3D SCORES ON THE HOLOLENS

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

Benedict Carey
The University of Music and Drama
Hamburg, Germany
benedict.carey@hfmt-hamburg.de

ABSTRACT

The authors describe work undertaken in porting the 3D, Max generated score for David Kim-Boyle's *5x3x3* (2018) to the Microsoft HoloLens. The constraints of various network protocols for communicating between Max and the Unity 3D platform, with which HoloLens applications are built, are discussed and the deployment process from Unity to the HoloLens is described. Various optimization considerations are outlined, and the paper concludes with a discussion of feedback from performers and a brief appraisal of some of the unique aesthetic affordances of the HoloLens platform.

1. INTRODUCTION

5x3x3 (2018), for any three wind instruments, is a work featuring a 3D graphic score generated in real-time. Originally developed in the Max programming environment and intended to be read from flat screens, the score presents a series of three cubic grids of colored nodes connected by thin, colored lines of variable length and curvature, see Figure 1, which gradually merge into a single, larger construct as the score develops. Line colors (red, green, blue, yellow, white) denote different pitches while node color, line length and curvature denote various articulative, temporal and timbral properties, respectively, of the pitches performed.As the three performers explore various pathways through the score, node positions, and the corresponding length of lines connecting them are dynamically transformed. The score of *5x3x3* is also responsive to timbral nuance with the spectral content and pitches of notes performed by the instrumentalists used to transform line curvatures and colors respectively. These various transformations provide new musical possibilities for each of the three performers to explore.

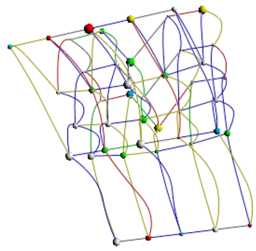


Figure 1. Detail from the original Max score for *5x3x3* (2018). Colored nodes represent different types of articulation of various pitches denoted by colored lines. Performers explore various pathways through the score as it is transformed during performance.

The display of 3D graphic notations on two-dimensional surfaces presents fundamental constraints [1] for which mixed reality devices such as Microsoft's HoloLens, see Figure 2, offer viable solutions. Released in 2016 and featuring an Intel Atom x5-Z8100 1.04GHz CPU, 2GbB RAM and a 32-bit OS, the HoloLens presents compelling interactive, three-dimensional visualizations and immersive sound to the wearer. While the device is receiving growing investigation for its applications in fields ranging from medicine to architecture, [2, 3] its use as a tool for the creative arts has received less attention although this is somewhat unsurprising given the recency of the device. For display of the score for *5x3x3* the HoloLens not only offers a means of displaying three-dimensional notational schemas in a realistic manner, but it also presents new aesthetic affordances through foregrounding the physical relationship between performer and virtual image.[1]

Figure 2. Microsoft's HoloLens.

[1] In February, 2019, Microsoft announced the release of the HoloLens 2 featuring a faster, Qualcomm Snapdragon processor, a doubled angle of view, and greatly improved ergonomics. For the research conducted in this paper, however, the original HoloLens device was used.

2. DATA TRANSFER FROM MAX TO UNITY

With the original intention of preserving as many of the generative processes as possible within the Max environment, the porting of *5x3x3* to the HoloLens began with an evaluation of how various network protocols would support transferring score data from Max to Unity 3D, the programming environment on which HoloLens applications are developed and deployed. The score information generated by the Max patch for *5x3x3* includes the *xyz* coordinates of nodes, RGBA color values of nodes, *xyz* coordinates for the terminal points of lines connecting nodes, *xyz* coordinates of curvature control point/s for each line,[2] and RGBA color values for lines. While some of this data is generated and transformed with respect to global time, and thus runs independently of performance interpretation, data derived from FFT analyses of instrumental timbre is used to drive a number of score properties including the curvature of lines.

In the original patch, developed in Max 7, score data is rendered to spheres and lines in OpenGL with *jit.gl.multiple*/*jit.gl.gridshape* and *jit.gl.mesh* objects. In initial development within the Unity 3D environment, the authors sent coordinate and color information for these graphic objects to Unity 3D, where they were rendered as 3D objects (spheres) and empty objects with line renderer components, over a TCP/IP socket via localhost through *jit.net.send* and *jit.net.recv* objects and use of Bukvic and Kim's Max-Unity interoperability toolkit [4]. Based on previous research, [5] the transfer of data via TCP/IP was initially thought to offer a quick means of facilitating data transfer between Max and Unity 3D despite it not being supported as a communication protocol within Unity applications running on the HoloLens.

While exploring the TCP/IP protocol, two problems were identified with the interoperability toolkit which was the basis for the original C# script handling the connection between Unity 3D and Max 7. Firstly, the connection between the server, hosted within the Unity application via a subroutine that runs in the update() method, and the client, hosted in the *mxj* object *net.tcp.send*, would drop after a period of time. This may have been caused by calling the subroutine from within the update() method which runs at the framerate and therefore overloads the thread. Secondly, the rate at which data could be transmitted using this protocol seemed to have an upper threshold that could not be adjusted from within the C# scripts adapted from the toolkit, which led to bottlenecking and messages not being received in a manageable timeframe, see Figure 3. For these reasons, the TCP/IP protocol proved not to be a viable solution at the time.

```
ws.OnMessage += (byte[] msg) =>
string text = Encoding.UTF8.GetString(msg);
```

```
IPEndPoint anyIP = new IPEndPoint(IPAddress.Any, 0);
byte[] data = client.Receive(ref anyIP);
string text = Encoding.UTF8.GetString(data);
```

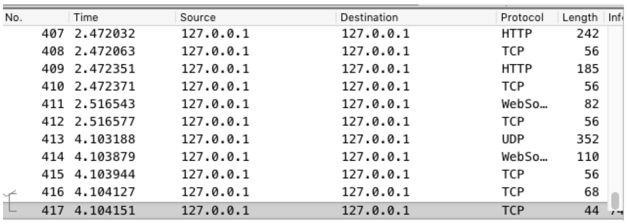
```
var incomingData = new byte[length];
Array.Copy(bytes, 0, incomingData, 0, length);
clientMessage = Encoding.ASCII.GetString(incomingData);
```

Figure 3. C# excerpts showing from top to bottom the endpoints for WebSocket (using *WebSocket Sharp*), UDP (Native support), and TCP/IP (Native support) messages for use in score instantiation and transformation methods.

While the WebSocket protocol [6] offered some functionality reducing network overhead, including allowing two-way communication across a single HTTP connection which doesn't rely on continuous polling, and a standardized handshake procedure that ensures a stable connection between host and client, as the protocol is built on TCP/IP and relied, in the authors' implementation, on the same interoperability toolkit scripts, the authors observed similar issues with respect to dropped packets and message bottlenecking. Alternate methods of facilitating Max/Unity 3D communication via WebSockets such as the WebSocket Sharp package available from the Unity Asset Store were, unfortunately, unable to be used as they had been deprecated during development.

Since the release of Max 8, WebSockets have been supported via Node.js. While WebSockets were not natively supported in Max 7, which the authors had been using for initial development, the integration of Node.js in Max 8 offered an attractive means of transferring data across platforms. After testing WebSockets via Node.js in Max 8 alongside UDP, which has been supported natively in Max since Max 5 [7] for speed and robustness, however, the authors chose to commit to the UDP protocol which was significantly faster and sufficient for the purposes of the project, see Figure 4. UDP's speed of data transfer was, perhaps, unsurprising given UDP's smaller data header size given its lack of error correction.

[2] Curves are rendered as both cubic and quadratic Bézier curves.



No.	Time	Source	Destination	Protocol	Length	Inf
407	2.472032	127.0.0.1	127.0.0.1	HTTP	242	
408	2.472063	127.0.0.1	127.0.0.1	TCP	56	
409	2.472351	127.0.0.1	127.0.0.1	HTTP	185	
410	2.472371	127.0.0.1	127.0.0.1	TCP	56	
411	2.516543	127.0.0.1	127.0.0.1	WebSo…	82	
412	2.516577	127.0.0.1	127.0.0.1	TCP	56	
413	4.103188	127.0.0.1	127.0.0.1	UDP	352	
414	4.103879	127.0.0.1	127.0.0.1	WebSo…	110	
415	4.103944	127.0.0.1	127.0.0.1	TCP	56	
416	4.104127	127.0.0.1	127.0.0.1	TCP	68	
417	4.104151	127.0.0.1	127.0.0.1	TCP	44	

Figure 4. Wireshark screen capture showing faster transfer of FFT data via UDP compared to TCP/IP and WebSocket connections.

The composer's decision to limit data sent between Max and Unity 3D to basic control information obtained from signal analysis, and higher-level Max commands, neither of which was time-critical or of significant volume for dropped packets to be of concern, lent further support to the choice of UDP as the best inter-application communication protocol for the execution of this particular work. Correspondingly, C# scripts for score instantiation within the Unity 3D platform were developed, further mitigating any risks associated with significant data transfer loss through the redundancy offered by allowing hard-coded scores to be used as a contingency during performance.

3. DEVELOPMENT AND DEPLOYMENT

3.1 Basic Scene Properties

On the HoloLens, the color black is used to indicate an absence of light so it is important that musical scores intended for display on the device do not feature black-colored textures as they will be rendered transparent. For these reasons as well, Unity's *skybox*, which colors scenes beyond rendered geometry, must be set to the color black. While red, green, blue, yellow, and white are the only colors featured in *5x3x3*, this provision implicitly privileges color as a structural property of a notational schema, a property it does not traditionally hold.

The size of game objects in Unity 3D are represented in a mixed reality space to scale, i.e. a sphere with an *xyz* scale of [0.25, 0.25, 0.25] has apparent dimensions in the HoloLens mixed reality space of $25cm^3$. Similarly, Unity game objects positioned between *xyz* coordinates [-1. 0. 0.] and [1. 0. 0.] will be distributed across a space of 2 meters. The translation of Unity's transform space to the mixed reality space of the HoloLens thus has an important impact on how those scores are physically engaged with during performance. In *5x3x3* each of the three nodal constructs in their initial instantiation span a space of approximately $60cm^3$ with the three constructs spanning a total space of 3m, see Figure 5. These dimensions were influenced by the physical space required for three performers, affording them the freedom to physically navigate around the score with minimal impact upon each other, but were also influenced by the HoloLens's relatively narrow 35-degree field of view. The narrow field of view is a particularly important consideration as the constructs merge together.

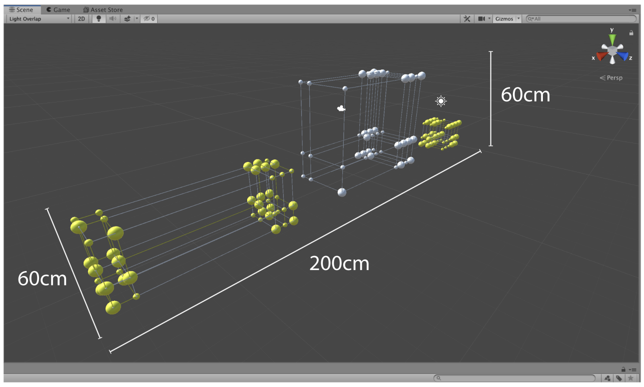


Figure 5. Physical dimensions in the mixed reality space of the score for *5x3x3* in its initial state.

Final, but not trivial consideration for HoloLens development, apply to Unity camera settings. In order to properly immerse the performers in a mixed reality space, Unity's Main Camera must be positioned at *xyz* coordinates [0, 0, 0] and *clip planes*, which determine the point at which a hologram will be culled as physical proximity to the hologram becomes too close, must be set as appropriate. In *5x3x3*, *clip planes* are set to .5m.

3.2 Mixed Reality Toolkit

The Mixed Reality Toolkit (MRTK) [8], released in 2017, offers an invaluable means of accelerating HoloLens application development with perhaps its most useful feature being the ability to deploy a Unity 3D scene to the HoloLens remotely over TCP/IP directly from the Unity platform, see Figure 6. This affordance bypasses the need to build a Visual Studio solution in Unity and deploy that solution to the device in Visual Studio providing considerable development time efficiencies. While the HoloLens emulator can also be used for development testing, it provided less performance feedback than remote deployment via the MRTK and was therefore of less use in the development of *5x3x3*.

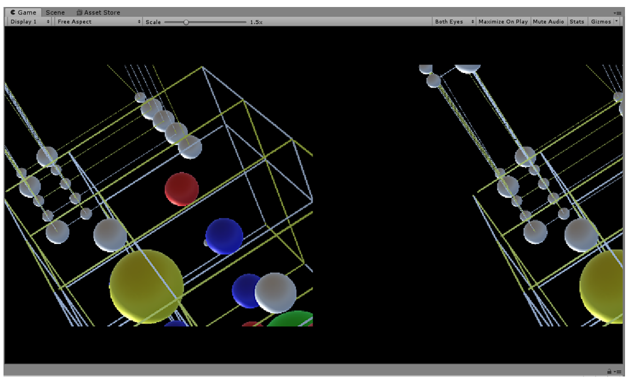


Figure 5. Left and right eye images from a single frame of the score for *5x3x3* generated through the Mixed Reality Toolkit.

In remote deployment a Unity 3D scene runs on the host machine's local CPU rather than that of the HoloLens. This has important ramifications for HoloLens optimization as the HoloLens CPU is unlikely to be as powerful as that on the local machine upon which development occurs. As a result, the scene will not accurately reflect optimization needs when the application is ultimately deployed to the HoloLens.

Aside from accelerating prototyping through remote emulation, the MRTK provides a bundle of scripts and components which improve development efficiency particularly with respect to Unity's Main Camera settings. Correct transform positions are automatically set and other important parameters such as Field of View angles, Clipping Planes distances, Occlusion Culling and Stereo Separation amongst many others may be quickly adjusted for optimal presentation.

Considerable CPU gain efficiencies are also achieved through a proprietary method of stereoscopic rendering termed Single Passed Instance Stereo Rendering. 3D stereoscopy achieves its effect by displaying different images to each eye. Irrespective of whether these images are distinguished through color filtering, the anaglyphic effect, or polarization, most 3D rendering systems employ a multi-pass rendering process where separate draw calls are made for the image projected to the left, and the image projected to the right eye. Rather than use a single draw call for separate right and left eye images, the MRTK renders stereoscopic images through a single draw call which is then multiplexed by the GPU resulting in CPU efficiencies of around 30%, a significant dividend given the relatively slow HoloLens processor.

Holograms may be shared across multiple HoloLens devices. In *5x3x3*, while each of the performers have their own HoloLens, they nevertheless view the same dynamic three-dimensional image which is locally managed on each device. Sharing holograms across multiple HoloLens devices requires a common reference point (*anchors*) be established and for coordinate systems to be shared across devices. This information is shared across a local area network and applied locally via each device's IP address. The MRTK includes components and APIs which keep track of IP addresses and port numbers of each HoloLens but also help manage the initialization of anchor locations and the continued tracking of their positions. Anchors are initially established locally on one device before being uploaded to the shared managing service. The initialization process does have some practical ramifications for musical performance applications, but once anchors are established and shared, the sharing process is seamless.

3.3 Optimization and HoloLens Constraints

The HoloLens has a relatively slow Intel Atom x5-Z8100 1.04GHz CPU which requires Unity 3D scenes be carefully optimized in order to minimize visual artifacts such as jitter (high frequency hologram shaking) and judder (unstable hologram motion) caused by failing to maintain a rendering speed of around 60FPS. [9] Fortunately, for the development of *5x3x3*, these efficiencies could be achieved with minimal impact as a number of commonly used but CPU intensive Unity affordances such as *physics* were not required. Similarly, other HoloLens affordances such as the ability to physically interact with a hologram, speech recognition, or projection mapping were also not required.

Visual artifacts were able to be minimized due to some inherent properties of the score for *5x3x3*. With a basic color palette (red-green-blue-yellow-white), color separation, where colors are not accurately mapped during hologram tracking, was never an issue. CPU load was further minimized through the use of standard Unity 3D shaders with no textures employed.

3.4 Application Building and Deployment

The final stage of the development process for *5x3x3* was the relatively straightforward one of building a Visual Studio solution, simply requiring the selection of the appropriate target platform (Universal Windows Platform) and latest SDK, and the deploying of that solution to the HoloLens. At the deployment stage, the application is installed onto the HoloLens and can be re-opened locally at any later time without having to deploy the application to the device again or upload the application to the Microsoft Store.

4. PERFORMER FEEDBACK

Initial testing of the HoloLens with performers provided invaluable feedback for development and application optimization. With respect to this, it is important to distinguish between the performance challenges inherent to interpreting generative graphic scores, [10] which this paper will not consider, and those presented by the HoloLens hardware itself.

In order to realistically project stereoscopic images, the HoloLens needs to be calibrated for the interpupillary distance of the wearer. As this distance is unique for each person, visual artifacts and incorrectly placed holograms may result if the calibration process is not properly completed. While calibration did not present any difficulties with those performers with whom testing was conducted, the process was usually a performer's first experience performing with a headset and with the ergonomics of the HoloLens in particular.

Weighing just under 600 grams and with an adjustable headband, finding a position that was physically comfortable while minimally impacting performance presented the greatest initial challenge. While the ergonomics of the original device have been improved on the HoloLens2 which features a swivel hinge allowing the lens to be raised and lowered, testing for *5x3x3* was only able to be completed on the original device.

While performing while wearing a headset was an unusual experience for most performers, the greatest challenge presented by *5x3x3* was with respect to the requirement to minimize quick head movements, a particularly difficult challenge for wind players. Unless movements were kept smooth and relatively slow, performers commented on a tendency for visual artifacts, notably image jumpiness and flickering to be introduced. While adjustments to focal planes was able to address this particular issue to a certain extent, the problem was exacerbated by the requirement in *5x3x3* for performers to physically explore the various three-dimensional pathways through the score by walking around the projected hologram. The freedom afforded by the dérive-like journey through the musical possibilities of *5x3x3*'s score was thus tempered by an unexpected physical discipline required for the scores' stability.

Finally, with respect to image readability, performers commented on how lighting conditions in rehearsal venues needed to be carefully adjusted to ensure the projected hologram was not washed out.

Figure 6. Performance testing.

5. FUTURE DEVELOPMENT

The HoloLens offers exciting creative possibilities for musical practice foremost of these, perhaps, extending from a foregrounded physical relationship between performer and notational schema. In *5x3x3*, this relationship is established through the need for the performers to physically move through the performance space in order to discover unique musical pathways through the score. In its originally developed screen-based realization, such an aesthetic affordance was clearly not a possibility.

While not explored in the development of the score for *5x3x3*, other HoloLens affordances for works specifically designed for the device include those drawing on the ability to projection map a score onto a physical environment. The HoloLens employs a spatial mapping process to assist with consistent and natural placement and interaction with holograms. This offers the possibility of translating a notational schema to spatial surfaces within a real space. The nodes in *5x3x3* could, for example, be aligned with the physical dimensions of a table or any other object within a space, drawing correlations between spatial geometry and musical form in a manner reminiscent of works based on architectural geometries such as Xenakis's *Pithoprakta* (1955-56) or *Metastasis* (1953-54).

Interactive possibilities with a hologram offer further opportunities for the design of unique musical forms and structures. While certain practical difficulties involved with the ability to perform with an instrument at the same time as interacting with a hologram through gestural input must be resolved, the musical possibilities are nevertheless of interest, extending the possibilities of *reactive scores* [11] in many exciting ways.

6. CONCLUSIONS

Porting the score for *5x3x3* to the HoloLens presented a number of expected and unexpected challenges, not least of which involved the challenges of efficiently and accurately transferring data from Max to Unity 3D. The realistic display of image depth addressed the most pressing performance challenge while foregrounding to a far greater extent the performer's physical engagement with the score. While the aesthetic value of this engagement had not been fully considered during the original development of *5x3x3*, it came to play an increasingly strong role. It also emphasized one particular concern with mixed reality systems in performance, namely that involved in obfuscating the mixed reality space navigated by performers from that in which the audience is physically immersed. While in some ironic respects this mode of performance approaches that of traditional musical performance in which scores are not foregrounded, it nevertheless presents a performance space in which the audience is inherently distanced. These issues notwithstanding, and as suggested in Section 5, the HoloLens does offer unique affordances for musical expression which we hope to explore in future work specifically designed for the device.

Acknowledgments

The development of *5x3x3* was supported by a grant received by the composer from the Australia Council for the Arts and a research residency at IRCAM undertaken in January 2018.

7. REFERENCES

[1] D. Kim-Boyle, "The 3-D Score," in *Proceedings of the International Conference on New Tools for Music Notation and Representation (TENOR'17)*, A Coruña, 2017, pp.33-38.

[2] R. Leung, A. Lasso, M. S. Holden, B. Zevin, and G. Fichtinger, "Exploration Using Holographic Hands as a Modality for Skills Training in Medicine," in *Proceedings of SPIE*, Vol. 10576, *Medical Imaging 2018: Image-Guided Procedures, Robotic Interventions, and Modeling,* 13 March 2018. Available: https://doi.org/10.1117/12.229549

[3] P. Hockett and T. Ingleby, "Augmented Reality with Hololens: Experiential Architectures Embedded in

the Real World," 13 October 2016. Available: arxiv.org/abs/1610.04281

[4] I. Bukvic and J.-S. Kim, "µ Max-Unity3D Interoperability Toolkit," in *Proceedings of the International Computer Music Conference 2009*, pp. 375-378.

[5] B. Carey, "SpectraScore VR: Networkable Virtual Reality Software Tools for Real-time Composition and Performance," in Proceedings of the International Conference on New Interfaces for Music Expression 2016, pp. 3-4.

[6] I. Fette and A. Melnikov, "The WebSocket Protocol," RFC, vol. 6455, 2011, pp. 1-71.

[7] Cycling 74, *Communications Tutorial 3: UDP Networking.* 2008. Available: https://docs.cycling74.com/msx5/tutorials/max-tut/communicationschapter03.html

[8] Microsoft, *MixedRealityToolkit*. 2018. Available: github.com/Microsoft/MixedRealityToolkit-Unity

[9] Microsoft, *Hologram Stability*. 2018. Available: microsoft.com/en-us/windows/mixed-reality/hologram-stability

[10] J. Freeman, "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Computer Music Journal*, Vol. 32(3), 2008, pp. 25-41.

