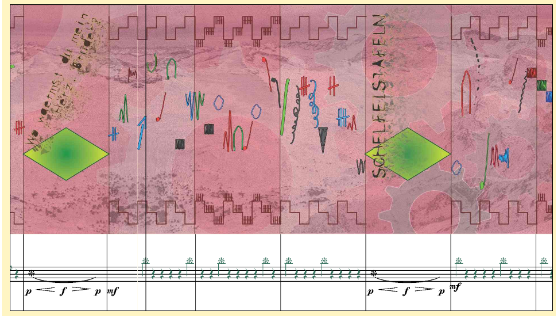


Figure 3. Musicians' score.

4.1 Server application

The autoconductor application itself is an application written in *C++* [15] using the Qt5 framework. Qt5 was chosen because of the good multimedia library, GUI toolkit and *WebSocket* [16] support. Written in plain C++ the system load caused by the application is low and it yields excellent response times required for low latency use cases. Main component is the video player with multichannel audio support, the video output is rendered in a separate window for dual screen setups: one screen for video, one for the control panel. In a separate thread there is a WebSocket server running, it keeps track of all connected clients and sends the current position of the video in regular intervals. The data transmitted over the WebSocket are formatted in *JSON* [17] and carries only the current image number and the current pixel position in the image. Its up to the client to display the correct images in the correct location, see Fig. 4.

4.2 Client-side JavaScript

The client side is a single web page, the initial view shows a drop-down box to select the instrument or track. Once selected, the drop-down box is hidden and a new HTML canvas element will use the entire view port to render the score.
All following action and communication is handled by client side JavaScript.
Immediately after the initial instrument or track selection, the JavaScript code will open a WebSocket to the server and start preloading all the JPEG images for the entire graphical

[11] `https://en.wikipedia.org/wiki/Portable_Network_Graphics`
[12] `https://en.wikipedia.org/wiki/JPEG_File_Interchange_Format`
[13] `https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp`
[14] `https://arxiv.org/abs/1901.00724`
[15] `https://isocpp.org/`
[16] `https://developer.mozilla.org/de/docs/WebSockets`
[17] `https://www.json.org`

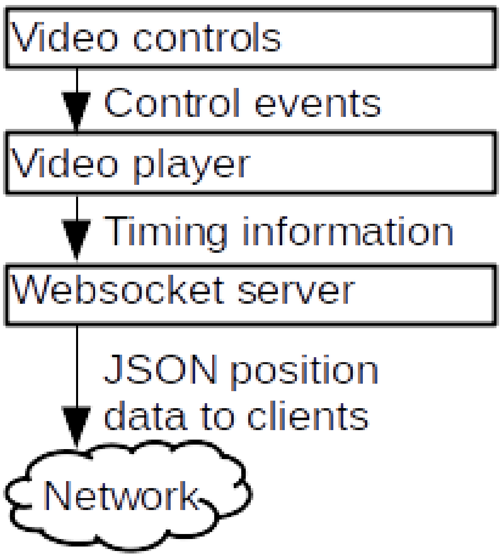


Figure 4. Server side.

notation and all PNG images for the precise notation, see Fig. 5.

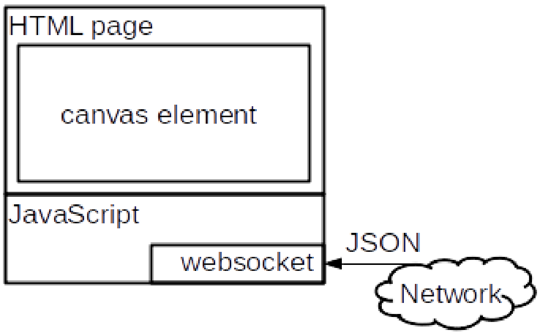


Figure 5. Client side.