[11] P. Rebelo, "Notating the Unpredictable," *Contemporary Music Review*, Vol. 29(1), 2010, pp. 17-27.

ANIMATED NOTATION, SCORE DISTRIBUTION AND AR-VR ENVIRONMENTS FOR SPECTRAL MIMETIC TRANSFER IN MUSIC COMPOSITION

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

Benedict Eris Carey
Hamburg University of Music and Theatre
Hamburg, Germany
benedict.carey@hfmt-hamburg.de

ABSTRACT

This paper seeks to make a case for a compositional ideal (the mimetic transfer of a recorded or synthesized sound to the instrumental/vocal domain) which today's technologies for animated/distributed musical notation have made more realistic than when it first appeared as a general aesthetic (with composers such as Tristan Murail and Gérard Grisey, or, in the realm of computer music, as with the practice of Jean-Claude Risset), simultaneously with the birth of the digital era in the 1970s. The concept of *mimesis* is here examined both as a (post) spectral compositional technique and as a common feature of many forms of musical score/representation. These theoretical considerations are then exemplified by musical examples and software demonstrations extract from the "*In memoriam Jean-Claude Risset*" cycle of compositions, scored for ensembles of various sizes (small chamber music group with players wearing head-mounted displays, choir and electronics, large instrumental groups with choir, and for the performance of an opera), all performed with the help of the SmartVox Score distribution system.

1. INTRODUCTION: MIMESIS

Just as mimesis was a governing principle of artistic creation in Ancient Greece, it was most certainly also at the heart of the French spectral movement of the 1970s. In this context, composers rarely put into question why for instance the faithful imitation of the harmonic spectrum of a bell sound qualified as art. Assuming that harmony in music is paramount, one can now examine how, in the case of singers for instance, this mimetic transfer might be facilitated when one receives an audio-score [1] through his/her ear, reproducing microtonal harmonies by just imitating the perceived pitch (as is the case with SmartVox[2]). In this case the (auditory) notation closely resembles the desired effect. However, Nelson Goodman demonstrated that this is not necessarily the case with musical notation: in fact even beyond the musical realm, the philosopher is famous for showing that one thing need not resemble another in order for that "other" to represent it. Working on real-time animated music notation and virtual reality [3], the the author has remarked in the past of an essential aspect of musical notation: '*the closer a musical unit gets to representing a direct action (that is, the movement of an object in space) the more* ***mimetic*** *it becomes*.' [4] This observation highlights the great complexity of rhythmic notation in common practice musical notation, when compared to modern video game-like notations such as, for instance, *Digital Audio Workstations* where a cursor scrolls from left to right, or in the popular arcade and video game *Guitar Hero* and modern piano-roll notation, where notes or tablature fall from top to bottom of the screen (a practice known as animated notation). In traditional western music notation on the contrary, the historically inherited practice of the arithmetic division of tuplets, their distribution into bars and beats, and the constant adjustments instrumentalists must make to coordinate with the gestures of a conductor, show as a whole how *mimetic* representations are—for better or worse—much easier to understand for beginners than the culturally inherited *memetic* (i.e. symbolic) common practice.

2. MIMETICS VS MEMETICS

2.1 The Memetic Nature of Animated Notation

When we score music by hand, we do so to varying degrees of mimetic accuracy, albeit often without reference to essential *extra-musical* information required to interpret the music, we are heavily engaged in a *memetic* process. The authors propose that an animated score can be described, in some part, by where it exists in terms of its *mimeticness* and its *memeticness*. Guitar tablature, and *Guitar Hero* make use of very *mimetic* forms of notation. That is, notation which very closely related to the (prescribed) action to be performed, and less bound by the semiotic or *memetic* information one finds in common practice notation, which imply abstract representations engaged with descriptions of the "end-result" sound, allowing for a wider range of possible interpretations, more so with graphic notation, and even more so in more abstract forms of animated notation. The authors believe that 'Mimetics' is therefore a useful term for the field scores which rely on imitation, to convey their desire for audification (a term attributable to Lindsay Vickery [7] and the author [4]).

2.2 The Limits of Mimetic Replication

The term *mimetics* serves as a broader term encompassing the study of all forms of replication. Richard Dawkins' seminal and influential concept of *memetics* takes this idea further into the domain of semiotic thought. Dawkins claims that since universal Darwinism suggests that replication and natural selection are the primary forces through which the universe comes into being, memes are but a tiny part of a far grander vision. Computer music is increasingly pollinated with ideas from all over the spectrum of the sciences; an art-science [5] in emergence as it is referred to by Gérard Grisey. So how do we dismantle and rebuild our understanding of the musical score into the 'essential' or 'base' form that analyzing animated notation in relation to static notation demands? One that isn't bound by two dimensions X and Y (generally seen as time and pitch in the score space), which would be far too specific for a multimedia focused compositional system to give but one example. As described by John Blacking in his book *How Musical is Man*, music contains a qualitative dimension. Would a non-systematic approach to animated notation be better suited to tackling this challenge? Here he describes with exactitude how cultural information is essential to our understanding of music:

'Consider the matter of 'feeling in music,' which is often invoked to distinguish two technically correct performances of the same piece. This doctrine of feeling is in fact based on the recognition of the existence and importance of deep structures in music.' [6]

Why people seek to represent through abstraction is not a total mystery, yet untangling the twisted ropes of meaning wrapping up every idea anyone has ever conceived probably presents a completely unsurpassable challenge. The utopian desires of universal Darwinism will likely not be realized through ever increasingly complex quantizations of the natural elements, and while defining moieties in an artistic context may be seen as futile by some, the so-called 'memes' that comprise the universal replicators of meaning imagined by Richard Dawkins and significantly expanded upon by Susan Blackmore (even to the point of suggesting that the meme is being superseded by a new unit, the 'teme'), present a very real and useful concept when applied to animated notation analysis.

2.3 Defining Mimetic and Memetic Scoring Systems

As mentioned, we propose that closer a musical unit comes to representing a 'direct action', the more *mimetic* it becomes. Within the scope of this definition the most *mimetic* score achievable, is a person performing any copyable action. Further, this paradigm can be abstracted into the domain of video, and through acusmatic methods, sound exclusively (such as is the case in many aural musical traditions for example). Lyrebird's and other parrots engage in this practice also, often with a far greater degree of accuracy than many human musicians. Further levels of abstraction can be seen in Ryan Ross Smith's particular brand of animated notation involving radials, cursors and other devices designed to control complex rhythmic coordination in ensemble contexts. Cat Hope's scrolling scores introduce a similar kind of mimesis, as do David Kim-Boyle's AR and 3D scores. Common practice notation introduces a higher degree of assumed knowledge and therefore is heavily dependent on *memetic* transfer, so in the same sense, direct copying of a gesture resulting in sound is a *mimetic* transfer, done without requiring a theory of mind, but once the sound begins to stimulate the physiological, auditory system of someone memetic and semiotic processes spring into action.

3. MIMETIC ORCHESTRATION

With the exception of the first opus (***In Memoriam JC Risset 1***), whose pitch material was extracted from the analysis of FM synthesis spectra, all the pieces of the cycle presented below constitute *mimetic* orchestrations of the PRISM laboratory synthesizer, developed by Richard Kronland-Martinet [8], who worked for thirty years with Jean-Claude Risset at LMA, Marseille, France.

This synthesizer is based on a perceptive model (as opposed to a physical model [9]) and carries the influence of Risset's exploration of timbre synthesis, examined through the perspective of human perception [10]. The great majority of the electronics in other pieces of the cycle (*In Memoriam 2, 3 and 4*) makes extensive use of the *liquefaction* parameter, which turns a continuous sound into small grains evoking the resounding impacts of drops on multiple resonant metallic surfaces, or some form of imaginary rain on metallic chimes. [1] The *Bach* [11] library was very helpful for musical transcription, subsequently allowing for transfer to the instrumental domain.

The authors forthcoming doctoral thesis 'Between Mimetics and Memetics' examines the credo assumed by various spectral composers, for whom the very act of composing was equivalent to translation of sounds from the acoustic to the instrumental domain. Classical contemporary music scenes such as those represented in Europe by the Gaudeamus or Darmstadt academies and festivals reveal, however, that such preoccupations are often today very far removed from what young generations of composers seemingly wish to express. Therefore, one can only talk of neo or post-spectralism when referring to such techniques. Whilst still taught in institutions like IRCAM, or briefly mentioned as part of the undergraduate composition curricula, one can deplore the fact that this aesthetic canon has seen its day now that technologies (such as the tools presented in the present paper) finally allow more intuitive and user-friendly interfaces to the composer (Bach, Ableton, PRISM synthesizer), auditory and visual computer assistance for microtonal intonation and synchronization adjustments to the performer (SmartVox [2]), and thus probably more convincing results from an audience's point of view.

3.1 Time Domain: Temporal Precision and Mimetic Transfer

At IRCAM in January 2018, Benjamin Matuszewski highly improved the synchronization possibilities of SmartVox

[1] A *liquefation* demonstration is available here: https://youtu.be/2kdIaqAhUGs

by implementing a client-side algorithm that puts back in sync devices whose drift exceeds a certain threshold ([12], Chapter 4.3). Until this new release, SmartVox had to deal with a more approximate temporality, but it can now explore a wide range of tightly synchronized musical situations, between musicians, electronics and video for instance.

In none-pulsed music, in spite of the great conducting tradition in chamber music and orchestral works, left-to-right scrolling cursors distributed on the performers' devices (as seen in the Decibel ScorePlayer or SmartVox) seem a far more straightforward strategy to obtain tight synchronization rather than the bars and beats 'encoding' (quantification by the composer with or without the help of algorithms) and decoding processes (a compromised interpretation by the instrumentalist, between the rhythmic values written on the page and the gestures of the conductor), inherited from a scoring tradition in which a regular meter was assumed.

The bach.roll object displays notation in proportional time, and outputs notifications of its playback status in real-time. These notifications can be interpreted in *Max For Live* in order to synchronize 1/ the notation for human players in *Bach* and 2/ the electronics in Ableton Live. Figure 1 shows how Ableton's playback controls can be accessed through the live-set path of the live object model (LOM), which makes constant back and forth playbacks possible between the score and the electronics, during the compositional process. This interface facilitated the composition of *Mit Allen Augen, In Memoriam J.-C. Risset 2*, a piece involving large forces performed in Paris in March 2019 (12 voices, 12 instruments and electronics). [2]

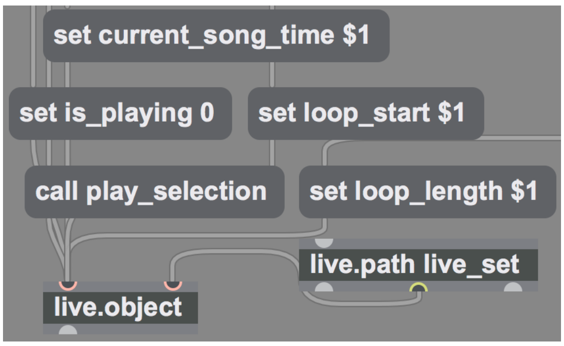


Figure 1. *Max For Live* device syncing *Bach* and Ableton.

Pieces of the Risset cycle all benefit from this improvement: in the first piece of the cycle, a passage is based on rhythmical games between performers, taking advantage of the cursor-type display of time. [3] In ***Das Hoheslied, In Memoriam Jean-Claude Risset 3***, premiered in June 2019, students of HfMT Hamburg were able to sight-sing (for the very first time) passages which strikingly resembled the target model. [4]

3.2 Frequency Domain: Harmonic Analysis of Frequency Modulation

Rather than citing the oft-used Shepard-Risset glissando illusion, the author found it more appropriate to re-activate one of the most successful techniques of the beginning of computer music: frequency modulation, which Risset was amongst the first composers to use in his piece Mutations (1969), thanks to Chowning's generous permission. This is why nearly all the harmonic material from *In Memoriam J.C. Risset 1 and 3* was generated by FM synthesis.

For frequency analysis, although *Iana* [13] was formerly used (and also favored by Daniele Ghisi [11] in some of his transcription patches), the *zsa.freqpeak* descriptor [14] offered seemingly more accurate results. A *Max For Live* device was built for pitch visualization purposes. [5]

As a reference to Risset's findings regarding the temporal evolution of the harmonic content of digitally recorded trumpet sounds (which becomes richer in high frequency harmonic when loudness increases [15], the evolution of frequency's 'Modulation index' [6], was orchestrated by successive entries of voices, from the lowest register (bass) to the highest (soprano). [7]

3.3 Conclusion on Mimetic Spectralism and Vocal Imitation

Some of the tools presented in this chapter show that the possibilities offered by today's technologies would have appealed to the supporters of the 'mimetic spectralism' imagined in the 1970s. If the compositional aesthetics followed by these composers is not to everyone's taste, they may at least provide a fertile ground for artistic/scientific research, because their aims and methods, concerned with the reproducibility of a model, are suitable to measurements and, for instance, quantifiable assessments. [8]

In a more scientific context, members of the PRISM [9] laboratory undertook an experimental study of vocal imitation which recalls the sight-reading experiment in *Das Hoheslied*. According to the author, Thomas Bordonné, this study "*aimed at determining the main characteristics of sounds used by participants during vocal imitations*". Bordonné concludes: "*Vocal imitations seem to be a good tool to access perception and determine which aspects of the sounds are relevant*" [16]. In this setup, therefore, the spontaneous vocal response of participants could be interpreted in similar ways to the SmartVox-led reading sessions in which the singers are asked to imitate what they hear.

[2] The following example shows the convenience of the Bach/Ableton inter-application communication https://youtu.be/VJvY5wYl_cM

[3] A rhythmical passage in *In Memoriam Jean-Claude Risset 1* https://youtu.be/hQtyu1dcCaI?t=349

[4] A *Target sound* is followed by its imitation by the choir in a sight-reading session of *Das Hoheslied, In Memoriam Jean-Claude Risset 3* https://youtu.be/EHYq9nFF6sE

[5] Real-time spectral analysis with *Bach* and Zsa.FreqPeak https://youtu.be/D6mCgx4pSxs

[6] See https://youtu.be/OnT-Zgkh5MA for demo purposes.

[7] See https://youtu.be/sgSjIpSD8yQ.

[8] As, for instance, in the aforementioned Hambourg's choir sight-reading example: ***Das Hoheslied, In Memoriam Jean-Claude Risset 3***.

[9] https://www.prism.cnrs.fr/

4. COMPOSITIONAL/PERFORMATIVE EXPERIMENTS WITH ANIMATED AND DISTRIBUTED NOTATION

4.1 Global Context

Ryan Ross Smith's ***animatednotation.com*** website demonstrates that today many composers find fixed common practice notation limiting. To solve this issue, attendees of the Tenor Conference [10] have proposed elements of response. With Kagel's *Prima Vista* (1962/64) setting a major precent, animated notation often relies on a large screen projecting the parts, as most famously exemplified in Smith's compositional practice. The score can therefore be seen by the performers but also by the audience, thus making notation part of the theatrical performance.

Other performance-oriented systems (SmartVox [2] [12], Zscore [17], Decibel [18] [19], Maxscore [20], comprovisador [21]) endeavor to distribute and synchronize each part of the score on the performer's devices (whether Smartphones, tablets or laptops). With SmartVox for instance, rendering the score in the browser directly revealed itself to be very effective for this kind of setup. Thanks to cross-platform web technologies, the application works with any browser capable device, and no installation is required by the client. The node.js [11] /websocket architecture of SmartVox will hopefully inspire more composers, researchers and developers to investigate the emerging musical practice of distributed notation.

4.2 Graphical Notation in *Bach*

Most pieces performed with SmartVox rely highly on *Bach* [11], a realtime computer-aided composition package for Max/MSP. The use of color and shape in pieces like ***In Memoriam JC Risset 1*** or *Mit Allen Augen (**Mit Allen Augen, In Memoriam JC Risset 2**)* was inspired by Cat Hope [22] and Lyndsey Vickery [23], two active composers, researchers, developers of the Decibel ScorePlayer [19], whose approach to animated graphical notation opens new territories for musical scores today.

Unlike other notation packages in Max (MaxScore [20], or the Symbolist [24]) which were imagined as symbolic representations in the first place (Rama Gottfried describes the Symbolist as a version of adobe illustrator in Max), the *Bach* environment emerged from *Open Music* [25] and computer-aided composition, which explains some of its limitations from a graphical point of view: for instance, it is impossible to insert a picture in a bach.roll (in proportional notation), and the bach.roll object does not support articulations (unlike the bach.score object in bars and beats notation). However, one can get around this graphical limitation by adjusting the duration line settings (see Figure 2). The use of colors is also easily customizable with the help of 'slots'.

In spite of its few display limitations, one of the strengths of *Bach* consists of its ability to control synthesizers from the notation directly. Slots can therefore be understood as metadata or temporal automations that can subsequently be

[10] http://www.tenor-conference.org/

[11] Server-side JavaScript, see: https://nodejs.org/en/

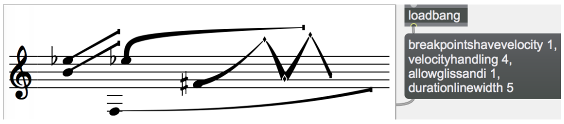


Figure 2. Duration line adjustment in *Bach*.

mapped onto any Max-programmed synthesizers, enabling staves of a score to be manipulated as tracks of a DAW [12] .

4.3 Distributed Musical Notation

Composers and researchers increasingly acknowledge the strong analogy which can be drawn between the traditional 'score and parts' musical practice led by a conductor, and the modern distributed systems or web applications (Zscore [17] - MASD [26] - SmartVox [2]), in which multiple clients coordinate their actions by passing messages to one another.

The Tenor 2018 conference in Montreal revealed an interesting similarity between the Decibel ScorePlayer and SmartVox, in the sense that the two software packages are rather elementary solutions both converging towards the *score player* performance-oriented paradigm, whilst other notation packages (Bach, Maxscore, Symbolist...) were also designed for more elaborate computer-aided composition and real-time processes. Although the Decibel ScorePlayer is today's most reliable iOS application solution for distributed animated notation, Cat Hope and Aaron Wyatt have evoked the possibility to migrate their system to the browser, for instant access over the internet presumably, but also optimization of networking issues, and cheaper production cost in local settings—e.g. for projects involving over fifty devices, the iPad becomes a very expensive solution). The most likely architecture to allow this transfer of technology would be a node.js server [13] (server-side javascript), with the WebSocket communication protocol, a solution used by SmartVox as early as 2015, thanks to Norbert Schnell and Benjamin Matuszewski from IRCAM in Paris. SmartVox sends and synchronises mp4 audiovisual scores, which has demonstrated undeniable robustness in large scale concerts and rehearsals, but also reveals the potential weakness of a 'non-realtime' solution (as a sort of fixed multichannel tape) in which performances of the piece would be similar each time. Other composers/researchers (Georg Hajdu, Rama Gottfried, Slavko Zagorac) are currently investigating forward thinking solutions which will allow the control of SVG (Scalable Vector Graphic) directly on the client-side HTML page. The authors MaxScore.Netcanvas object acted as a precursor to a later project which made use of this notion, through performance of the 50 part real-time generated piece *Magnetic Visions 50* relying on WebSockets to transfer instantaneously generated score fragments. The architecture chosen for this by Georg Hajdu and Rama Gottfried is node.js, sending osc over WebSockets. This project resulted in a realtime distributed 144-part site-specific composition in Hamburg in May 2019, which was justifiably well received due in part to its technically ambitious and

[12] see https://youtu.be/s4qS2khwkT0 for demonstration.

[13] According to a private email conversation with the developper Aaron Wyatt.

groundbreaking nature.

4.4 Augmented Reality Distributed Notation

In Memoriam JC Risset 1 [14], premiered in September 2018 at the Gaudeamus Festival (Utrecht), constitutes the author's first experiment using head-mounted displays for notational purposes, principally inspired by Benedict Carey's *SpectraScore VR* [3] in this regards. By simply displaying each part of the score over the heads of the performers (for flute and clarinet only), this piece has revealed interesting potentials to be exploited in future works.

Just as traditional scores placed on a music stand, screen-scores displayed on a tablet (for instrumentalists) or on a phone (for singers) oblige musicians to look and orientate their body constantly in the direction of the score. This well-established convention of the classical concert setup considerably limits the possibilities of staging music, in a theatrical context for instance. In such cases as the Ictus Ensemble's interpretation of *Vortex Temporum* by Gérard Grisey [15], where the musicians had to learn the score by heart, and perform without a conductor, Wi-Fi-synchronized head-mounted displays might be an interesting way to help musicians coordinate in time and space, while moving on stage or around the audience, without insurmountable performance challenges. The idea was therefore explored further on a larger scale in ***Mit Allen Augen, In Memoriam JC Risset 2***. [16]

According to the performers' feedback, head-mounted setups provide a large and comfortable display, since the environment is still visible around the score (see Figure 3), or through the score in the case of holographic display (such as the Aryzon headset for instance). The performers also showed interest in their ability to move freely on stage or elsewhere (the piece was choreographed differently according to the venue: in a church, the flute and clarinet started the piece behind the audience and gradually approached the altar), which relates to the term *phygital* (physical+digital) coined by Fabrizio Lamberti [27], who claims that with AR and VR, the possibilities of physicalization of gaming will soon encompass other fields.

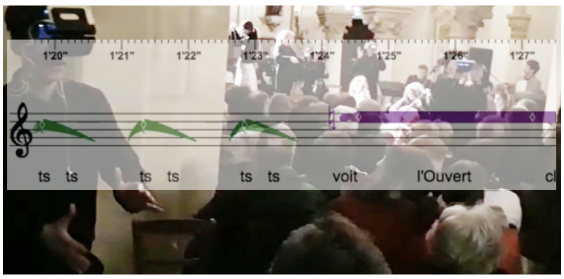


Figure 3. Score display from the performer's point of view.

While awaiting more discrete (such as Google glasses, which would invisible from an audience's point of view) but still affordable solution, AR opens vast domains for further research, which could go far beyond the mere display of animated notation. However, important questions then arise regarding what would be pertinent to achieve with such technology: in a VR context for instance, with a player on stage, how can the limitation due to the destabilization of the performer's proprioception be overcome? Also, now that any form of immersive score can be prototyped, [17] how would this be of musical interest, and how could this have a convincing musical impact on the compositional/performative outcome? SmartVox, for instance, delivers and synchronizes mp4 files, and should soon support 360-degree videos [18], but how can this benefit the performer in a musical sense? Australia holds today one of the world's largest contingents of composers/researchers working in animated and XR notation, and Australian composers such as The author [3] and David Kim Boyle [28], who specifically investigate 3D-scores in VR environments, present elements of response to this issue.

5. NETWORKED MUSICAL PERFORMANCES - NMPS

5.1 SmartVox.eu, Score Distribution over the Web and Remote NMPs

With the exponential growth of the web, hosting **Smartvox** [19] on the internet (i.e. on a remote server rather than a local one) appeared as a necessity. However we demonstrated (in Chapter 4.2 'Measurements of timing accuracy' [12]) that, although the synchronization of different parts was quite accurate in this way, local solutions remained safer (in the same room, over Wi-Fi). Also, notwithstanding the fact that SmartVox undoubtedly belongs to the realm of Networked Music Performances, it became clear that its practical application falls under the 'local NPMs' sub-category defined by Gabrielli and Squartini [29].

Since the advent of the internet, thorough research has been undertaken in the realm of remote Networked Musical Performances (NMP) [30]. Today however, local NMPs still seem today more viable, both technically and artistically speaking. Despite many 'millennials' being known for their enthusiastic engagement in online gaming (i.e. remotely), with friend avatars or anonymous players often distributed widely around the globe, many examples would show that musicians still feel more inclined to reading a score or improvising together in the same room. In *Embodiment and Disembodiment in Networked Music Performance* [31] Georg Hajdu explains that "*appreciation [of the musical experience] relies on the plausibility between physical action and sonic result.*" Since the emergence of NMPs, [20] some disappointment arose from the fact that, from the audience's

[14] A recording of the piece is available at the following address: https://youtu.be/hQtyu1dcCaI

[15] See: https://youtu.be/JFvrYy6EeWE

[16] A recording of the piece is available at the following address: https://youtu.be/ET$_O BgFWx$04

[17] Such as the ones imagined by Mauricio Kagel's film *Ludwig Van* (1969), see for instance: https://youtu.be/7l8vPWFIgxI?t=1591

[18] This could be done by simply implementing a 360 mediaplayer in HTML5, e.g.: https://bitmovin.com/demos/vr-360

[19] Each instrumental/vocal part of the piece *And the Sea* is accessible through the following url www.SmartVox.eu, and can be accessed simultaneously from e.g., an iPad for the flute, Android tablets for piano and cello, and a phone for the singer. A trailer of the premiere of the piece is accessible here: https://youtu.be/prcXUbhd-ZY

[20] The American computer music network band 'the Hub', formed in 1986, contributed to the popularization of this new genre.

point of view (but also for the musicians themselves), very little can be seen and therefore understood, which worsens with distant performers: "*Because of the remoteness of the participants locations, these actions may not always be perceived directly or immediately [...], classical cause-and-effect relationships [...] are replaced by plausibility, that is the amount to which performers and spectators are capable of 'buying' the outcome of a performance by building mental maps of the interaction. In NMP, this can be facilitated by the use of avatars, projected visually, and carefully orchestrated dramaturgies, involving participants in game-like scenarios.*" Many VR environments today focus on this interaction between a physical action and its sonic result. *New Atlantis* ([32]), for instance, is a multi-user sound exploration platform in which several players can interact with each other. [21] With their ability to represent avatars of musical performers, these new interfaces may soon have the ability to recreate the visual entity producing sound (or music). By recreating virtual causal links between an action and its sonic result, these online game interfaces should provide users with the missing visual element that prevent NPMs from becoming a genuine globalized musical practice.

5.2 The 'BabelBox', a Raspberry Pi Hardware Embedded System Solution for Local NMPs

SmartVox and MaxScore.Netcanvas have proved to be suitable for large scale projects such as *le temps des nuages* [22] and Magnetic Visions 50, in which eighty singers and musicians [12] and fifty musicians had their score synchronized through the same network respectively. These productions still enjoy increasing interest from choirs and ensembles, [23] requiring a very modest technical setup (only one Wi-Fi access point, e.g. Ubiquiti Unifi) for choirs up to thirty singers, with a node.js server running on a Mac computer. However, in the more intimate context of chamber music (as in ***In Memoriam JC Risset 1***), the focus is slightly different: the network load is much lighter, which encouraged finding a minimal hardware solution, in order to make possible rehearsals without the physical presence of the composer or the mobilization of a technician only to setup a network. Installing the server on the performer's computer remotely has often been successful with musicians unfamiliar with technology, thanks to the flexibility of the Node Package Manager (NPM), which reduces the installation of SmartVox to a few command lines. [24] This installation process nevertheless remained an obstacle for the dissemination of SmartVox. 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 from the dedicated router (e.g. tp-link...).
- Starting SmartVox at boot.

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 (less than 65 €, for a Raspberry Pi and a router) 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, the system was rented to the Caen French conservatoire via BabelScores [25] , thus proposing a rental of performing scores (separate parts) of a new kind.

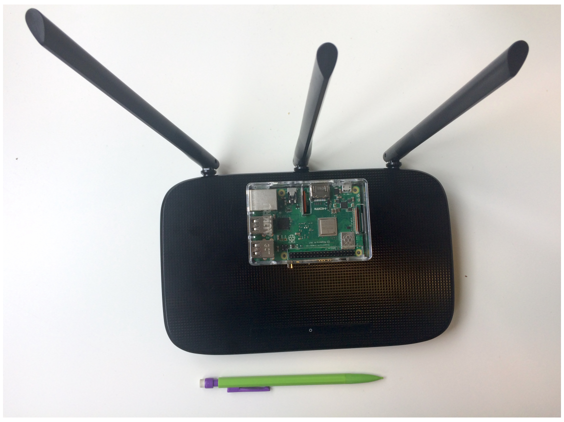


Figure 4. Score display from the performer's of view.

6. CONCLUSION

Animated notation, score distribution, AR and VR technologies were here presented as tools for composition and performances in which *mimesis* is examined under multiple notational perspectives. The *mimetic* value of notation relates, on the 'prescriptive' side, to its ability to mimic an instrumental gesture (such as the representation of the position of fingers on tablature notation), but also, on the 'descriptive' side, to its ability to help performers imitate

[21] Towards the end of the following extract, three players can be observed producing sounds together: https://vimeo.com/264626943

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

[23] SmartVox was used for the rehearsals and performance of *To See The Invisible*, an opera by Emily Howard, Aldeburgh, Snape Maltings 2018https://snapemaltings.co.uk/concerts-history/aldeburgh-festival-2018/to-see-the-invisible/

[24] SmartVox is open source and ready to download via GitHub: https://github.com/belljonathan50/SmartVox0.1.

[25] Babelscores (https://www.babelscores.com/) is an online score database for classical contemporary music, currently actively supporting the SmartVox project: http://1uh2.mj.am/nl2/1uh2/lgi4u.html.

a *target sound*, as exemplified in the workflow of spectral composers [26]. Initially conceived almost exclusively as a rehearsal tool for choral practices [2] [12], recent use cases have shown that SmartVox is better described as a *distributed score player*, allowing for vocalists, instruments and electronics or multimedia to interact in a *Networked Music Performance*.

Acknowledgments

We would like to thank Cat Hope, Lyndsey Vickery, Georg Hajdu, Daniele Ghisi, Benjamin Matuszewski, Norbert Schnell and Richard Kronland-Martinet.

7. REFERENCES

[1] 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/

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

[3] B. E. Carey, "Spectrascore vr: Networkable virtual reality software tools for real-time composition and performance," in *International conference on New Interfaces for Musical Expression (NIME), Brisbane, Australia*, 2016.

[4] ——, "From mimetics to memetics," Ph.D. dissertation, Hochschule für Musik und Theater Hamburg, (unpublished).

[5] C. Arrell, *Pushing the Envelope: Art and Science in the Music of Gérard Grisey*. Cornell University, Aug., 2002. [Online]. Available: https://books.google.fr/books?id=lzJ3GwAACAAJ

[6] J. Blacking, *How Musical Is Man? (Jessie and John Danz Lectures)*. University of Washington Press, 1974.

[7] L. Vickery, "Hybrid real/mimetic sound works," 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. 19–24.

[8] S. Conan, E. Thoret, M. Aramaki, O. Derrien, C. Gondre, S. Ystad, and R. Kronland-Martinet, "An intuitive synthesizer of continuous-interaction sounds: Rubbing, scratching, and rolling," *Computer Music Journal*, vol. 38, no. 4, pp. 24–37, Dec 2014.

[26] composers for who the rhythmic and harmonic characteristics of a recorded or synthesised sound serve a as basis for composition and orchestration

[9] N. Ellis, J. Bensoam, and R. Causse, "Modalys demonstration," in *Proceedings of International Computer Music Conference*, Barcelona, Spain, Sep. 2005, pp. 101–102, cote interne IRCAM: Ellis05a. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01161344

[10] J.-C. Risset and D. Wessel, "Exploration of timbre by analysis and synthesis," in *The Psychology of Music*, D. Deutsch, Ed. Academic Press, 1999, pp. 113–169. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00939432

[11] 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.

[12] J. Bell, "Audiovisual Scores and Parts Snchronized Over The Web," in *TENOR 2018*, Montreal, France, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01779806

[13] T. Todoroff, É. 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 *Proceedings of the 1995 International Computer Music Conference, ICMC 1995, Banff, AB, Canada, September 3-7, 1995*, 1995. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.1995.088

[14] M. Malt and E. Jourdan, "Zsa.Descriptors: a library for real-time descriptors analysis," in *5th Sound and Music Computing Conference, Berlin, Germany*, ser. 5th Sound and Music Computing Conference, Berlin, Germany, Berlin, Germany, Jul. 2008, pp. 134–137. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01580326

[15] J. C. Risset, "Computer study of trumpet tones," *The Journal of the Acoustical Society of America*, vol. 38, no. 5, pp. 912–912, 1965. [Online]. Available: https://doi.org/10.1121/1.1939648

[16] T. Bordonné, M. Dias-Alves, M. Aramaki, S. Ystad, and R. Kronland-Martinet, "Assessing sound perception through vocal imitations of sounds that evoke movements and materials," in *Computer Music Multidisciplinary Research (CMMR2017)*, Matosinhos, Portugal, Sep. 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01810880

[17] 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.

[18] 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.

[19] 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 – TENOR'15*, M. Battier, J. Bresson, P. Couprie, C. Davy-Rigaux, D. Fober, Y. Geslin, H. Genevois, F. Picard, and A. Tacaille, Eds., Paris, France, 2015, pp. 58–69.

[20] G. Hajdu and N. Didkovsky, “Maxscore: Recent developments,” 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. 138–146.

[21] 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.

[22] C. Hope, “Electronic scores for music: The possibilities of animated notation,” *Computer Music Journal*, vol. 41, no. 3, pp. 21–35, 2017.

[23] L. Vickery, “Some approaches to representing sound with colour and shape,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 165–173.

[24] 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*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 111–118.

[25] J. Bresson, C. Agon, and G. Assayag, “OpenMusic – Visual Programming Environment for Music Composition, Analysis and Research,” in *ACM MultiMedia (MM'11)*, Scottsdale, United States, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01182394

[26] D. G. Nathan Magnus, “Musician assistance and score distribution (masd),” in *Proceedings of The International Conference on New Interfaces for Musical Expression – NIME'2012*. Ann Arbor.: University of Michigan, 2012.

[27] M. L. Lupetti, G. Piumatti, and F. Rossetto, “Phygital play hri in a new gaming scenario,” in *2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*, June 2015, pp. 17–21.

[28] D. Kim-Boyle, “The 3-d score,” 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 Coru na, Spain: Universidade da Coru na, 2017, pp. 33–38.

[29] L. Gabrielli and S. Squartini, *Wireless Networked Music Performance*. Singapore: Springer Singapore, 2016, pp. 53–92. [Online]. Available: https://doi.org/10.1007/978-981-10-0335-6_5

[30] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, “An overview on networked music performance technologies,” *IEEE Access*, vol. 4, pp. 8823–8843, 2016.

[31] G. Hajdu, “Embodiment and disembodiment in networked music performance,” in *Body, Sound and Space in Music and Beyond: Multimodal Explorations*. Taylor & Francis, 2017.

[32] P. f. Sinclair, R. Cahen, J. Tanant, and P. Gena, “New Atlantis: Audio Experimentation in a Shared Online World,” in *Bridging People and Sound. 12th International Symposium, CMMR 2016, São Paulo, Brazil, July 5–8, 2016, Revised Selected Papers*, ser. Lecture Notes in Computer Science, S. Y. Mitsuko Aramaki, Richard Kronland-Martinet, Ed. Springer, 2017, vol. 10525, pp. 229–246. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01791975

[33] K. Fox, “Accretion: Flexible, networked animated music notation for orchestra with the raspberry pi,” in *Proceedings of the First International Conference on Technologies for Music Notation and Representation – TENOR'15*, M. Battier, J. Bresson, P. Couprie, C. Davy-Rigaux, D. Fober, Y. Geslin, H. Genevois, F. Picard, and A. Tacaille, Eds., Paris, France, 2015, pp. 103–108.

DRAWSOCKET: A BROWSER BASED SYSTEM FOR NETWORKED SCORE DISPLAY

Rama Gottfried
Hochschule für Musik und Theater
Hamburg, Germany
rama.gottfried@hfmt-hamburg.de

Georg Hajdu
Hochschule für Musik und Theater
Hamburg, Germany
georg.hajdu@hfmt-hamburg.de

ABSTRACT

We present DRAWSOCKET, a new platform for generating synchronized, browser-based displays across an array of networked devices developed at the Hochschule für Musik und Theater, Hamburg. Conceived as a system for distributed notation display with applications in music and spatial performance contexts, DRAWSOCKET provides a unified interface for controlling diverse media features of web-browsers which can be utilized in many ways. By providing access to browser mouse and multitouch gesture data, and the ability to dynamically create user-defined callback methods, the DRAWSOCKET system aims to provide a flexible tool for creating graphical user interfaces. Included is a discussion of the architecture design and development process, followed by an overview of the features, and syntax considerations for the DRAWSOCKET API.

1. DRAWSOCKET

The DRAWSOCKET design approach is based on the "o.io" paradigm developed at the University of California, Berkeley's Center for New Music and Technology (CNMAT), which uses the OpenSoundControl (OSC) encoding [1] to create a uniform user application programing interface (API) by "wrapping" vendor- and protocol-specific details in an interoperable API syntax [2, 3]. In this way, the DRAWSOCKET system is an "o.io" wrapper for web browser display and interaction, aiming to provide a homogenous OSC API for manipulating the graphic building blocks of Scalable Vector Graphics (SVG), [1] Cascading Style Sheets (CSS), [2] HyperText Markup Language (HTML), [3] and a curated collection of client-based JavaScript libraries for animation and sound production.

1.1 Architecture Overview

The DRAWSOCKET architecture is structured as a server-client system, using Max [4] as the primary controller interface. From Max, control messages are sent to specified client browsers via a Node.js [5] server, which routes the messages by a given client addresses. The messages are then parsed and executed in the client-browser, to generate content or perform other actions.

The choice of Node.js for the server backend was particularly helpful due to the Node Package Manager (NPM), which is bundled with Node.js, allowing DRAWSOCKET to leverage the active community of Javascript library development of tools for browser-based display and inter-computer communication [4]. Further, NPM provides a practical method for managing libraries dependancies, via the *package.json* system.

In an effort to scale to larger groups of clients running off of the same server, a loosely defined *model-view-controller* [5] pattern is used to separate the processes. The server is used primarily to relay and cache the drawing commands, while the drawing implementation is offloaded to the client browsers, which have become quite efficient with recent developments in mobile computing. [6]. See figure 1 for an overview schematic of the system.

1.2 Controller

Max. Currently the primary targeted user server control platform is Max, which provides many algorithmic and interprocess communications tools. Since the release of Max 8, Max now includes the Node For Max (N4M) [6] framework, which embeds the Node.js server engine within the Max programming environment, accessible through a set of Max objects.

The first versions of the DRAWSOCKET system used an independent Node.js application running from the command prompt and a User Datagram Protocol (UDP) [7] socket to send and receive OSC bundles to and from Max, which were then broadcast to subscribed clients. However, after comparing benchmarks measuring the roundtrip messaging time between Max and node.js, the N4M system was found to be faster, and so we adopted this platform as the primary use case. However, the DRAWSOCKET system is well compartmentalized, and so could still easily be reconfigured for use with other control applications.

Within Max, the core Node.js server script, *drawsocket-server.js*, is run within Max's *node.script* object. For convenience, the node.script object is wrapped in Max abstraction called *hfmt.drawsocket* which aids in managing server

[1] https://www.w3.org/TR/SVG11/
[2] https://www.w3.org/Style/CSS/specs.en.html
[3] https://www.w3.org/TR/html52/
[4] https://cycling74.com
[5] https://nodejs.org
[6] https://docs.cycling74.com/nodeformax/api/
[7] https://tools.ietf.org/html/rfc768

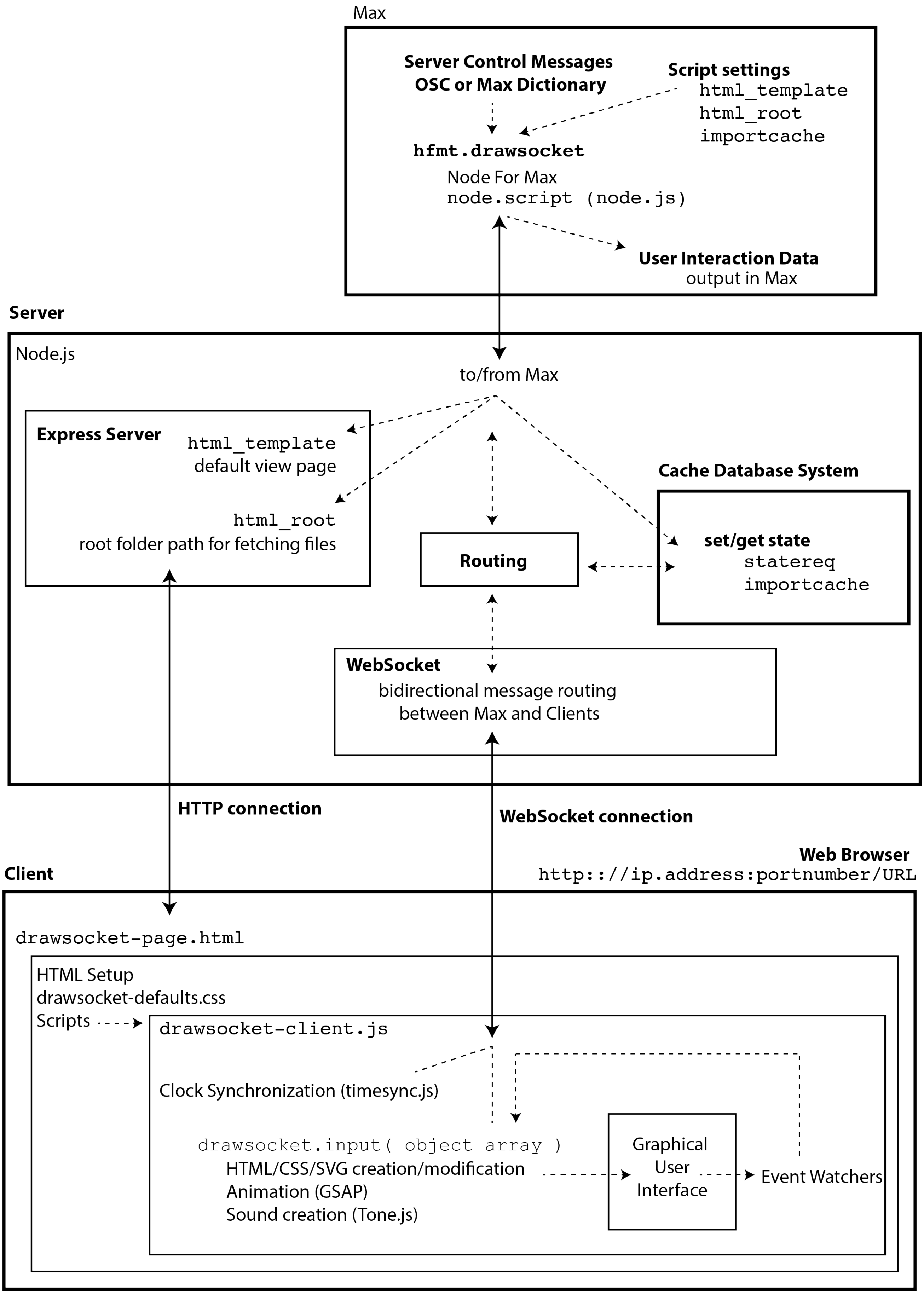


Figure 1. DRAWSOCKET Server/Client Architecture.

asset paths, and handles user interaction messages returning from the client (see section 2 for more details).

OSC-JSON representation. The DRAWSOCKET data is formatted as a key-value tree, which in Max can be represented as either OSC, or in Max Dictionary format, both of which can be easily transformed to JavaScript Object Notation (JSON). [8] See section 3 below for an in-depth description of the DRAWSOCKET messaging syntax.

1.3 Server

Node.js. The Node.js server consists of four main processes: (1) an Express HTTP server, [9] (2) a WebSocket [10] connection manager, (3) state caching, and (4) handling messages from the client (either forwarding them to the Max host environment or responding back to client, as in the case of clock synchronization and on-load initialization).