The WebSocket connection to the server is opened and incoming JSON messages are parsed. The incoming data is used to display the correct images on the HTML canvas.
The initial approach was to load the images on demand but this caused spikes in network load and caused jitter on the display timing: whenever a new image was due, all connected clients issued two new requests to the webserver to load the two images needed. On average, the network load was low, but the congestion by periodic spikes caused temporary missing images on the client side. Also, some of the sync packets were lost, causing jitter in displaying the images, see Fig. 6.
Preloading the images consumes more memory on the client side and also causes peak in the network load at the initial load, but there is no additional network traffic during the performance. Even a single client reloading the entire graphical notation during the performance causes some negligible load and has no noticeable impact on the other clients. Images are preloaded in order, so if the current position is not at the very beginning, a few seconds could be missed.

Figure 6. First test with a server and 10 client devices.

Another major advantage of preloading all the images is that jumps during rehearsals are instantly: Scrolling in either direction on the central server causes the clients to show the correct position in real time.

5. POSSIBLE ENHANCEMENTS

The software was purpose build for the project *The Glacier - Opera 2.0*, but with some improvements this might be a generic tool.
Here is a short list of possible enhancements that might make the software usable for a wider audience.

- Variable speed throughout the score. Currently the score scrolls by at a fixed speed, it would be nice to have a feature that allows to set the scroll speed for each passage.
- Conductors score, one page with all parts. There is no way to display all scores on one screen at the moment. This is mainly a layout issue to fit everything on one screen in a legible manner.
- User friendly way to import and sort score images. Finding the correct images relies on filenames – each image has a dedicated filename. It would be nice to have a tool that splits the scores from a PDF and renames them automatically.
- Plugin for *MaxMSP* [18] . *MaxMSP* is quite popular, providing an interface to it would be a great addition.
- A midi-output synchronized with the score. In this way, electronic instruments, lights via *DMXIS* [19] , effects or events could be controlled remotely in *Ableton* [20] or similar digital audio workstations.
- Live measurement and indication for the current latency. Some basic latency measurements have shown that a wired network is indeed preferrable over a wireless network. A display of the quantitative latency and the variance between clients might give interesting insights for future improvements and research.

[18] `https://cycling74.com/`
[19] `https://www.enttec.co.uk/en/product/controls/dmx-lighting-control-software/dmxis/`
[20] `https://www.ableton.com`

6. CONCLUSION

After initial difficulties on the first rehearsals, the software ran stable from the final rehearsal starting a week before the first performance. 15 devices running different operating systems logged in via wifi. Sometimes there were stutters. But this is due to the wifi network, this could be remedied with a better router. A lot of different devices from notebooks to tablets to smartphones were used, so that the Webapp system proved itself in any case. Autoconductor is going form the basis for many new compositions and performances. The software is constantly being developed further and the current version for Linux can be found at this link:`https://www.christianklinkenberg.com/research/autoconductor/`

7. REFERENCES

[1] M. Jago, *Adobe Premiere Pro CC Classroom in a Book*. San Fransisco: Adobe Press, 2017.

[2] L. V. C. Hope, *Visualising the Score: Screening Scores in Real-time Performance*. Interactive Media E-Journal, 2011.

[3] R. L. C. Hope, *The Decibel ScorePlayer - A Digital Tool for Reading Graphic Notation*. ECU Publications, 2015.

[4] G. Hajdu, "Quintet.net: An environment for composing and performing music on the internet," *LEONARDO*, vol. 38, no. 1, pp. 23–30, 2005.

[5] B. M. J. Bell, "Smartvox-a web-based distributed media player as notation tool for choral practices," in *Proc. Int. Conf. on New Tools for Music Notation and Representation - TENOR 2017*, La Corua, Spain, 2017.

[6] K. Stone, *Music Notation in the Twentieth Century: A Practical Guidebook*. New York City: WW Norton & Co., 1980.

[7] E. Karkoschka, *Das Schriftbild der Neuen Musik*. Celle: Moeck, 1991.

[8] N.-L. Müller, *The Bohlen-Pierce Clarinet: Theoretical Aspects and Contemporary Applications*. Books on Demand, 2020.

[9] S. S. M. Stahnke, *1001 Mikrotne*. Von Bockel Verlag, 2015.

[10] S. F. D. Kastan, *On Color*. New Haven: Yale University Press, 1991.

[11] J. Wolfe, *Sound Musicianship – Understanding the Crafts of Music*. Cambridge Scholars Publishing, 2012.