Express. The Express JS library is used to create the web server and handle HyperText Transfer Protocol (HTTP) [11] requests from client browsers. By default, the server responds to all page requests with a default HTML file which contain the basic setup necessary for most uses of the DRAWSOCKET system, with links to dependency JS libraries, fonts, and a default CSS stylesheet.

If a custom HTML page is desired, users can send the *html_template* message from the Max interface, to set a new default HTML file.

The Express server uses a static public root folder, which exposes a selected folder path that client browsers may load files from. To set the root public folder, users can set the *html_root* folder path as an initialization argument to the *hfmt.drawsocket* abstraction.

User configuration of the system is described further in section 2.

WebSockets. WebSockets are used as the primary server-client communication exchange protocol. The server accepts WebSocket requests from client browsers, and subscribes clients to receive messages addressed to their browser's URL (Uniform Resource Locator).

Control messages composed in Max as an OSC bundle or Max Dictionary are received in JSON format by the server, and routed to the clients identified by their corresponding URL web-address. [12]

URL state caching. Upon receiving control messages addressed to a new URL, the server first forwards the messages via WebSocket to the specified URL address, and then sends a copy of the message to the state cache system.

On connection to a new WebSocket, the client requests the current state of its URL from the server cache. This provides a mechanism for preloading a set of drawing commands to a given URL, so that when a user first loads the page, or hits refresh, the current drawing state of the page will be loaded.

[8] https://www.json.org/
[9] https://expressjs.com/
[10] https://www.npmjs.com/package/ws
[11] https://www.w3.org/Protocols/
[12] https://www.w3.org/Addressing/URL/url-spec.txt

Client return messages. WebSocket connection is bidirectional, and is used to handle messages from the client: responding to clock synchronization requests, initialization requests, and forwarding user interaction information to Max.

Connection Port. On startup, the server provides its IP address and connection port to the Max environment. Clients may connect remotely via network IP address, or if on the same computer, use the localhost identifier, followed by the port number (currently 3002 by default), separated by a colon (e.g. *http://localhost:3002*).

1.4 Client

Running inside a web browser, the client-side component of the DRAWSOCKET system handles the drawing and sound generating commands, clock synchronization, and user interface event watchers.

Web browser. Currently the system targets Safari and Chrome. At the current time of writing, Firefox SVG 2 feature support is behind the above two. Other browsers may also work, but are not currently being tested.

Layout. The central browser display layout consists of one HTML <div> node, which contains an <svg> element, and an SVG <g> group node.

```
<div id="main-div">
   <svg>
        <defs id="defs"></defs>
        <g id="main-svg"></g>
   </svg>
</div>
```

The default formatting for the basic page elements is setup in the *drawsocket-defaults.css*. When users add SVG elements, they are added to the "main-svg" group. Users may also create new SVG groups and add elements to these new groups, to control stacking order, described below in more detail.

Libraries. DRAWSOCKET currently makes use of the following client-side Javascript libraries, all available from NPM:

D3.js, [13] a library for SVG information visualization. [14]

PDF.js, [15] a PDF viewer library developed by Mozilla, providing support for PDF file reading, viewing and page turning.

Tone.js, [16] a web-audio library by Yotam Mann. Currently, DRAWSOCKET uses Tone.js to provide basic soundfile playback functionality, this may be expanded in the future.

Timesync.js, [17] a clock synchronization library, used in DRAWSOCKET to synchronize animations, and provide a mechanism for timed commands.

[13] https://d3js.org/
[14] We are now mainly using d3.js for DOM node creation and manipulation. So, eventually it is likely that we will remove the d3.js dependency to streamline the codebase. However for the time being d3's utility functions are convenient for rapid development, and appear to be performant enough.
[15] https://mozilla.github.io/pdf.js/
[16] https://tonejs.github.io
[17] https://www.npmjs.com/package/timesync

TweenMax and TimelineMax, from the GreenSock Animation Platform (GSAP), [18] a high-performance system for JS and CSS animation.

Client-side Script. The client-side script running in the browser is called *drawsocket-client.js*, which handles the command processing logic of the system and its execution in the browser.

On load, the script first requests a new WebSocket connection to the server using the browser's URL address, sent to the server via the the WebSocket URL identifier (e.g. *ws://localhost:3002/violin*). On successful WebSocket connection, the script begins the clock synchronization process which runs in the background on the client system, requesting new clock readings from the server at regular intervals [19] . Once the initial clock synchronization is completed, the script sends a state initialization request to the server, to which the server responds with a sequence of commands corresponding to the current state of the given client OSC address.

The central command processing is performed in the function *drawsocket*, which parses an array of time-tagged command objects, and executes the corresponding graphic and sound manipulations in the browser. The *drawsocket* function expects one or more objects with a *key*, *val*, and *timetag* key-value pairs:

```
{
    timetag: current time (supplied by server),
    key: command string
    val: command arguments
}
```

Generally, these objects are formatted in the Node.js server from the API command messages received via Max, however they may also be created by user scripts called from event watchers.

2. DRAWSOCKET USER SETUP

The interface for DRAWSOCKET is designed for use in Max, and is distributed as a Max Package, currently hosted on GitHub at the following url:

https://github.com/HfMT-ZM4/drawsocket

To install, users download the repository and place it in Max's Packages folder. Once installed, users can instantiate the DRAWSOCKET system by creating a hfmt.drawsocket object in a Max patch.

The dependency NPM libraries are not distributed with the package, so on first loading hfmt.drawsocket, you need to send the object the *"script npm install"* message, which asks NPM to download all of the dependencies listed in the node project's *packages.json* file. [20]

Setting the public folder. As mentioned above, by default the DRAWSOCKET server responds to HTTP URL page requests with a default HTML page. Custom HTML pages, and/or other types of assets can also be served to the client from a static root public folder.

The public root folder is a method commonly used to control client access to server folders, and can be set in Max by supplying the relative path to the user's Max patch as an argument to the hfmt.drawsocket object. This system allows users to organize their project in a mobile way, easily moved or installed on a new system.

Within the hfmt.drawsocket abstraction there is a helper script called *startscript.js* which retrieves the user patcher's folder path, and passes the path information as an argument to the node.script object on startup. [21] By default the folder containing the user's patch is used as the root public folder, however, users may wish to choose a different folder. For example, by setting the path "public_html", DRAWSOCKET will expect a folder called "public_html" to be located in the same folder as the Max patch running the hfmt.drawsocket abstraction, and if found will use this folder as the public root folder.

In larger projects it is often convenient to sort assets into separate folders for images, sound files, etc. For example if the user wishes to load an image file called "foo.jpg" located in a */public_html/images* subfolder, they would refer to their file at the address */images/foo.jpg*.

3. DRAWSOCKET API

The DRAWSOCKET API has developed organically as features are gradually added to the system, and has been rewritten several times as new use contexts have arisen.

The API was initially designed in keeping with the conventional "message" format used in the Max environment, so that drawing commands could be easily adapted from commands used for other Max drawing objects such as LCD, or jit.gl.sketch.

A Max message is structured as an array, beginning with a selector string, followed by a list of values, which is interpreted by the receiver based on a preexisting schema. However, as more features were added, some complications arose in regard to the sequence ordering, and as a result an alternative object-oriented API was developed, which makes use of a key-value approach that has proven more extendable for the DRAWSOCKET system (discussed below in section 3.2).

Note that in the discussion below, we use the term *message* as a general purpose term for communication via message events, which could be in different formats (OSC messages, Max messages, JSON objects sent as messages, etc.).

URL address routing. All messages sent to the server are addressed to a URL, which is used by the server to route messages to the appropriate clients. Multiple clients may be logged into the same URL, in which case they would all receive the same drawing commands. For example if you had a group of violins all playing the same part you could have them log into the URL:

http://server.ip.address:port/violin1

Then to send messages to the violin 1 section, you would use the URL address */violin1*, in the same way you wold use an OSC address.

[18] https://greensock.com/gsap

[19] Currently, the script is configured to check every 5 seconds, but this may change depending on performance on a larger scale system

[20] The node project is located in the package's /code/node folder, and includes all scripts, and configuration files.

[21] Note that this requires the user patcher to be saved to disk first, so that it has a valid folder location.

To send to all clients, regardless of URL address, DRAWSOCKET provides usage of the OSC /* wildcard address.

Note that in OSC, /* matches one single address level, whereas the DRAWSOCKET server uses the wildcard address to match any *URL*, which may include multiple slashes. To avoid confusion with OSC convention, it is strongly recommended to use single level addresses (i.e. use */violin1*, not */violin/1* in cases of indexed sub-groupings).

Object references. DRAWSOCKET uses the Document Object Model (DOM)[7] *id* attribute as the primary mechanism for referencing individual client objects from the server. On creation, the client-side script logs a reference to new objects with their unique ID, in a set of associative arrays which can be used for fast object lookup by name. Through this method, objects may be referred to by ID, modified, styled, transformed, or removed.

SVG and HTML nodes use the provided unique identifier as the node's *id* attribute, as per the DOM standard, while other objects such as GSAP animation or Tone.js sound objects are not in the DOM, but logged in the DRAWSOCKET's internal object model.

3.1 List-oriented API

The original "Max style" list-oriented DRAWSOCKET API design used a bundle of individual OSC messages, each of which performed an action on the client system. The list-oriented API has now been replaced, however a discussion of this approach is valuable, since it illustrates a structural limitation that we encountered with this syntax approach.

In the list-oriented API, the OSC message address was used as a way to specify several functional layers at once, through concatenating together multiple values separated by slashes. The general address syntax was made of three main levels: (1) the client URL address, (2) a unique object ID, followed by (3) a command string specifying the process to execute on the client system.

The commands provided a streamlined way to create and modify elements on the client browser, using a curated set of parameters.

For example the following OSC bundle:

```
{
    /violin/foo/draw/rect : [100, 100, 25, 25],
    /violin/foo/style/stroke-width : 1
}
```

contains two messages prefixed by */violin* which indicates that the server should send these commands to all clients logged into the */violin* URL.

Once received on the client system, the script would parse the OSC address, separating the ID from the command string. Here, the ID is "foo" and the command string is "draw/rect".

For each OSC message, the value attached to address was parsed by the client script based on schema defined in the documentation. In the case of "draw/rect", the message's value would be interpreted as defining an SVG rectangle's *x*, *y*, *width*, and *height* values. The second message works in a similar way, except that rather than creating a new object, it adds an inline SVG/CSS *style* attribute to the object node, setting the *stroke-width* parameter to a value of 1.

Grouping. Things start to get a little more complex with the list-oriented approach when attempting to define SVG group objects and object definitions. In these cases child objects can be grouped together and manipulated as a single graphic object, while not requiring each child object to have a unique ID.

The first problem we encountered was when trying to include two objects of the same type within the object. In the first implementation, a sub-bundle of OSC messages was used to group elements together, however since no IDs were required, the following example fails:

```
/*/groupex/draw/group : {
    /text : [210, 210, "hi"],
    /text : [310, 210, "bye"]
}
```

It fails because in OSC you are allowed to have multiple messages with the same address, however in Max the OSC messages need to be first converted to Max Dictionary format to be passed to the node.script object, and Max Dictionaries do not allow duplicate addresses.

Another complication arose when trying to style individual objects within a group, since there is no unique identifier to reference for adding inline style tags, and this is not accessible from the list-oriented API syntax. To address these two issues, the list-oriented grouping syntax was adapted to use an array of objects (aka sub-bundles in OSC). For example:

```
/*/groupex/draw/group : [{
    /path : "M200,200a30,90,0,0,0,0-60a30,30,0,0,0,0,60",
    /style : "fill: black"
}, {
    /text : [210, 210, "hi"],
    /style : "fill: red"
}]
```

In this case, the *style* message is bound to the *path* message by wrapping them together in an object. This solution led to a reevaluation of the DRAWSOCKET API syntax, and resulted in the development of the object-oriented based API.

3.2 Object-oriented API

While the list-oriented approach provides a compact, one-line syntax, the list format is also limited, in that the list requires a predefined schema for how the list can be interpreted, and which types of operations the list values may address. The list-based approach is thus less easily extendable, since adding a new value to the list requires adding a new step in the interpreting script. The main benefit of the list syntax is that its compactness makes it sometimes faster for rapid prototyping, however the object-oriented approach can be more easily expanded as we will show below, and additionally, the object-oriented approach is helpful since it is self-describing, emphasizing legibility by associating a parameter name with each value.

For example, whereas above we drew a rectangle with a list, such as:

```
/violin/foo/draw/rect : [100, 100, 25, 25]
```

The same rectangle could be drawn with the object-based API using the "svg" *key*, and an *val* containing one or more objects to process. The *"new"* keyword notifies the client that it should create a new SVG element:

```
/violin : {
  /key : "svg",
  /val : {
      /new : "rect",
      /id : "foo",
      /x : 100,
      /y : 100,
      /width : 25,
      /height: 25
    }
}
```

In the object-based approach each variable now has a name associated with its value, telling us what the variable represents. The list-approach is a more concise, requiring less typing, however, when we consider further what the messages are representing in the context of the DRAWSOCKET system, the benefit of the object approach becomes clearer.

SVG is based on the Extensible Markup Language (XML) format,[22] and is designed as tree of *nodes*, each with a set of *attributes* which are defined as key-value pairs. By using the same attribute names within the DRAWSOCKET object API, the client script can then simply insert as few or many of the attributes as it receives, rather than needing a specific set of attributes, as with the list-based approach. Also, by staying close to the original SVG API, the user can refer to the SVG specification directly to figure out which attributes they can use, rather than needing to limit their control parameters to those setup in the list parsing schema.[23]

For example, extending the above example, here we create two new objects, a rectangle and a circle, by defining them in an array, and additionally assign a CSS *class* reference for each:

```
/violin : {
  /key : "svg",
  /val : [{
    /new : "rect",
    /id : "foo",
    /x : 100,
    /y : 100,
    /width : 25,
    /height: 25,
    /class : "room"
  }, {
    /new : "circle",
    /id : "bar",
    /cx : 112,
    /cy : 112,
    /r : 5,
    /class : "source'
  }]
}
```

Keywords. There are currently four reserved keywords used with *svg* objects: *new*, *style*, *parent*, and *child*.

On receiving an *svg* object (or array of objects), the client-side script iterates each element of the array, and checks if there is an already existing object with that *id* tag; if so, it selects that element from the DOM lookup table. Next, the script checks if there is a *new* message in the object; if so, it creates a new node, either replacing the element at the existing *id*, or creating a new node if not already existing, and then processes the rest of the object messages.

If the object already exists, and no *new* is found, DRAWSOCKET will use the values in the object to update the object attributes. For example:

```
/violin : {
    /key : "svg",
    /val : {
      /id : "foo",
      /width : 100
    }
}
```

will change the *width* attribute of the node "foo" without modifying any other attributes that may have already been set.

3.3 Parent and child elements

Appending child nodes to parent SVG element can be accomplished via the *parent* and *child* keywords.

The *child* keyword, is a high-level API helper function that assists the user in specifying one or more child nodes in a tree syntax. The value attached to this address will be inserted as a child of the parent node, for example the inner text of a <text>element, or a new node within a <g>the SVG grouping element tag.

Here is an example of a circle and line contained in new SVG group, called "noteline":

```
/violin : {
    /key : "svg",
    /val : {
        /new : "g",
        /id : "noteline",
        /x : 100,
        /y : 100,
        /child : [{
            /new : "line",
            /id : "liney",
            /x1 : 10,
            /y1 : 5,
            /x2 : 100,
            /y2 : 5,
            /style : {
                /stroke-width : 1
            }
        }, {
            /new : "circle",
            /id : "circley",
            /cx : 5,
            /cy : 5,
            /r : 5,
            /style : {
                /stroke-width : 2,
                /fill : "none",
                /stroke : "black"
            }
        }]
    }
}
```

Nodes with a *parent* attribute are inserted as children of the node with the *id* specified by the *parent*, as long as the parent element is already existing in the DOM. If no *parent* element is specified, the node is inserted into the default SVG group "main-svg".

[22] https://www.w3.org/TR/xml/

[23] That said, we have not yet fully tested the entire SVG specification. We believe the object API provides access to everything, but there maybe some unaddressed aspects.

For example, the above tree syntax could also be written this way:

```
/violin : {
    /key : "svg",
    /val : [{
        /new : "g",
        /id : "noteline",
        /x : 100,
        /y : 100
    }, {
        /new : "line",
        /id : "liney",
        /parent : "noteline",
        /x1 : 10,
        /y1 : 5,
        /x2 : 100,
        /y2 : 5,
        /style : {
            /stroke-width : 1
        }
    }, {
        /new : "circle",
        /id : "circley",
        /parent : "noteline",
        /cx : 5,
        /cy : 5,
        /r : 5,
        /style : {
            /stroke-width : 2,
            /fill : "none",
            /stroke : "black"
        }
    }]
}
```

3.4 SVG layer drawing contexts

In an SVG file, each object element is drawn in the same order as they are written in the file, from top to bottom, with the last element being drawn last, "on top" of any objects that may have been drawn in the same location. In the DRAWSOCKET system, the drawing sequence is set through the order of the object creation (using the *new* keyword).

Using the *parent* and *child* keywords, new nodes can be created and inserted as children of existing nodes. The order in which the child nodes are created, sets the drawing order of the nodes. Importantly, *editing* nodes (i.e. setting values without the use of the *new* keyword), does *not* change the drawing order. Similarly, inserting nodes does not change the drawing order of the parent nodes. This rule makes it possible to use SVG groups as drawing layer contexts, which maintain stacking order relative to each other.

As an illustration, let's say you would like to have three layers, a background, middle and overlay. You could create three new groups within the main SVG node, called "back", "main", and "overlay", in a specific drawing order, like this:

```
/violin : {
    /key : "svg",
    /val : [{
        /new : "g",
        /id : "back"
    }, {
        /new : "g",
        /id : "main"
    } , {
        /new : "g",
        /id : "overlay"
    }]
}
```

You could then use the *parent* keyword to append nodes to the newly created groups. New nodes are draw above older nodes, but since the groups maintain their drawing order, you can use them as layers. In this example, the "overlay" layer, will always be drawn after the "back" and "main" layer-groups.

```
/* : {
        /key : "svg",
        /val : [{
            /parent : "main",
            /id : "clef",
            /new : "text",
            /child : "&#xE050",
            /class : "bravura_text",
            /x : 40,
            /y : 50
        }, {
            /parent : "back",
            /new : "rect",
            /id : "rect",
            /x : 5,
            /y : 5,
            /width : 100,
            /height : 100,
            /fill : "red"
        }, {
            /parent : "overlay",
            /new : "circle",
            /id : "circle",
            /cx : 50,
            /cy : 50,
            /r : 10,
            /fill : "blue"
        }]
}
```

Using this approach, multiple layers of SVG elements can be grouped together and manipulated (with some limitations as described in the SVG specification).

3.5 SVG CSS Styling.

The ability to dynamically apply CSS styling operations on SVG elements provides the user with an extremely flexible mechanism for composing, and manipulating the graphic layout. For most common DRAWSOCKET usages, a set of default layout properties are defined in the file *drawsocket-default.css*, which is loaded with the default HTML file (*drawsocket-page.html*). The linked stylesheet sets some defaults for SVG element types, for example *lines* have a default stroke width value so that they are visible by default. [24]

DRAWSOCKET also provides dynamic access to CSS rules, for which it is useful to understand the hierarchy of SVG style properties.

There are three levels of inheritance with SVG CSS styling:

(1) *presentation attributes*, set within the element, e.g.: *<rect fill="red" >*;

(2) *stylesheet* definitions, loaded via an attached CSS stylesheet document, or within a <style> element in the HTML document; and

(3) *inline styling*, a snippet of CSS wrapped in a string and set in an elements's *style* attribute, e.g.: *<rect style="fill: red; stroke: 2" >*.

[24] Note that this is not always desirable, for example when importing SVG files exported by a program like Adobe Illustrator, which assumes that there are no pre-exiting CSS rules in place. For these cases, DRAWSOCKET users can either override the defaults via a new CSS definition, or change the .css file by hand.

Each is overridden by the next: stylesheets override presentation attributes, and inline styles override all the others. [25]

Using CSS *class* selector syntax opens up many possibilities. For example, here is an example using the object-array syntax to set create two CSS classes: (1) ".notehead" which sets defaults for fill and stroke properties, as well as the radius value, *r* ; and (2) ".notehead.open", a sub-class of ".notehead" which overrides the fill property.

Following the *css* definitions, a new SVG circle object is created and configured with the "notehead open" class.

```
/violin : [{
   /key : "css",
   /val : [{
     /selector : ".notehead",
     /props : {
         /stroke : "black",
         /stroke−width : 2,
         /fill : "black",
         /r : 5
     }
   }, {
     /selector : ".notehead.open",
     /props : {
         /fill : "none"
     }
   }],
 }, {
   /key : "svg",
   /val : {
      /new : "circle",
     /id : "foo",
     /class : "notehead open",
     /cx : 20,
     /cy : 20
   }
 }]
```

3.6 SVG import and library definitions

DRAWSOCKET provides access to several methods for importing and reusing fragments of SVG. This is a useful approach for reducing the amount of data that needs to be sent over the network, and can greatly simplify the construction of more complex notation situations.

Referencing SVG definitions. There are two node types in the SVG specification which allow the user to create prototypes of graphic elements, *defs* and *symbol*, which can be applied like a stamp via the *use* tag.

Within the DRAWSOCKET main SVG element there is an element group called *defs* which is not directly drawn to the screen, but is visible by using the browser's HTML element viewer tool. DRAWSOCKET uses the same drawing context syntax for the *defs* node, as it does for the other drawing layers.

For example, the following snippet makes a new SVG group object in the *defs*, called "noteline", which contains a line and a circle:

```
/violin : {
   /key : "svg",
   /val : {
     /parent : "defs",
     /new : "g",
     /id : "noteline",
     /child : [{
         /new : "line",
         /x1 : 10,
         /y1 : 10,
         /x2 : 100,
         /y2 : 10
      }, {
       /new : "circle',
       /cx : 5,
       /cy : 5,
       /r : 5
     }]
   }
 }
```

Typically the user would send a library of definitions at the beginning of the piece, and then refer to the set of definitions as needed via the *use* SVG element and its *href* attribute, creating an internal reference to a given definition selected through its *id* attribute. [26]

For example, the following new SVG object "foo", references the "noteline" definition above, offset to x, y position $\{100, 100\}$:

```
/violin : {
   /key : "svg",
   /val : {
     /new : "use",
     /id : "foo",
     /href : "#noteline",
     /x : 100,
     /y : 100
   }
 }
```

Importing fragments. The *use-href* syntax approach can also be used to import elements from external SVG files stored in the public HTML folder, by adding the target object's *id* to the external file path. For example, to reference an object with the ID "boo" in an external file called "other.svg" that is located in the public subfolder called "media" you could use the following snippet:

```
/violin : {
   /key : "svg",
   /val : {
     /new : "use",
     /id : "foo",
     /href : "/media/other.svg#boo"
   }
}
```

If *x* or *y* attributes are set in the *use* node, the referenced object will be offset by the amount specified by the *use* attributes.

DRAWSOCKET also provides an additional option with the *href* attribute. If the *href* value is a list, the second value is non-zero, the script will find the original object's bounding box and offset so that it lies at the origin $\{0, 0\}$, and then applies the x, y values as a second operation. The benefit of this feature is that it allows you to coordinate positions of objects without needing to know their original position in the reference file.

[25] With one exception, stylesheet definitions with the *!important* tag will override inline styles.

[26] Note that for all selections we are using the HTML/CSS # sign to specify that the following string is an *id*.

3.7 PDF import

PDF files may be imported into DRAWSOCKET. For example, to load a PDF, storing it at the DRAWSOCKET ID "foo", setting its *x* position, *width* and setting it to display page 2:

```
/∗ : {
   /key : "pdf",
   /val : {
     /id : "newpdf",
     /href : "/media/flint_piccolo_excerpt.pdf",
     /width : 600,
     /x : 100,
     /page : 2
   }
}
```

3.8 Animation

While DRAWSOCKET objects may be animated using native CSS transitions and keyframes, the GSAP TweenMax and TimlineMax libraries were introduced to provide a much more convenient and cross-browser supported method. With the TweenMax library users can create a "tween" transition between the object's current position and current CSS property values, to another set of values over a given amount of time, using the TweenMax.to function via the *tween* DRAWSOCKET key. For example:

```
/violin : {
   /key : "tween",
   /val : {
     /id : "aaa",
     /target : "#note",
     /dur : 10,
     /vars : {
       /x : 100,
       /y : 100,
       /opacity : 0
     }
   }
}
```

moves the SVG object "note" to the xy position $\{100, 100\}$, and fades the opacity to zero over a course of 10 seconds. The *tween* is stored as object in the DRAWSOCKET script at the given ID (here "aaa"), and may be recalled at will (see the online documentation for more details). The CSS selector *target*, *dur* and *vars* are plugged directly into the argument fields for the TweenMax.to function. [27]

More complex animations can be implemented with the TimelineMax function, via the DRAWSOCKET *timeline* command, which is comprised of an array of tweens (which can also have different targets). As with the TweenMax.to function, an effort was made to make the encoding syntax as close to the native GSAP Timeline function as possible so users can refer to the GSAP documentation for full reference.

```
/violin : {
   /key : "timeline",
   /val : {
     /id : "foo_line",
     /init : {
         /paused : "true",
         /yoyo : "true",
         /repeat : 20
     },
     /tweens : [{
         /target : "#bar",
         /dur : 1,
         /vars : {
             /y : 270,
             /x : 100,
             /scaleX : "200%",
             /opacity : 1,
             /ease : "linear"
         }
     }, {
         /target : "#bar",
         /dur : 2,
         /vars : {
              /y : 10,
              /x : 0,
              /scale : "100%",
              /opacity : 1,
              /ease : "linear"
         }
     }]
   }
}
```

DRAWSOCKET provides the *cmd* keyword for tweens (and timelines of tweens) to start, stop, reset, reverse, etc.

Synchronization. All commands sent from the server are timestamped, which provides DRAWSOCKET with a mechanism to synchronize animations. Using the Timesync.js library, the client periodically asks the server for its current clock time and logs an offset value between the two clocks. [8] Then, whenever a new animation start request is received, the client checks the message's timestamp relative to the current client clock time minus the logged difference from the server time to get the corrected animation start time in terms of the server clock. The client then checks the duration of the animation (tween or timeline) to make sure it hasn't already missed the end time for the animation, if not, the client script starts the animation, fast-forwarding if necessary to compensate for network lag.

3.9 Sound

In addition to providing access to browser-based drawing tools, DRAWSOCKET also makes use of the Tone.js [9] WebAudio [28] Framework for browser-based sound production.

The Tone support library also adds a new keyword, *call* which expects an object containing a *method* and optional *args*. Additionally, a the *call* object may also contain a *then* object which can be used as a sequential *call*, applied to the return value form the parent method call.

For example, he we create a new Tone.Player, load an mp3 file, tell it to start looping playback, and call the *toMaster()* Tone.Player method:

[27] https://greensock.com/docs/TweenMax/TweenMax()

[28] https://www.w3.org/TR/webaudio/

```
/* : {
  /key : "sound",
  /val : {
    /new : "Player",
    /id : "kick",
    /vars : {
        /url : "/media/808_mp3/kick1.mp3",
        /autostart : "true",
        /loop : "true"
    },
    /call : {
      /method : "toMaster"
    }
  }
}
```

3.10 HTML5

DRAWSOCKET provides access to HTML nodes via the *html* tag.

For example, this loads a video:

```
/* : {
  /key : "html",
  /val : {
    /new : "video",
    /id : "foo",
    /child : {
        /new : "source",
        /type : "video/mp4",
        /src : "somerandommovie.mp4"
    }
  }
}
```

Some HTML5 JS objects also support the *call* keyword.

For example, this starts playing the above video:

```
/* : {
  /key : "html",
  /val : {
    /id : "foo",
    /call : {
        /method : "play"
    }
  }
}
```

3.11 User Interaction

Lastly, DRAWSOCKET also sends user interaction information back to the server, outputting the data into Max where it can be used to control other processes, through mouse and multi-touch event listeners, and via HTML textfield input forms. When the user's mouse or fingers move over the screen DRAWSOCKET reports the x, y position and the top-most graphic object under the fingers or cursor, and bound with the URL address. [29]

DRAWSOCKET also provides access to HTML text input fields. To create a text field, users first create a form with a default text prompt and then position the form by applying a CSS transform, or tween. When a client enters text into the text input field and hits enter or clicks outside of the form, the text is sent back to the server and output in Max in a similar fashion to mouse and multi-touch data.

User defined event callbacks. The main client-side processing function can also be invoked from a callback for handling user interaction, exposed to the global JS namespace as *drawsocket.input*. For example, the following snippet, which creates an SVG path object, and assigns an *onclick* callback function which triggers a sample playback when the client user clicks on the path object:

```
/* : {
  /key : "svg",
  /val : {
    /new : "path",
    /id : "wow",
    /style : {
        /fill : "red"
    },
    /d : "M100,100a30,30,0,0,0,0−60a30,30,0,0,0,0,60",
    /onclick : "drawsocket.input({
        key: 'sound',
        val: {
            id: 'kick',
            call: {
                method: 'restart'
            }
        }
    })"
  }
}
```

[29] Currently there is no unique client identifier, i.e. all users on the same URL will send their user interaction data identified by the same URL address. A unique tagging system will likely be implemented at some point.

4. FUTURE WORK

DRAWSOCKET currently still considered "in development", that said, the system has already been used in several live performances, and appears to be fairly robust. As we prepare for the large-scale extension of the system to the St. Pauli Elbtunnel [10] we will have a good opportunity to fully stress-test the system.

One potential issue that we imagine could arise is with the caching system. Currently the caching routine is processed within the callback function that gets called when a new dictionary arrives from Max. On receiving a new dictionary, the server routes the data, sending packets to the appropriate clients, and then sends the packets to the cache system which unions the data with any nodes with a matching *id* (or *selector* in the case of CSS). There is a question about the scalability of this approach.

Node.js, like vanilla JS, uses an "event driven", "single threaded event loop model", which uses a queue of event callbacks which need to be processed asynchronously. However, it is possible to block the event loop[11] within a callback function, should the execution take too long. In particular, the JSON.parse and JSON.stringify operations are potentially expensive, with a complexity of $O(n)$; so, depending on the size of the incoming dictionary, this could significantly slow down the response of the server. In our testing so far, we have already noticed some issues with processing very large dictionaries arriving from Max, but we need to investigate further. It is possible that since the data is broadcasted before being sent to the caching system, that the blocking of the event loop will be less noticeable on the client-side, however, the responsiveness of the server will be reduced and this will likely effect the clock-synchronization routine, and could also in extreme cases result in a buildup of events to process in queue. To address this issue, we might look into storing the URL states in a separate database, which runs as a separate process.

Acknowledgments

The authors would like to thank Jacob Sello for his detailed testing the system which pushed the development of many new features and design considerations. We would also like to acknowledge the Federal Ministry of Education and Research in Germany (BMBF), for their support of this research through the Innovative Hochschule initiative.

5. REFERENCES

[1] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193–200, 2005.

[2] A. Freed, D. DeFilippo, R. Gottfried, J. MacCallum, J. Lubow, D. Razo, and D. Wessel, "o.io: a Unified Communications Framework for Music, Intermedia and Cloud Interaction." in *ICMC*, 2014.

[3] J. MacCallum, R. Gottfried, I. Rostovtsev, J. Bresson, and A. Freed, "Dynamic Message-Oriented Middleware with Open Sound Control and Odot," in *Proceedings of the International Computer Music Conference (ICMC'15)*, Denton, TX, USA, 2015.

[4] A. Trockman, S. Zhou, C. Kästner, and B. Vasilescu, "Adding sparkle to social coding: an empirical study of repository badges in the NPM ecosystem," in *Proceedings of the 40th International Conference on Software Engineering*. ACM, 2018, pp. 511–522.

[5] G. E. Krasner, S. T. Pope *et al.*, "A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System," *Journal of object oriented programming*, vol. 1, no. 3, pp. 26–49, 1988.

[6] M. Halpern, Y. Zhu, and V. J. Reddi, "Mobile CPU's rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction," in *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 64–76.

[7] M. Champion, L. Wood, G. Nicol, S. B. Byrne, A. L. Hors, P. L. Hégaret, and J. Robie, "Document object model (DOM) level 3 core specification," W3C, W3C Recommendation, Apr. 2004, http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/.

[8] "timesync.js," accessed 2019-1-9. [Online]. Available: https://www.npmjs.com/package/timesync

[9] Y. Mann, "Interactive music with tone.js," in *Proceedings of the 1st annual Web Audio Conference*. Citeseer, 2015.

[10] R. Gottfried and G. Hajdu, "Massive networked music performance in the old elbe tunnel," in *2019. Proceedings of the International Conference on Technologies for Music Notation and Representation-TENOR2019. Melbourne*, 2019.

[11] node.js. (2019) Don't Block the Event Loop (or the Worker Pool). [Online]. Available: https://nodejs.org/en/docs/guides/dont-block-the-event-loop/

LIVE STRUCTURES: OCULAR SCORES™

Linda Bouchard
Independent Composer
Berkeley, California, United States
linda@lindabouchard.com

Joseph Browne
Visuals, Software Design and Programming
matralab — Concordia University
Montreal, Quebec, Canada
joseph.a.browne@gmail.com

ABSTRACT

Live Structures is a research and composition project that explores different ways to interpret data into graphic notation and compositions. The Live Structures project started in October 2017, supported by an Explore and Create Grant from the Canada Council for the Arts received by Bouchard. One of the goals of the Live Structures project is to interpret data from the analysis of complex sounds into a visual musical notation. The tool, developed in collaboration with Joseph Browne of matralab at Concordia University, is called Ocular Scores™ . So far, three iterations of the Ocular Scores Tool have been created, each performing multiple functions: a) the ability to draw an image from the analysis of complex sounds that can be used as gestural elements to compose new works or to compare a complex sound against another complex sound, b) the ability to draw full transcriptions of a performance for future interpretation, and c) the ability to draw images in real time and to manipulate those images to create interactive projected scores to be performed live by multiple performers.

These various applications and how they can inspire composers and performers will be described in more detail in this paper.

1. INTRODUCTION

Music notation is by definition a visual notation: images and codes that convey a wide variation of precision and systems to express pitch over time. Musical notation that utilizes mostly images, often described as "graphic notation," has evolved since the 1950s, with more and more composers experimenting to find new ways to express their unique musical vision.

But when we look at the 14th century Baude Cordier "eye music" scores [1] (see Figures 1 and 2) or the 19th century Tibetan Yang-Yig notation [2] (see Figure 3), it is striking to see similarities with some scores that have been created in the last 60 years.

Figures 1 and 2. 1380–1440 Baude Cordier creates two early examples of what has been called "eye music." From Timelines of language, communication and semiotic systems.

There seems to be a certain "Preset of the Mind" [3] in the coding language that musicians are attracted intuitively to use, certain elements in the semantic of the graphical language that seem to repeat themselves through time.

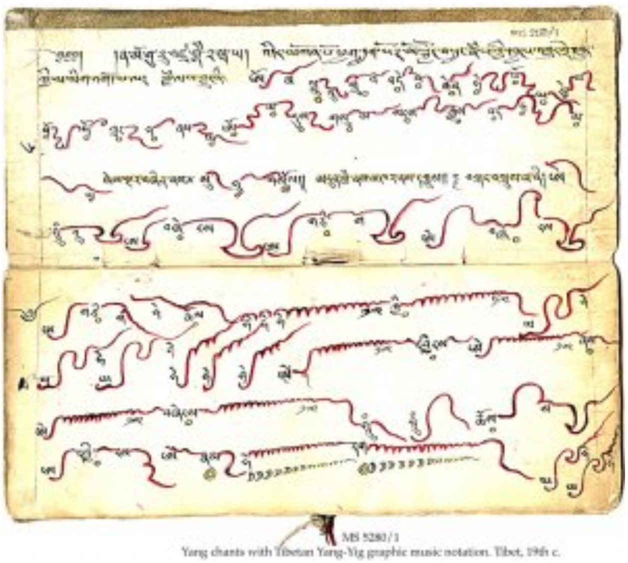


Figure 3. 19th century Tibetan Yang-Yig notation from the Schøyen Collection

With Ocular Scores™ , Bouchard and Browne have chosen to interpret data from the analysis of complex sounds using a custom-made tool that leaves a tremendous freedom of interpretation to the composer and the musicians. Despite being an interpretation of data, the images feel familiar. Bouchard and Browne strived to keep this sense of immediacy while creating images that are replicable.

2. CONTEXT

For forty years, Bouchard's work has taken its inspiration from the world we live in: natural, sociological, and political. It is often inspired by nature's geometry and textures, as if composing music could begin by staring with a magnifying glass at nature's elements — water-gas-rock formations and chemical reactions — and creating from these images a series of abstract musical landscapes.

We have many tools available to analyze natural phenomena, complex sounds, and human behaviors. In the Live Structures project, Bouchard went one step further to include data interpretation to conceive of a new way to compose music: In the case of Ocular Scores™, we are translating data from the analysis of rich and complex sounds into graphic notation and symbolic systems.

3. IMPETUS

During the past few years, Bouchard has collaborated with advanced improvisers while she created multimedia performances. It was inspiring to work with such innovative musicians, and it made Bouchard want to be able to notate their unique collaborations in such a way that they could be interpreted by other musicians. Initially, Bouchard simply wanted to find a way to draw images of complex sounds and extended techniques and integrate those images into traditional scores. From this modest beginning, Bouchard and Browne have developed all of these other applications.

4. DESIGN GUIDELINES

From the start, we looked for a notation that conveyed a sense of immediacy — a visual representation that was familiar and intuitive to musicians adhering to the "preset of the mind" concept. The images had to be replicable: Created from the interpretation of the sound analysis, the images had to be a reliable interpretation of a complex sound. This graphic notation needed to contribute something unique to our already sophisticated and precise musical notation system.

Because of our limited budget, it was impossible to create a digital tool from scratch. In this first exploratory phase, it proved to be unnecessary: There are plenty of tools available that could be applied to achieve our goals. For designing the visual aspect of the tool, Browne chose to use TouchDesigner [4], a node-based visual programming language for real-time graphics and interactive media.

Our initial design process was one of elimination so that we could arrive at something elegant and practical and a tool that we would actually want to use because it would inspire us as creators and would inspire performers as well.

In Ocular Scores™, black lines are drawn on a white surface, as if a pen is drawing an image. See Figure 4.

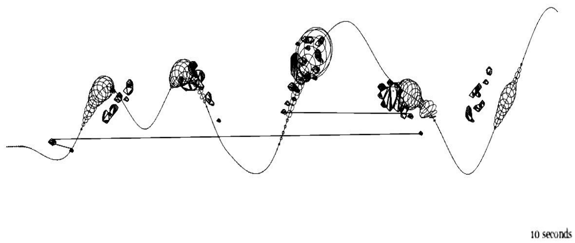


Figure 4. This image was created with Version 1 of Ocular Scores™ and represents five short bursts of wind sounds by François Houle, recorded at the Banff Center for the Arts. [5]

We have the option to use a white line drawn on a black surface when we want to project in a theater space. We also kept the traditional conventions of time on the x-axis and frequency on the y-axis.

We have the option to plot main pitches on music staves (treble and bass) with the option to add additional lines (orange). See Figure 5.

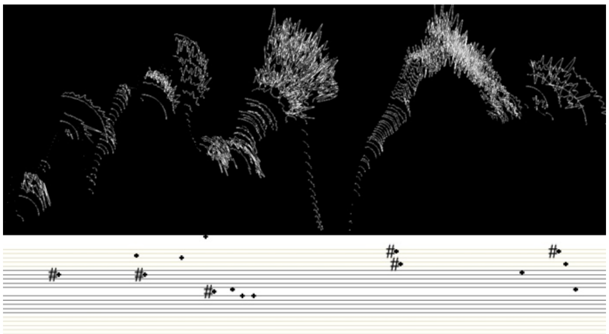


Figure 5. This image was created with Version 2 of Ocular Scores ™ and represents a musical gesture from an improvisation by Lori Freedman, performed and recorded at matralab, Concordia University, Montreal, Quebec, Canada. [6]

Ultimately, we were very careful not to clutter the page; whenever we add a layer of information, we always have the option to take it off.

5. PROCESS

Ocular Scores™ was developed in collaboration with master improvisers who tested each prototype and gave us their expert feedback. Ocular Scores™ is still in development.

6. OCULAR SCORES™

6.1 Version 1: Capturing Gestures

In Version 1, the first step is to upload a prerecorded sound file to be analyzed. See Figure 6.

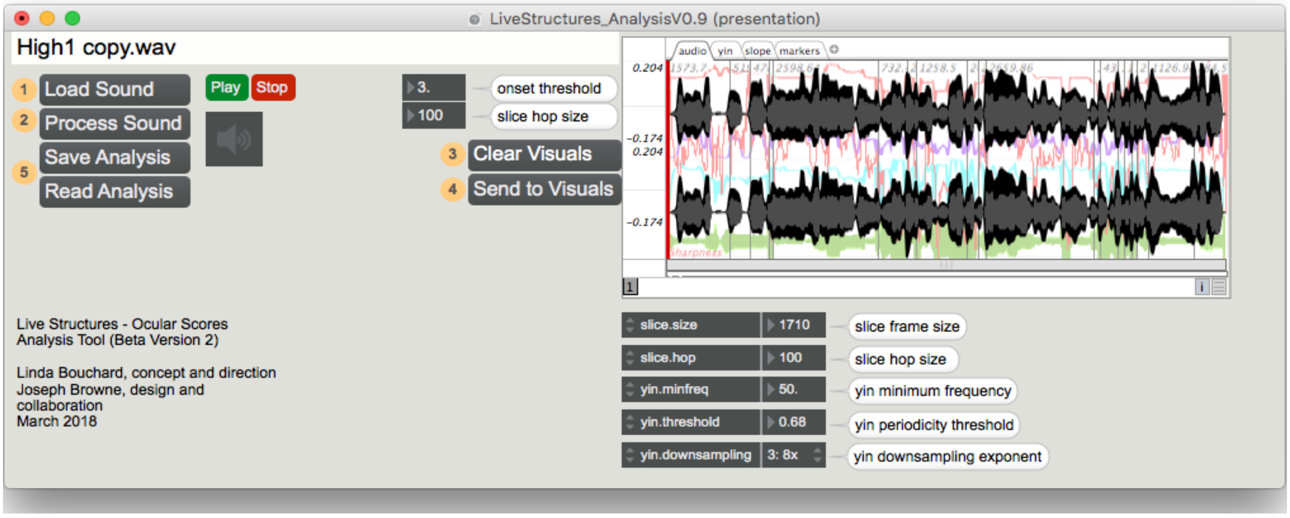


Figure 6. A short sound file is loaded into Max.

The analytical data is produced using the Max package MuBu For Max[1] by IRCAM [7] and their *pipo.yin* [8] audio descriptor to extract frequency, amplitude, and periodicity information from the analyzed sound file. See Figure 7.

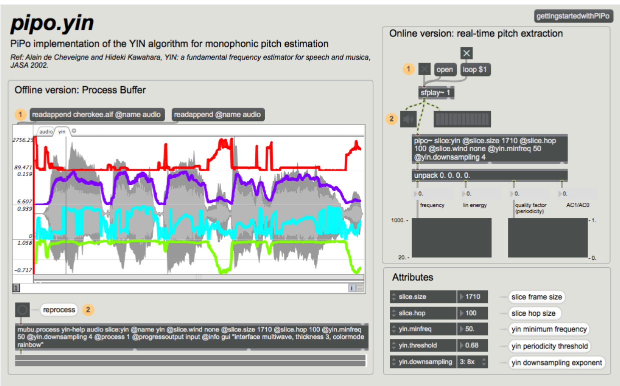


Figure 7. The analytical tool used for Ocular Scores™ Version 1 is *pipo.yin* audio descriptor from MuBu for Max package by IRCAM.

The data is then sent to TouchDesigner over a local UDP network and is translated into a visual representation of the sound. Time is plotted on the x-axis from left to right, frequency data is mapped to translation on the y-axis, and amplitude is represented by the size of the shape being drawn. Periodicity is mapped to rotation of the shape on the z-axis, which created a visual irregularity that defines when the sound file is richer and noisier. The visual has a more uniform pattern when the sound is more periodic.

A variety of scaling parameters and image-based parameters can be adjusted to create an image that the composer feels offers the best impression of that sound. We aimed to limit the available possibilities in order to maintain a cohesive vocabulary for representing sounds visually, but enough options remain that one can truly interpret how the sound is visualized. See Figure 8.

Figure 8. This image shows the user interface from Ocular Scores™ Version 1. On the left are parameters that affect the scaling of the input data — how the image will appear spatially. On the right are parameters that affect the character of the image and repositioning after scaling. With a single data set, one can produce several distinct images.

The result is a mixture of data interpretation and intuitive decision-making.[2] By saving presets, we can

[1] MuBu For Max is a tool set for analysis of sound and motion and interactive sound synthesis and machine learning created by IRCAM in Paris (http:// forumnet.ircam.fr/product/mubu-en/).

[2] Every step of the way we asked ourselves, "What does this sound look like to me? If I were to try to imitate this sound, what image would inspire me in that direction?"

replicate the same image with the same sound over and over. It is an authentic data interpretation.

The advantage of this prototype is that we have many different ways of interpreting a single musical gesture by using different presets. It is easy to create a musical score with a collection of images. [9]

It is possible to add handwritten annotations and fingerings and load them into other performance software, like ForScore[3], or simply print the pages on paper. It is also easy to use this prototype as a pedagogical tool to practice new complex sounds or to imitate complex sounds from other instruments to expand one's vocabulary.[4]

The first full composition using this technology is called "DROP." The musician performed from a series of images that were created from the analysis of different water sounds: droplets, waves, breaking ice, etc. The images [10] were loaded into ForScore and were further annotated by the performer.

As we worked on the piece, it was revealing to see that the visual score required the same kind of commitment that a regular score requires. The same way a musician learns a score with a traditional notation, Kyle Bruckmann, for whom the work was composed, devoted time to understanding the piece, interpreting a code, and committing to specific musical gestures. [11]

A few months later, "DROP" was performed by Canadian soprano saxophone improviser Kyle Brenders. This second interpreter never heard the performance of Kyle Bruckmann, and yet, the composer was happy to hear the similarity of gestures in both performances. The character of the work was definitely recognizable. [12]

It is important to realize that by utilizing a graphic score of this nature, the composer relinquishes aesthetic control to the performer.[5]

6.2: Version 2: Live Input

In this prototype, both the analytical data and the images are produced inside TouchDesigner, bypassing Max. We can upload a sound file in the same way we did in Version 1, or we can analyze the audio input from a live performer. The images are created in real time on a continuous scrolling image. See Figure 9.

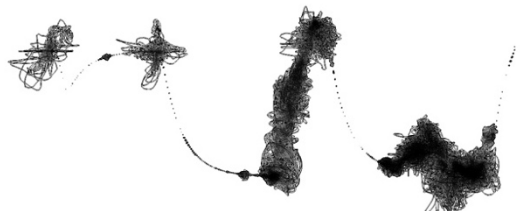


Figure 9. This image was created using the stereoscope shape option while analyzing a sound file of Lori Freedman improvising. [13]

Frequency analysis is achieved by using the analyze node in TouchDesigner to find the First Peak or the Highest Peak of the audio spectrum. The option to use First Peak or Highest Peak gives a more or less accurate analysis depending on the musical context — if it is a more noise-based sound, then highest peak is more accurate; if it is a more pitched-based sound, then first peak (fundamental) is more accurate.

Periodicity analysis is achieved by summing the audio spectrum and normalizing the data. If a signal is more noisy/less periodic, then the sum of the spectrum is greater. If a signal is more pure/more periodic, then the sum of the spectrum is less. Amplitude data is achieved by analyzing the incoming RMS power[6] and normalizing the data output.

We are able to plot main pitches by detecting onsets from the amplitude analysis. Frequency analysis is translated into note symbols that are plotted on the staff when those onsets occur, in the same method for drawing the graphical notation (as if a pen is stationary on the right side of the page and the paper is pulled from the right to left). See Figure 10.

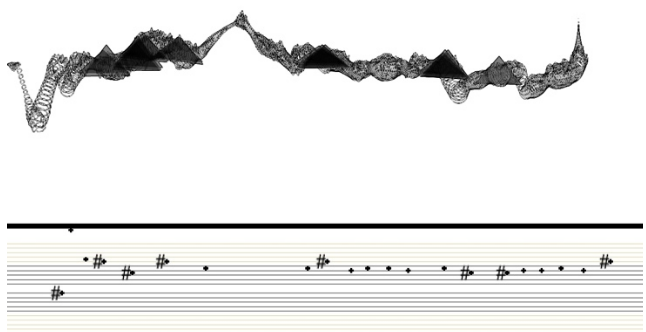


Figure 10. This image was created using the mix-shape option while analyzing a sound file using First Peak. Performed by François Houle, recorded at the Banff Center for the Arts. [14]

[3] ForScore is an application downloadable from the iTunes store where one can upload a PDF and use it as a digital sheet-music reader.

[4] A student could look at an Ocular Score image and play his or her imitation of that sound into the Ocular Scores tool and compare his or her image to the original. In other words, Ocular Scores could be used as a visual feedback instrument.

[5] The stylistic and aesthetic questions that are raised by the use of open notation such as the ones discussed in this paper, in which the harmonic language is left to the performers, is beyond the scope of this paper.

[6] RMS is "root mean square" and is the process used to determine the average power output over a long period of time.

This prototype continues to give us the option of creating still images to capture a gesture — 8 seconds or less (which is the size of the current window). The advantage of this second prototype is that we now have white space (silence) when there is no sound.

A number of parameters affect scaling and the character of image, and those parameters can be changed in real time to affect the image as it is created. We have similar parameters as in Version 1. See Figure 11.

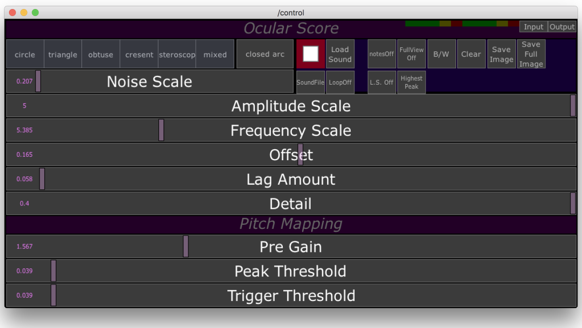


Figure 11. In Version 2, we have more choices of shapes to represent the sounds (top left menu), and we can switch from shape to shape in real time, as we decide the best way to represent the sound being performed.

In this prototype, the continuous recording of the images, creates a long score that can be saved and exported as a very long jpg. We can import this long score into other scrolling score software for future performances.

This tool can be used to create musical scores or to document a performance for repeated performance. The long score might also be used as a visual representation of a piece to analyze the larger structural elements of a work. See Figure 12.

Immediately below the large dots in Figure 12, we see a series of small dots. This is the "long jpg" that has been captured over a period of time — a visual recording of the musical events over about a 2-minute period. The staves below track the first harmonic of the sound event in the top window.

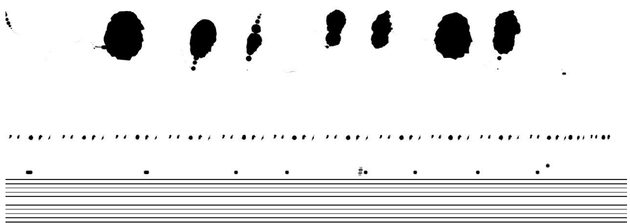


Figure 12. This image shows seven attacks — discrete "musical events."

We are currently working on extending the maximum duration of the long score feature in order to be able to record with no duration limitation.

6.3 Version 3: Performance Applications

The last prototype being developed is a performance tool. Ocular Scores™ Version 3 is designed so the composer interacts with the graphic score as it is being created in a performance context. The composer creates a dynamic score with the help of filters that are programmed in advance with a series of presets and manipulates the graphic score in real time by choosing certain shapes for the musical gestures and by using delays; changing the tessitura of the musical gesture; adding color and distortion; and changing the size and scale of the shapes. See Figure 13 for the parameter choices.

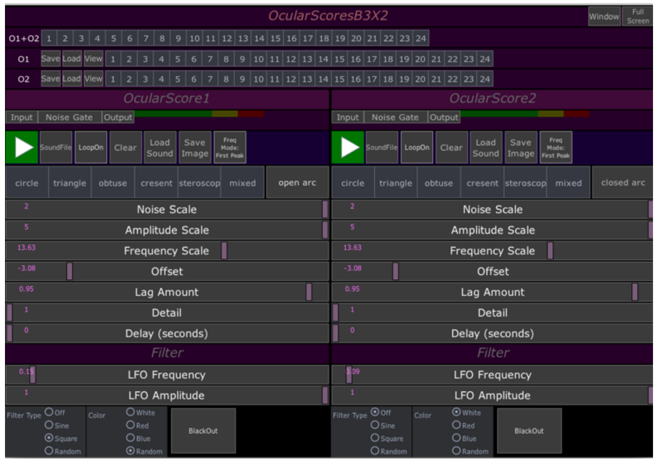


Figure 13. In Version 3, we have three rows of presets. The top row triggers the two rows immediately below. O1 is the preset for one Graphic Score and O2 is for a second Graphic Score. The bottom section shows how we use a low-frequency-oscillation (LFO) to shape the phrasing of the musical gestures by periodically removing the image entirely.

The following sections describe two examples of Live Structures compositions that utilize Version 3.

6.3.1 Murmuration/Murmure: Live Structures #3

"Murmuration/Murmure" is a "live structure composition" for two improvisers who each interpret the other's music. Two graphic scores, which are a visual interpretation of each improviser, are projected for the performers and the audience to see. See Figure 14.

The images of the score are created in real time from the analysis of their improvised sounds. The composer controls the "structure" of the work by manipulating "live" the graphic notation system (shapes), the flow and density of the work, which unfolds as a collaborative improvisation and a committed interpretation of musical code.

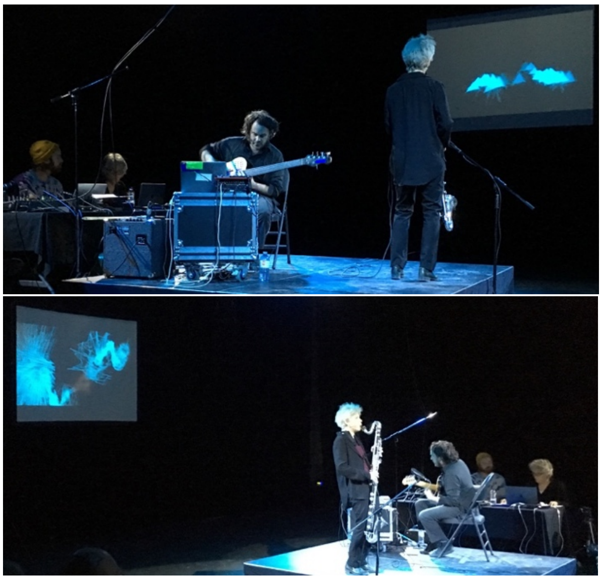


Figure 14. "Murmuration/Murmure" was premiered on March 2, 2019, during La Nuit Blanche, MNM Festival in Montreal, Quebec, Canada with Browne and Bouchard manipulating the Ocular Score™ and Lori Freedman with Alexandre St-Onge performers. [15]

"Murmuration/Murmure" was also performed at Center for New Music and Audio Technologies (CNMAT) at the University of California, Berkeley on April 10, 2019, performed by Kyle Bruckmann and Jacob Felix Heule. [16]

See Figure 15 for examples of scores from two separate screens that were produced during the performance of "Murmuration/Murmure" at CNMAT, UC Berkeley.

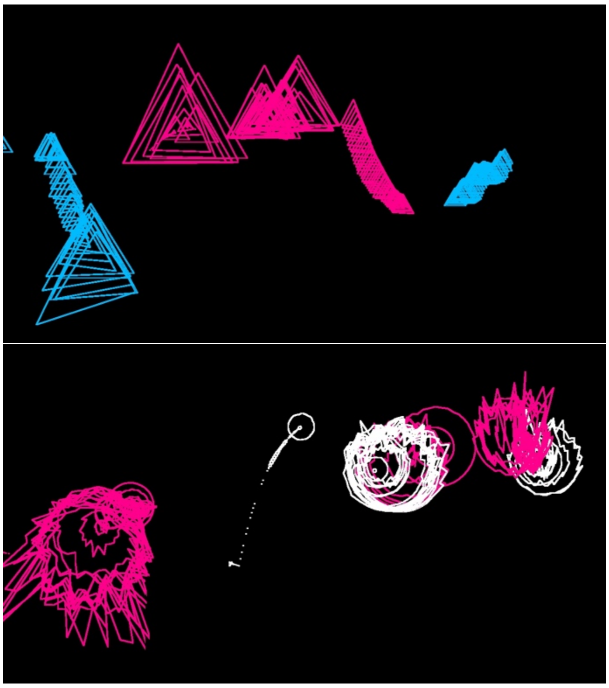


Figure 15. Two screens created simultaneously by two improvisers.

A series of iconographic images were created from the score "Murmuration/Murmure" [17]; these images should be reproducible during specific moments of the piece no matter what the musicians are playing.

6.3.2 Gaggle: Live Structure #5

"Gaggle" is another way of utilizing Ocular Scores Version 3x2, where the graphic scores that are projected for the musicians to interpret are created from a sound file that is also accompanying the musicians.

In "Gaggle," the composer is using a stereo sound file that was created from pre-recorded sounds of the two musicians from a previous project. The composer created two different mono sound files, and each musician performs the visual score that is created from one specific mono sound file. The score is created following a set of presets that are being launched at specific moments during the piece. [18]

7. MUSICAL CODE

Ocular Scores™ exist in the realm of interpretation, but as we worked with improvisers, we agreed that we could commit to a general musical code, mostly because most of what we had created was based on traditional principles or was intuitive.

- The score is scrolling from right to left (x-axis), so the musicians can read from left to right. The musician can choose to "read" at the left edge, in the middle, or on the right side of the score.
- The y-axis is the frequency range. If all the visual activity (shapes) are drawn at the top of the score, the musician is encouraged to play in the top register of his or her instrument (higher frequency). If shapes are drawn at the bottom of the score, it will represent the lower register.
- When the image is large and fills nearly the entire screen or page, it becomes a "landscape" or visual environment, and the frequency range is left to the discretion of the performer.
- When the shape is continuous, it could inspire a more continuous textures and/or a legato musical event.
- We use two ways to express dynamic: the density of the shape (transparency) and the size of the shapes and color.
- We use several shapes: circles, triangles, opaque shapes, crescent-like shapes, stereoscope-like shapes, and variations of those shapes. We conceived of the different shapes to express different textures. For example, we think of sharper shapes, with defined edges, as having more attack, more percussive beginning.
- We utilize three colors: white, pink-red, and blue-green. White is an open format. Pink-red is for expressing more presence, more dynamic. Blue-green is for softer, more transparent textures.

8. CONCLUSIONS

The initial goal of Ocular Scores™ was to explore different ways to interpret data into graphic notation and compositions. This initial assumption of using an analysis tool to create images that would be replicable has proven to be an inspiring way to produce images that can be interpreted by live musicians. The research and the three specific versions of Ocular Scores™ proved to be useful composition tools for Bouchard's creative work.

Bouchard's research grant is closing by the end of 2019. She is hoping to eventually create and distribute a simple app for Version 1 of Ocular Scores™ .

Acknowledgments

The Live Structures project [19] has received funding from the Canada Council for the Arts. Bouchard is grateful for residencies at matralab, Concordia University and at the Banff Center for the Arts. Visual designer and programmer Joseph Browne has been a true collaborator with tremendous creativity, problem-solving abilities, and patience. Many thanks to Sandeep Bhagwati, who supported this project from the start and kept asking pertinent questions as we were thinking our way through. Many thanks to François Houle, Lori Freedman, and Kyle Bruckmann for their invaluable feedback as we tested the tools.

9. REFERENCES

[1] Timelines of language, communication and semiotic systems. https://sites.google.com/site/ahistoryoflanguage/a-history-of-music-and-musical-notation

[2] The Schøyen Collection. https://www.schoyencollection.com/music-notation/graphic-notation/tibetan-yang-yig-ms-5280-1

[3] "Preset of the Mind" from Presets—Digital Shortcuts to Sound by Stefan Goldmann. Page 13. Published in 2015 by The Bookworm.

[4] TouchDesigner is a visual development platform produced by Derivative (www.derivative.ca).

[5] François Houle, clarinet, audio file recorded at the Banff Center for the Arts used to create the image for Figure 4. https://drive.google.com/file/d/1g619BMwWFwdI_NuVSYq2UP_kzSFumaGZ/view?usp=sharing

[6] Lori Freedman, bass clarinet, audio file recorded at matralab, Concordia University, Montreal, Quebec, Canada, used to create image for Figure 5. https://drive.google.com/file/d/1g619BMwWFwdI_NuVSYq2UP_kzSFumaGZ/view?usp=sharing

[7] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, and Riccardo Borghesi, "MuBu & Friends — Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/MSP," in ICMC, Montreal, 2009. [BibTeX] http://articles.ircam.fr/textes/Schnell09a/index.pdf

[8] Norbert Schnell, Diemo Schwarz, Joseph Larralde, Riccardo Borghesi. PiPo, A Plugin Interface for Afferent Data Stream Processing Modules. International Symposium on Music Information Retrieval (ISMIR), Oct 2017, Suzhou, China. https://hal.archives-ouvertes.fr/hal-01575288/document

[9] Collections of images created from recorded sound files performed by François Houle during our residency at the Banff Center for the Arts, Canada. https://vimeo.com/274766970

[10] Collections of images from the score "DROP." https://vimeo.com/337136681

[11] Premiere of "DROP" with Kyle Bruckmann. Other Minds "On the Nature of Music," Berkeley. https://vimeo.com/335315621

[12] "DROP" with Kyle Brenders, Banff Center for the Arts, Alberta. https://vimeo.com/336498980

[13] Lori Freedman, bass clarinet, audio file recorded at matralab, Concordia University, Montreal, Quebec, Canada used to create image for Figure 9. https://drive.google.com/file/d/10iz4hGZSRjsHInRkLdg22Mg_YvyklnQF/view?usp=sharing

[14] François Houle, clarinet, audio file recorded at the Banff Center, Alberta, used to create the image for Figure 10.
https://drive.google.com/file/d/10iz4hGZSRjsHInRkLdg22Mg_YvyklnQF/view?usp=sharing

[15] "Murmuration/Murmure," excerpt of premiere with Lori Freedman and Alexandre St-Onge, Festival MNM, Montreal, Québec, March 3, 2019. https://vimeo.com/331701538

[16] "Murmuration/Murmure," excerpt with Kyle Bruckmann and Jacob Felix Heule, CNMAT, UC Berkeley, April 10, 2019. https://vimeo.com/331704604

[17] A series of iconographic images from the score "Murmuration/Murmure." https://vimeo.com/337170319

[18] Premiere of "Gaggle" with Kyle Bruckmann and Jacob Felix Heule at CNMAT, UC Berkeley, April 10, 2019. https://vimeo.com/337212831

[19] www.livestructures.com

SOUND AS SCORE:
LIVE GENERATED AUDIO SCORE AND AUDIO SCORE BASED ON ACOUSTIC MEMORY

Elisabeth Schimana

Vienna
elise@ima.or.at

ABSTRACT

As a composer and musician of electronic music since the 1980s my medium is sound. When I was asked in 2009 to compose a piece for RSO (Radio-Symphonieorchester) Vienna I had to think about how to communicate with this sound body. I opted for what I do best - sound and listening. Since that year I have developed two different methods of communication with musicians - the live generated audio score, where the performers have to imitate the live generated electronic sounds they hear through a loudspeaker, and the audio score based on acoustic memory, where the musicians are given a set of sound samples for interpretation on their instruments and then in the performance this interpretation has to be played from memory. This paper examines the method, scoring, practice and rehearsal, as well as the artistic results using examples from *The Virus series* and the music theater piece *Pricked and Away*.

1. INTRODUCTION

1.1 Psychoacoustic impact on composition

In twentieth- and twenty-first century contemporary music, acoustic and psychoacoustic research has a great impact on the composition practice of composers. Both in instrumental and electronic music, composers deal with phenomena like critical bands (James Tenney, *Critical Band*, 1988), the hearing threshold level (Alvin Lucier, *Elegy for Albert Anastasia* for electromagnetic tape using very low sounds most of which are below human audibility, 1962–1965), or their focus is on listening itself (Maryanne Amacher, *ways of hearing* since 1980). In this way compositions are becoming an experimental test set up.

With continuing technological progress, the possibilities for generating electronic sound have become manifold. This allows electronic music composers to use the medium sound as opposed to writing, as a process-oriented score for acoustic instrument players. But if sound is to be interpreted by musicians, we need to ask questions about the properties of sound and the perception of sound by the listeners. Psychoacoustic studies like ASA (auditory scene analysis) by A. Bregman [1] as well as empirical phenomenological research such as that
conducted by Daniel Schmicking [2] provide insight into these subjects.

1.2 Aesthetics of perception

In studies about the perception of hearing, one distinguishes between universals, in other words innate, culturally independent attributes, and attributes which go along with conscious hearing and are connected with personal experiences and cultural backgrounds. My compositions primarily deal with conscious hearing and address a very specific target group: the professional musician.

If sound is the score's medium, this establishes a wide field for research on the subject of interpretation and the aesthetics of perception. Helga de la Motte-Haber [3] asks whether an aesthetics of perception in musicology really exists because in addition to the analysis of the compositions it would also have to focus on the listeners.

1.3 In the open field, the composition as an experimental test set up

Usually acoustic and psychoacoustic research takes place in the specific situation of a laboratory. My compositions, however, are conducted in an open field. No environmental influences are blocked out, no exceptional circumstances are generated; the situation corresponds with common practices of musicians and composers. The task is, as usual, to interpret a score, and although the setting may be unconventional to many of the musicians, the basic approach is still a familiar one. And the creative aspect exists, the participant is called upon to actively contribute in an artistic way within the framework of the specific conceptual formulation. Therefore, my compositions are an experimental test set up in the open field.

Sandeep Bhagwati describes his approach - the Elaborate Audio Score (EAS) [4] as follows: "...this term denotes a type of score that uses headphones as its interface to the musician and conveys musical information primarily via acoustical messages."

Headphones are a forbidden planet in my open field approach. All disturbances are welcome and part of the set up.

2. METHODS

2.1 Live generated audio score

I developed this method for the *Virus series,* starting in 2011 with *Virus #1.0*[1] for string quintet. In the beginning there were several major questions: How should a live generated sound score be designed? How should the set up be organized? Who hears what, and why does somebody choose a specific interpretation? Where are my own listening limits? And how is it possible to merge the digital, live generated score with the interpretation of the acoustic instruments?

2.1.1 Basic considerations

The electronic resonating sound body[2] is generated live during the performance and is the audio score for the acoustic instruments. It corresponds to the image of a host cell. The acoustic body corresponds to the image of a virus, because the musicians have to attach and adapt to the electronic sound body, they penetrate into it and a synthesis between the two resonating bodies results.

The *Virus series* is an expedition into acoustic perception, a sounding of the responses of our brains in the span of milliseconds, a plea for the precise acoustic moment. It deals with the question "And what do you hear?" This question is dedicated to the audience, the musicians and myself. It implies that every self will experience something different.

2.1.2 Set up

The musicians sit or stand spread out in the space each in front of a loudspeaker and try to play what they hear as precisely as possible on their instruments. The electronic sounds – the sound score coming from the loudspeaker - as well as the interpretation by the musicians are audible. The instruments are unplugged, so the speakers project only the sound score. It is important to consider the distances between the musicians, in order to give each participant the possibility to focus on his or her own part within the sound score. A minimum of 3 meters for instruments with middle or high frequencies and a minimum of 5meters for instruments with low frequencies are necessary. The composer generates the score live and is therefore a part of the responding system, and together with the musicians she builds a feedback loop. Like the musicians, the composer listens and reacts to what she hears, making decisions for the further progress of the composition. The audience sits amongst the musicians.

2.1.3 Acoustic instruments and electronics

The digital electronic sound body in the *Virus series* has the job of doing what it does best: the precise execution of an algorithm. The acoustic sound body also has the job of doing what it does best: the imprecise execution of an algorithm.

I will explain this statement in the following simple example: a digital sine-wave oscillator oscillates with a frequency of 440 Hz for 2000 ms - break for 1000 ms - repetition - break for 500 ms - repetition. The program will execute this algorithm precisely. Translated for a musician this algorithm would sound like this: play the note A (440 Hz) for 2 seconds, pause for 1 second - repeat - pause for half a second - repeat. Compared with the machine the musician will execute this algorithm imprecisely and as the number of repetitions increase, this imprecision will also raise.

This fact demonstrates one of the most interesting side effect of this method - emerging fuzziness.

2.2 Audio score based on acoustic memory

I first used this very different approach in 2017 for the piece *Vast Territory. Episode 1 Lily Pond*[3]. This first attempt was for 7 minutes only, for strings and wind instruments. The major question was: How does our acoustic memory work. And for how long is this possible without the addition of a written score? In my online research I could not find any relevant answers - this, of course, might be because of my limited financial and time resources. In 2018, I composed the music theater piece *Pricked and Away*[4] working with this method for the ensemble part of the piece.

2.2.1 Basic considerations

The musicians follow a sound score played by memory and are thus tied to the oral tradition. It is like telling a story from aural memory. The audio score consists of field recordings lasting 15 to 45 seconds and are given to musicians in advance for interpretation and memorization. Working with field recordings in audio scores is quite common, but usually the musicians get them during the performance via headphones for interpretation. The only approach tied to the oral tradition in contemporary music I know is Eliane Radigue's way of communicating her compositional ideas to musicians, and she even asks these musicians to pass on the piece through oral tradition to another musician. This other musician has to reference the musician from whom the information is coming. In this way a line of references is created[5].

1 http://elise.at/project/Virus_1

2 I distinguish between an electronic sound body - electronically generated sounds like sine waves or noise, and an acoustic sound body - sounds generated by acoustic instruments

3 http://elise.at/project/Vast-Territory

4 http://elise.at/project/Pricked-and-Away

5 This information comes from a conversation I had with Eliane in December 2018 at her home in Paris.

Field recordings are a very complex type of sound material. They stimulate the imagination or trigger memories about a specific environment mostly connected with very specific emotions.

At the same time field recordings consist of multilayered sound events and most of the time it is not possible to translate all this acoustic information at once for interpretation on the acoustic instrument. As human beings we filter this information in an environment and concentrate on what seems to be relevant for us at a specific moment. A microphone is stupid and will transmit all acoustic information out there. The musician then has to decide which acoustic information is relevant for interpretation, and from musician to musician this can differ quite a lot!

2.2.2 Set up

Each musician follows his or her very personal interpretation of the sound files in regard to a specific time line score. In a short piece each musician has a stopwatch to help him or her execute the time line. In a long piece the timeline has to be a moving visual score with a cursor to indicate the exact time. The individual timeline scores for each musician have to be synchronized. All sounds - the interpretation of the field recordings - are played by memory, the original audio score - the field recordings - is inaudible.

3. SCORING

What is a score? It is a special form of recording - notation - in a musical context. The design of a score depends on its function, whether it is a notation for the composer as a stored memory, or a performance score, which serves as a tool for communication. In performance scores composers notate musical structures for interpreters. Usually the notation is a visual media and the communication with the interpreter implements a reading process and translation process into sound. If the score is sound, the reading process as well as the translation from one medium to another is skipped.

Are the programmed algorithms, the code - the notation for the machine a score?[6][5]

6 In a book sprint I collaborated on notation we came to the following conclusion: Program == Score? Let us take this opportunity to raise another difficult question: Is the code or the patch the score? In much computer music, the composer (who may also be the performer) creates a piece of software as a patch or as code. It can be argued that this code is the score. However it is important, if not vital that the symbols in a score should have the potential to be executed by any, or at least other, software/program with any hardware, and/or any human being able to connect to the context. Chosen symbols for a score should go beyond a specific software or hardware, creating a metalanguage for interpretation. Otherwise it is not, in some sense, a score, rather it is an encoding of a specific piece and performance of music. It is a notation of it, perhaps too specific to be a score. (Booksprint 2012)

3.1 Machine notation for the *Virus series*

To communicate with the machine I use the program MAX/MSP and Modalys for physical modeling and the interfaces mouse, keyboard and MIDI controller.

To be able to concentrate on listening during a performance the patch and controler design is of immense importance. A maximum reduction is desirable in order to ensure mobility and flexibility. The main objects are poly~ to create multiple voices and pattrstorage to store and transform parameters. During the performance I mainly control the volume of each oscillator[7] and some very specific parameters, depending on the musical needs of the piece.

3.2 The sound material for the *Virus series*

The digitally generated electronic sounds are either very basic electronic materials like sine waves, triangles, filtered saw tooth with a specific ADSR, pitch and meter or physical models of tubes, strings or membranes.

The main objective is to create a “different” similarity to the sound properties of the instruments - e.g. a high flute tone will sound very similar to a sine wave, but still different. This corresponds with the idea that the audio score consists of material which cannot be created on an instrument and which is generally poor in harmonics, thus the instruments, the viruses, add timbre, or color. The sounds created by physical modeling constitute an exception, for they are rich and similar but at the same time totally artificial.

This concept results in a high level of fusion of the electronic and acoustic instrumental sounds.

3.3 Notes for the composer for the *Virus series*

The notation on paper for myself are notes about the ability of each instrument - I noticed that every musician and every instrument differs in possibilities - a table with frequency, meter and ADSR values, and notes about the structure of the piece.

f	H2 30.87 HZ
m	15000
ADSR	A1500, D1000, S0.9, St3000, R600

Table 1. Example of notated parameters of one oscillator

The only information passed to the musician is the pitch information - e.g if an instrument has 7 oscillators only these seven pitches will appear in the audio score. But there is no information as to when and in which combinations this is to occur. That is created during the performance.

7 Each instrument has 5 to 10 or more well defined oscillators, e.g. a sine wave with definite pitch and meter, which can be played simultanously for pattern creation.

3.4 Sample and timeline score for *Pricked and Away*

The audio score consists of 13 sound samples for each of the 4 musicians, all different, but with similar sound properties in one category - e.g. sound sample 1 for flute has properties similar than sound sample 1 for harp. This creates similarities but at the same time differences, because an interpretation on the harp is quite different from that on the flute. Each sound sample has a duration of anywhere between 15 to 60 seconds at intervals of 5 seconds. I consider this to be quite important in creating a clear feeling of timing, which manifests itself after a while.

The musicians receive the sound samples quite a long time before the actual performance. This gives them ample time to listen to them repeatedly and get more and more acquainted with the microstructure of the sound file.

After this process, I create a timeline with a structure that determines which sound file is to be played at which time during the piece. For this part I had to use a visual score, since it is not possible to remember a duration of 30 min. During the performance each musician will see only his or her part, like in a usual written score. I opted for synchronized tablets on which the score is a video file with a cursor, which indicates the current time. This method has been borrowed from electroacoustic music practice, which is used when a piece is created with multiple layers of field recordings.

As mentioned in 2.2.2 the original sound files are inaudible and the musicians perform the sounds from their acoustic memory.

4. PRACTICE AND REHEARSAL

4.1 Creation and rehearsal process

In 2011, when I started with my first *Virus* for string quintet, I arrived in Kiev to work with the New Music Ensemble Ricochet with a more or less finished programmed patch, and the whole ensemble was ready to rehearse. I quickly realized that this was not the way to do it! I understood that I had to work more individually or with specific instrumental groups to hold the musicians' attention and keep them from losing patience.

After seven years of experience, I have developed a very economical and meaningful way of creating and rehearsing a *Virus.*

a) I start by meeting with each musician for about half an hour to collect information about the instrument, limits in performance techniques, pitch and speed. During this process musicians usually offer me a lot and we already try to push limits. Because I work with the highest and deepest frequencies at the very limits of the instrument, this is a very important half an hour. At this point musicians would probably say: I could even play one tone lower if I practiced a bit. No technical equipment is needed, just the acoustic instrument.

b) I create the pitch/meter system based on the collected information and think about the structure.

c) One more individual meeting of an hour to verify what I have programmed and for the musician to get familiar with the listening and interpretation process.

d) Based on the results of c) I reprogram the patch, create the interface design and rehearse.

e) First rehearsal with the ensemble in the space where the performance will take place. This usually occurs a day before the concert. The amount of audio equipment depends on the number of musicians and is a lot in most cases. Remember, each musician has his or her own speaker, so with 20 musicians you need 20 speakers plus the subwoofers for low frequency instruments. I call it a living acousmonium.

f) Main rehearsal and concert.

We rehearse parts, possible emerging patterns, transitions and a lot of time is needed to tune the system to the needs of the musicians. All in all, this usually amounts to 3 hours of rehearsal time, and in this way, we manage to fit the rehearsal concept into the ridiculously tight schedule allocated for concert preparation.

I am part of the system, my presence is needed.

In *Pricked and Away* the process is similar and different at the same time.

a) I select and prepare the sound material and pass it on to the musicians. At this stage it is test material, not the final audio score.

b) Individual meetings with the musicians, who offer possible interpretations on their instrument, discussions and a joint search for possibilities.

c) I edit the sound files for the final version of the audio score and prepare the timeline structure.

d) Two ensemble rehearsals, during which musicians are allowed to use the notes they took for their interpretation. Continuous fine-tuning of the timeline structure and microstructure of the interpretations.

e) Main rehearsal and concert at the concert space

Written notes for interpretation of the sound files are no longer allowed.

I am not part of the system my presence is not needed.

4.2 Practice

Whereas Sandeep Bhagwati [4] mentions that “the first approach to an audio score is very similar to that needed for a conventional new music score that uses many non-standard symbols” I wish to focus on the differences.

a) An audio score allows us to create extremely complex patterns with very simple commands. The main command is to play what you hear as precisely as possible in terms of all parameters including pitch, rhythm, timbre, etc. Aware that a perfect copy/imitation is not possible, a creative translation process is triggered in the musician. But - and this is very important in my work - it is not about improvisation, the target has to be a perfect copy/imitation. This directed focus gives rise to very interesting results, especially in microstructures. The precondition is to listen intensely and carefully.

With a live generated audio score like in the *Virus series* it is possible to create e.g. timeshifted pulse oscillations (each musician has a different meter) without complicated written scores/code or click tracks via headphones.

b) Musicians do not need to decode a visual medium, e.g. a new music score that uses many non-standard symbols. As musicians told me this decoding process usually demands a lot of time and very often in the end the acoustic result is not satisfactory. One of my favorite musicians, I composed a solo violin *Virus* for him, told me that this was the reason he stopped playing in a contemporary music ensemble. An audio score works from the other end: the acoustic result is the starting point.

c) Like in oral tradition, the collaboration between composer and musician is an intimate one. This first individual rehearsal time, see 4.1, is of great importance. It creates a personal connection and deeper understanding of each other. In aural communication it is easy to adapt to the personal abilities of a musician and at the same time push the limits a bit. With written scores, this process is eliminated, usually composers deliver the score and that is it.

d) A specific characteristic of the *Virus series* is, that I am part of the system, and together the musicians and I build a recursive feedback loop. I listen and react, just like the musicians, and decide the further progression of the piece. We are talking about decisions made in milliseconds, and that is only possible because our auditory system is so fast! In order for this feedback loop to work at optimum capacity intense concentration is needed. I think it comes close to a state of trance. There is no time to think, just time to act!

e) In my experimental test set ups - the audio scores – I have gathered a lot of information about how my target group, the professional musicians and also myself, perceive sound and sounding structures. Most of the observations I have made confirm psychoacoustic research results. But - and this is most interesting - professional musicians can go beyond this for example: the auditory stream theory by Tougas and Bregman states: "The principle of grouping tones by their frequency proximity was found to dominate over the principle of grouping tones that follow a smooth trajectory." [6] I confirm this, but musicians start to create patterns beyond that.[8] This is a very beautiful moment for me.

5. ARTISTIC RESULTS

So far my *Viruses* increasing and spreading over the globe, which is exactly what viruses should do. The series includes #1[9] – for one group of instruments, e.g. strings, or solos - #2[10] for two groups of instruments, e.g. strings and percussion – and #3[11] for mixed ensemble.

Audio scores based on acoustic memory were used in *Vast Territory. Episode 1 Lily Pond* [12] and *Pricked and Away*[13] only. A specific aspect of this last work was the combination with a text-speaking performer which caused specific needs I still have to work on.

6. CONCLUSIONS

Sound as score has an innovational potential. Its impact in terms of aesthetics and performance practice is still relatively unexplored terrain. Also, the focus on the listeners in the field of reception aesthetics is a marginal one. In particular the live generated electronic score, so-called electronic music, interpreted by acoustic instruments is virgin soil. From the perspective of an electronic music composer the two methods I describe here are a pleasurable and artistically satisfying way to communicate with musicians.

Acknowledgments

Compositions leading to this paper were supported by bundeskanzleramt:kunst, SKE austro mechana and all ensembles and festival organizers who commissioned a piece. Special thanks to Christian Scheib who commissioned a piece for RSO Vienna – this was the starting point.

7. REFERENCES

[1] Albert S., Bregman, *Auditory Scene Analyses*, Massachusetts Institute of Technology, 1990

[2] Daniel Schmicking, *Hören und Klang,* Könighausen & Neumann, 2003

[3] Helga de la Motte-Haber, Reinhard Kopiez [Ed.], *Der Hörer als Interpret*, Peter Lang GmbH, 1995

[4] Sandeep Bhagwati, *Elaborate Audio Scores: Concepts, Affordances and Tools*, Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18

[5] The Contributors, Booksprint 2012 *On Turtles and Dragons and the Dangerous Quest for a Media Art Notation System*, Hrg. Time's Up, Linz, 2012

[6] Yves Tougas and Albert S. Bregman, Crossing of Auditory Streams, Journal of Experimental Psychology: Human Perception and Performance, 1985

8 When I play pulses in the highest frequency range of the instrument and at the same time in the lowest frequency range most of the musicians will follow either the high or the low pulses. A few of them will start to create patterns jumping from one frequency range to the other.

9 http://elise.at/project/Virus_1

10 http://elise.at/project/Virus_2

11 http://elise.at/project/Virus_3

12 http://elise.at/project/Vast-Territory

13 http://elise.at/project/Pricked-and-Away

RECENT AUDIO SCORES: AFFORDANCES AND LIMITATIONS

Charlie Sdraulig
Stanford University
Stanford, CA, USA
sdraulig@stanford.edu

Chris Lortie
Stanford University
Stanford, CA, USA
cjl93@stanford.edu

ABSTRACT

A growing body of contemporary composers produces audio scores where sound is the integral mediator between the composer and performer. While many musical scores deploy some form of symbolic visual representation of sound or movement, audio scores represent information and instructions in the same domain as the performed product. This paper aims to survey the affordances and limitations of audio scores enacted thus far. Within the field, we identify two primary sub-categories associated with the temporal relations between performer and audio score: reactive and rehearsed. Louis d'Heudieres' *Laughter Studies 1-3* (2015-16) and Lara Stanic's *Open Air Bach* (2005, rev. 2013) are examples of reactive audio scores. Representative examples of rehearsed audio scores include Carola Bauckholt's *Zugvögel* (2011-12) and Cassandra Miller's *Guide* (2013). These primary sub-categories may be combined and hybridized to varying degrees, as in Carolyn Chen's *Adagio* (2009). Finally, in light of our survey of the possibilities offered by audio scores, we propose some further avenues of exploration for creative practice.

1. INTRODUCTION

This paper aims to survey the affordances and limitations of a growing body of audio scores, defined here as scores which employ sound as the primary means of communication between composer and performer.

Most existing literature on this topic comprises self-reflective analytical commentary by composers [1] [2] or discusses audio scores within the larger theoretical frame of real-time scoring strategies (alongside animated or video scores, for example) [3] [4]. In the latter context, d'Heudieres analyses Gavin Bryars' *1, 2, 1-2-3-4* (1970) where each performer listens to popular songs over headphones and spontaneously imitates their instrument's part in the recordings. Taking Bryars' work as his primary historical exemplar, d'Heudieres stresses real-time applications in his definition of the medium: "an audio score is one in which the instructions from composer to performer are communicated *in performance* through sound" (our emphasis), thereby excluding recorded samples assisting performers solely in rehearsal [4, p.18]. Alternatively, Bell [1, p.43] [3, p.2] conceives of his 'audio-scores' as extensions of visual symbolic notations, explicitly incorporating 'learning by ear' oral tradition practices within written ones (e.g. pitch cues and click-tracks delivered over earpieces). For Bell, the audio cues merged into this hybrid format primarily serve as solutions to rehearsal and performance difficulties when realizing complex written tempi or microtonal pitches. By contrast, Bhagwati [2, p.25] adopts an expansive definition of what he terms 'Elaborate Audio Scores,' simply emphasizing the primacy of auditory communication between composers and performers as the sole requirement. This definition encompasses work conforming to both d'Heudieres' and Bell's positions, and is the one we have followed here.[1] Bhagwati goes on to discuss the various "conveyance modes" he typically employs in his own compositions [2, p. 26-28], including event and instruction cues for executing sounds or choreographies, as well as prompts for improvisation, and imitation of samples or styles.

To the best of our knowledge, no general analytical survey exists which sketches the diversity of recent efforts in the nascent field of audio scores; both Bell [1, pp.46-54] and Bhagwati [2, pp.25-26] only provide brief descriptive lists of works that parallel or influence their own creative endeavors. As such, our paper seeks to both complement this existing literature and extend its scope by drawing scholarly attention to bodies of work hitherto un- or under-examined in this context.

As a methodological lens for this survey of the state of the art, we employ James J. Gibson's [5, p.127] notion of affordances: the potential actions made possible by an object or environment to a given individual—a concept that implies a mutually influencing, transactional relation between actor and object. A material format alone does not wholly determine the action possibilities it affords; composers and performers (i.e. the actors in this context), as well as audio scores (i.e. the object), are themselves situated and dynamically shaped within wider networks and

[1] Of course, whether a given hybrid score (akin to Bell's practice) uses sound as a *primary* mediator between composer and performer, or not, will be open for debate on a case by case basis.

histories of cultural practice. These practices mediate and constrain potential relations: for instance, the act of deploying a sound recording *as if* it were a score suggests a translation of prior scoring practices across media; equally, functioning as a score is just one of the many potential use cases afforded by recorded or generated sound. Nevertheless, at this particular intersection of cultural practice and material format, we contend that audio scores representing information and instructions in sound afford some *distinct and different* possibilities to composers and performers when compared to scores which deploy some form of symbolic visual representation of sound or sound-producing movement. Via the contrasting pieces we have taken as case studies (all primarily associated with Western experimental art music), supplemented by a table of related works (see Figure 2 toward the end of this paper), we intend to offer a preliminary sketch of some of the affordances enacted thus far in this dynamic, *in process* contemporary field.

Categories of Audio Scores

Within this field, we identify two primary sub-categories associated with the temporal relations composed between performer and audio score: reactive and rehearsed.[2] On the one hand, performers primarily react to the audio score *during* performance; on the other, the audio score shapes the performers' interpretations in rehearsal, well *before* public performance.[3]

2. REACTIVE AUDIO SCORES

2.1 Louis d'Heudieres, *Laughter Studies 1-3*

Louis d'Heudieres' *Laughter Studies 1-3* (2015-16)[4] for two vocalizing performers [6] are examples from our reactive category that translate a practice associated with visual scores into a different format to enact distinct affordances. In d'Heudieres' series, each performer listens to a different audio score on headphones and alternates between vocally imitating and verbally describing what they hear to the audience.

The audio scores feature a fast, collage-like succession of many and diverse samples, which are edited and processed to varying degrees. The performers are asked not to overly familiarize themselves with their audio scores beforehand; rather, they should spontaneously react to the often unpredictable changes and transformations between sounds. This arbitrarily-imposed constraint is a kind of translation of sight-reading practices, though it differs in some fundamental respects. Static visual scores tend to represent sounds in a metaphorical spatial configuration [7, p.23] allowing performers to anticipate and read ahead of their performed actions at any given moment. In addition, a performer may infer the likelihood of future events in a given visual score based upon characteristic features of the dimensions prioritized by the composer, which all serve to situate the score with reference to similar examples from similar genres. By contrast, d'Heudieres' *Laughter Studies* lock the performers into a relation with the audio score that is always reactive—the format ensures there is little way of hearing or inferring ahead.[5] d'Heudieres couples this limitation with an affordance of sound reproduction technologies: the format mediates and renders commensurable a vast array of sound sources—recorded, synthesized, and processed, etc. Potentially *any sound* within the range of human hearing and the frequency response of the microphones, speakers or headphones etc. could be reproduced, manipulated and organized in countless ways. As if to take full advantage of this, many of the sounds in the *Laughter Studies* audio scores are crowdsourced from online repositories.[6] The resulting high degree of unpredictability and diversity in the sounds deployed by d'Heudieres encourages varied modes of performer listening, imitation, and description: highly affective sounds, such as laughter or crying, invite imitations and descriptions emphasizing semantic associations; by contrast, more abstract sounds or processing invite greater attention to acoustic qualities, etc.

However, unlike many visual scores featuring transcription, d'Heudieres does not explicitly transcribe and prioritize certain dimensions of his chosen sounds *beforehand* and on behalf of the performers; no guidance is given on *how* the performers should accomplish their tasks, and few concessions are made to idiomatic vocal conventions in the choice of sounds. Rather, the particular intersections of translated sight-reading practices and audio score format we have enumerated allow d'Heudieres the possibility of *recomposing the relations between performers and score*—the vocalists transcribe and mediate freely in their own, individual way. The ease or difficulty of the task over time for those specific performers are dimensions of compositional exploration, traceable via the *variable fidelity* of their imitations and descriptions—sonic and gestural stumbles or hesitations demarcate each performer's limits. Overall, the performers' personas convey no pretensions toward mastery of their materials, inspired interpretation,

[2] Bhagwati raises and rightly critiques a seemingly similar division between "situative and fixed audio scores" [2, p. 30]. Rather than emphasize the nature of the score-in-itself, we stress the character of the encounter composed *between* score and performer. For example, when an audio score is heard for the first time by a performer, it may make little difference to them if the score is generated 'live' (i.e. situative) or fixed beforehand. But 'live' or fixed, the relation between performer and score in this scenario will be reactive.

[3] As will become apparent later in our discussion of Carolyn Chen's work, these categories are not inviolable divisions; they draw attention to the primary emphasis of the relation composed between performer and score. For example, a performer will likely studiously rehearse Lara Stanic's work and set-up, gaining experience and facility in negotiating the unpredictability arising out of the complexity of the system. However, the primary emphasis of the composition remains the performer's spontaneous reaction to, and negotiation of the audio-score in real-time.

[4] Documentation: http://www.louisdheudieres.com/works.html [6]

[5] Bhagwati primarily notes this affordance as a limitation [2, p.29], again serving to highlight how constraints may both open or foreclose creative possibilities depending upon how they are framed and negotiated within a given work.

[6] For example, freesound.org, a collaborative online database of Creative Commons Licensed sounds

or faithful reproduction. Rather the affordances enacted in the relations between the performers and this reactive audio score evoke the register of candid, theatrical improvisatory play and games.

2.2 Lara Stanic, *Open Air Bach*

In a similarly playful vein, not only does the performer in Lara Stanic's *Open Air Bach* (2005, rev. 2013)[7] [8] react to an audio score during performance, but this audio score reacts dynamically to the performer's actions as well. Stanic enacts the affordances of the varied assemblage of technologies at her disposal to compose a feedback loop. In this, performer, computer, microphones, and loudspeakers mutually influence one another, interacting via live processing of recorded sound. Three external microphones follow the amplitude of the output of three loudspeakers attached to the performer's body. The amplitudes picked up by each microphone determine the speed and pitch of the live, computer processed playback of each part of an instrumental recording—the third movement of J.S. Bach's Sonata in E minor for flute, cello, and harpsichord BWV 1034. The performer's goal is to achieve the 'correct' playback of the recording through silent movement and somewhat awkward gestures. The closer the performer moves to the microphones, the greater the amplitude of the signal they pick up, and the more accurate the playback becomes; however, the performer must constantly adjust their distance from each microphone due to the continually changing volumes of the recordings [9, pp.164-165].

Generally static formats and their associated practices, such as conventional symbolic visual scores, allow the performer varying types and degrees of interpretative flexibility centered upon a largely stabilized, if relationally defined object. Many of the score's details and their situated meanings are negotiated before the act of public performance. By contrast, thanks to live electronic sound processing, Stanic composes a situation where the score-object itself is highly mutable and fluid *during* performance. Stanic defines an interactive system, which dynamically responds to *and* scripts the performer's movements. Since the audience also hears these processed, fluctuating sounds, the audio score becomes an object of aesthetic interest as well.[8]

3. REHEARSED AUDIO SCORES

3.1 Carola Bauckholt, *Zugvögel*

Thus far, our case studies have surveyed a reactive relation between performer and audio score. By contrast, the reed quintet *Zugvögel* (2011-12)[9][10] by Carola Bauckholt [10] invites players to interact with and *rehearse* the audio score well before public performance. The audio recordings, which consist of bird calls of various species, are not played in performance; instead, the quintet members are instructed to familiarize themselves with these bird calls and memorize all their nuances in order to reproduce them on their respective instruments.

Bauckholt's use of the audio score format directs this activity towards high-fidelity transcriptions of bird calls rather than reduced abstraction or overt musicalization (the obvious historical precedent for the latter occurs in the works of Olivier Messiaen) [11]. By using recordings of her sources, Bauckholt's audio score affords a higher degree of specificity and dimensionality to performers than most visual, symbolic representation of those sources (particularly in regards to spectro-temporal variation in timbre, for example.)

In *Zugvögel*, each performer establishes a distinct relationship with the recordings due to their objective of producing a faithful imitation through whatever means necessary. This mimetic process begins with the task of parsing each recording's most salient elements. In contrast to the traditional practice of bird call transcription in a conventionally-notated medium, the recorded format omits much of the symbolic filtering and prioritizations of the composer. It instead defers any such filtering to the instrumentalists of the reed quintet, allowing a more intimate understanding of the instruments' capabilities to inform a precise rendering. Hence, the performers' personal knowledge of their instrument's compatibility with the source material enables a higher degree of fidelity in its reproduction. Through this process, the performer may also come to discover previously hidden action-potentials in relation to their instrument.

It should be noted that each recording is symbolically transcribed by Bauckholt in an accompanying visual score. She adds in her prefatory notes, however, that "the notation should only be taken as a guide," [12] suggesting that the notated transcription holds only a supplementary role in relation to the audio score. However, this visual aid helps to mitigate one of the audio score format's primary limitations: the act of memorization. Because the minute features of each recording must be encoded into memory and recalled in performance, the musical information itself is subject to variability and even corruption over time—employing a visual score in conjunction with the audio score seems to effectively redress this limitation.

3.2 Cassandra Miller, *Guide*

Guide (2013)[11] for 10 singers [13] by Cassandra Miller offers a similar implementation of the rehearsed audio score in emphasizing the specificity of its recorded materials; however, where Bauckholt instructs the performers to *imitate* the sound sources as closely as possible, Miller instead opts for a *qualitative embodiment* of the audio score. The players are directed to thoroughly familiarize them-

[7] Documentation: https://youtu.be/mdbK0S_PvM4 [8]

[8] See Figure 2 for further examples of reactive audio scores and associated affordances

[9] The title translates to "migratory birds"

[10] Documentation: https://youtu.be/KEAHxLNyVxw [10]

[11] Documentation: https://soundcloud.com/cassandra-miller-composer/guide-exaudi [13]

selves with a recording of "Guide me, O Thou great Jehovah" as interpreted by American folk singer Maria Muldaur in 1968. The particular recording was selected, according to Miller, because its melody exhibits "swoops over a large tessitura that, more than anything, sounds like it feels good to sing" [14]. As in *Zugvögel*, *Guide's* visual score serves only a supplementary function; Miller employs a "quasi-neumatic" open graphic notation, providing only the starting pitch, tempo, and general contour of each vocal line in order to support a flexibility of interpretation.

"Singability" is prioritized in agreement with the free-flowing, improvisatory characteristics of Muldaur's original interpretation. These characteristics need not be translated through a direct imitation of the recording by rendering all its time-based nuances; instead, the audio score affords a *qualitative embodiment* of Muldaur's vocal characteristics as the basis for a public performance of *Guide*. Muldaur's vocal identity is appropriated and filtered through the bodies of each singer; the audio score allows access to the minute sonic details of Muldaur's registration, phonation, and articulation. In her instructions, Miller references the score's resemblance to oral tradition; the composer acknowledges this larger cultural practice in *Guide*, whereby material is shaped by the corporeal and performative identities of multiple generations of interpreters [15, p.89]. The liberties granted to the singers allows this phenomenon to be communicated in an arguably more acute, concrete way than a symbolic transcription.[12]

4. HYBRID AND CROSS-MODAL APPLICATIONS

Carolyn Chen, *Adagio*

The two primary sub-categories of audio scores we have outlined, reactive and rehearsed, serve well as an initial distinction to guide our survey; however, these categories may be combined, weighted, and hybridized to varying degrees. In Carolyn Chen's *Adagio* (2009)[13] [16] three or four performers listen to an audio score on headphones, synchronizing emotive facial expressions with the music—specifically, an excerpt of the second movement from Bruckner's 7th Symphony as performed by the Münchner Philharmoniker, conducted by Sergiu Celibidache live in 1994.

The distinct affordance this work enacts via its associated entities is the encapsulation of a chain of translations of *specific* media, histories, and socio-cultural phenomena. First, a recording encapsulates Celibidache's interpretation of Bruckner's visual score and its associated practices. Then, an excerpt of this *specific* recording is subsequently appropriated by Chen and situated as an audio score. Bruckner's visual score also acts as an adjunct to this audio score, annotated with affective descriptors that presumably translate and embed Chen's own listening practices and relations with the excerpt [17]. The performers *rehearse* and memorize descriptors such as "elevated," "yearning," and "solitary" [18]—often suggesting, but not prescribing the exact facial expressions to perform. The audio score provides not only a time-based structure for these performative events (akin to a click track[14]), but a live stimulus which the performers react to, mediating Chen's affective instructions. This 'private' listening practice is then transposed into a concert setting. The audience hears no sound; they voyeuristically observe the staging of a 'private' listening experience they know well, speculating upon the incomplete trace of the aforementioned chain of mediations. At the very least, the performers' exaggerated facial expressions (as in Figure 1) may clue the audience into the 'romantic' qualities of the unheard musical referent.

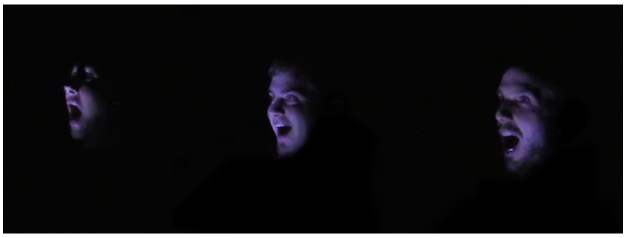


Figure 1. Still from a performance of Carolyn Chen's *Adagio* by Constantin Basica (left), Chris Lortie (center), and Charlie Sdraulig (right). Filmed by Dave Kerr. Used with permission.

5. CONCLUSIONS

Overall, in contrast to their visual counterparts, audio scores often represent information and instructions in the same domain as the performed product. As such, many audio scores afford the imitation and embodiment of diverse sound sources with varying degrees of fidelity. Unlike many visual scores, a composers' conception of a voice or instruments' capabilities need not determine the fidelity of the performers' realization—a somewhat standardized vocal or instrumental approach can be deemphasized in favor of an individual musicians' knowledge of the action-potentials of their specific instrument (as in d'Heudieres' *Laughter Studies*, Bauckholt's *Zugvögel*, and Miller's *Guide*). Distinct affordances are enacted through the intersection of prior cultural practices and the audio score format: in *Laughter Studies*, sight reading practices are translated into a format that limits performer's abilities to listen or infer ahead; a mediated oral tradition in *Guide* enables the qualitative embodiment of the recorded material's specific characteristics; Chen's use of Celibidache's recording in *Adagio* playfully situates and encapsulates successive mediations of Bruckner's score,[15] etc. In addition, digital manipulation of recorded sound affords a mutable score-object as in Stanic's *Open Air Bach*.

[12] See Figure 2 for further examples of rehearsed audio scores and associated affordances

[13] Documentation: https://vimeo.com/194820531 [16]

[14] See Bell [1] [3] and Bhagwati [2] for further discussion of click tracks as a form of audio score

[15] cf. Michael Baldwin's *BUZZED* (2015) for solo French horn where several recordings from the collaborative process are edited and layered in the final audio score https://michaelbaldwin.online/buzzed/ [19]

Category	***Affordances / Limitations***	***Pieces, Situations***
Reactive Emphasis	Real-time improvisatory response to sound/performers cannot hear or infer ahead	Bryars, *1, 2, 1-2-3-4* (1970) • instrumentalists imitate their instruments' parts in recordings of popular songs [4] [21] [22] Brown, *Tomorrow, When I Grow Up* (2017) • unpredictability of audio score stimulates a performed anxiety [23] Cassidy, *I, purples, spat blood, laugh of beautiful lips* (2006) • singer matches the pitch of an undulating sine tone generated by a computer in real time [1, pp.52-53] [24] d'Heudieres, *Laughter Studies* (2015-) • spontaneous free vocal imitation and description of an unpredictable collage of processed recorded and generated sound (see also *FOR___ ON___* (2015)) [4] [6] Palme, *Cantu Foliato* (2012) • choir spontaneously responds to and imitates pre-recorded voices [1, p.51]
	Instructive/performative prompts	Bhagwati, various pieces • event and instruction cues for executing sounds or choreographies, as well as prompts for improvisation, and imitation of samples or styles [2] Castonguay, *Le Souffleur* (2009) • performative prompts sent through Max/MSP [2, p. 26] [25] Chen, *Adagio* (2009) • Bruckner recording serves as an affective prompt [16] [18] Improv Everywhere, *The Mp3 Experiments* (2004-) • a large crowd of participants receive synchronized instructions for physical actions through headphones in a public space [26] Mason, *felt | ebb | thus | brink | here | array | telling* (2004) • tempo and pitch cues, as well as instructions referencing symbolic visual notation absent in performance [1, p.49]
	Mutable score-object	Baldwin, *BUZZED* (2015) • audio re-arranged in DAW during collaborative process [19] Brown, *Tomorrow, When I Grow Up* (2017) • audio re-arranged in DAW before each performance [23] Castonguay, *Le Souffleur* (2009) • composer controls the output to each pair of headphones in performance [2, p. 26] [25] Schimana, *Virus #1.0 - #1.7* (2011-17) • resonant body provides live-generated sounds which influence the performance [2, p. 25] [27] Stanic, *Open Air Bach* (2005, rev. 2013) • live processing of existing recording tied to performer's movements [8]
	Real-time ambient sound as audio score	Kubish, *Electrical Walks* (2004-) • participants engage with electromagnetic signals in the environment via the use of custom headphones [28] Lucier, *Vespers* (1968) • blindfolded performers echolocate using the reverberant properties of the space as a guide [2, p.25] [29] Oliveros, *Sonic Meditations* (1974) • participants reinforce and contribute to environmental sounds [30]

Category	***Affordances / Limitations***	***Pieces, Situations***
Rehearsed Emphasis	High-fidelity reproduction	Bauckholt, *Zugvögel* (2011-12) • recordings of bird calls allow performers to execute a high degree of timbral detail in their imitations [10] [12] Bell, various pieces • cues and click tracks aid performance of microtonal tunings and complex tempi, sometimes in non-standard spatial configurations [1] [3] Hadju, *Der Sprung*, Intermezzo (1999) • performers use audio score as an aid for microtonal tuning [31] Lortie, *Incorporate* (2018) • audio score expedites the communication of complex timbral detail [32] Mazulis, *Ajapajapam* (2002) • performers use audio score as an aid for microtonal tuning [1, p.49]
	Corruption of memory	Bauckholt, *Zugvögel* (2011-12) • sound recordings memorized, but not played back during performance; audio score supplemented with a conventionally-notated one [10] [12] Lortie, *Incorporate* (2018) • sound recordings memorized, but not played back during performance; cues on a computer monitor instruct when these are to be performed [32] Miller, *Guide* (2013) • sound recordings memorized, but not played back during performance; supplemented with a quasi-nuematic score using open graphic notation to encourage flexibility [13] [14]
	Embedding and situating personal, musical and collaborative histories	Applebaum, *Clicktrack* (2015) • each performer creates their own audio score by reciting and recording a poem, which subsequently acts as their individual click track [33] [34] Baldwin, *BUZZED* (2015) • indexing of physical movements linked to sounds; the audio score layers successive recordings from the collaborative process between performer and composer [19] Chen, *Adagio* (2009) • successive mediations of specific instantiations of the slow movement from Bruckner's 7th symphony [16] [18] Miller, *Guide* (2013) • audio score enacts a mediated oral tradition: performers embody the characteristics of a specific vocal model chosen by the composer [13] [14] Walshe, *1984 IT'S O.K.* (2015) • audio and visual reference material for specific performance attitudes/identities to be embodied [35]

Figure 2. A summary of the principle affordances and limitations of the pieces cited in this paper, Bell [1], Bhagwati [2], and d'Heudieres [4]. We have also included additional pieces known to us at the time of writing, which conform to our definition of an audio score (see footnote 1). For those audio scores used in performance, many employ headphones or ear pieces as the means of transmission (with some exceptions e.g. Brown, Lucier, Oliveros, Schimana, and Stanic).

6. FURTHER AVENUES

Finally, our survey of the affordances enacted in recent audio scores point towards further avenues of exploration for creative practice. Chen's piece is just one of many possible hybrid approaches to the relations between audio score and performer; cross-modal interactions and mappings could be extended further;[16] within a single piece, certain performers might engage with visual scores and others with audio scores; varying degrees of prescription or flexibility in the interpretation of audio scores might be explored; the audio score could be at times mutable via live processing and at others largely fixed; the score itself could be revealed as an object of aesthetic interest over time—diffused over speakers in different spatial configurations, including headphones; evidently, these diverse approaches and others might be applied asymmetrically across individual members of a given ensemble or at different times within one piece. In all of these cases—speculative as well as realized—representations of information and instructions in sound afford the possibility of *fundamentally recomposing the interpretive encounter between performer and score*.

Acknowledgments

The authors would like to thank Dr. Erik Ulman for his comments on early drafts, the featured composers for providing access to scores and recordings, David Kerr for the performance still, and the Department of Music and Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University for sponsoring this research. Early versions of this paper were presented in 2018 at Harvard University, the University of Tennessee, Knoxville, and the University of North Georgia.

REFERENCES

[1] J. Bell, *Audio-scores, a resource for composition and computer-aided performance.* Doctoral Thesis. Guildhall School of Music & Drama, 2016.

[2] S. Bhagwati, "Elaborate audio scores: concepts, affordances and tools," in *Proceedings of the 4th International Conference on Technologies for Music Notation and Representation (TENOR '18),* Montréal, 2018 pp. 24-32.

[3] J. Bell and B. Matuszewski, "SMARTVOX – a web-based distributed media player as notation tool for choral practices," in *Proceedings of the 3rd International Conference on Technologies for Music Notation and Representation (TENOR '17),* 2017, Coruña, Spain.

[4] L. d'Heudieres, "Colourful interactions: composers, theatrical scores, and music as performance," in *Collaborative and distributed processes in contemporary music-making*, L. Redhead and R. Glover, Eds. UK: Cambridge Scholars, 2018, pp. 12-26.

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

[6] L. d'Heudieres. "Laughter Studies 1-3 (2015-16)" [Online]. Available: http://www.louisdheuderes.com/works.html

[7] N. Cook, *Beyond the score: music as performance*. New York: Oxford University Press, 2015.

[8] chyvane. "Open Air Bach (2005) by Lara Stanic," *YouTube*, Feb. 6, 2017. [Video file]. Available: https://www.youtube.com/watch?v=mdbK0S_PvM4

[9] C. van Eck, *Between air and electricity: microphones and loudspeakers as musical instruments*. New York: Bloomsbury Academic, 2017.

[10] Score Follower. "Carola Bauckholt — Zugvögel [w/ score]," *YouTube*, Dec. 1, 2013. [Video file]. Available: https://www.youtube.com/watch?v=KEAHxL-NyVxw

[11] J. O'Callaghan. "Mediated mimesis: Transcription as processing," in *Proceedings of the Electroacoustic Music Studies Network Conference: Meaning and Meaningfulness in Electroacoustic Music*. Stockholm., 2012 [Online]. Available: http://www.ems-network.org/IMG/pdf_EMS12_ocallaghan.pdf

[12] C. Bauckholt, "Zugvögel" for reed quintet. Freiburg, Germany: Thuermchen Verlag, 2011-12.

[13] C. Miller. "Guide - EXAUDI Vocal Ensemble (2013)," *SoundCloud*, Oct. 22, 2013. [Audio file]. Available: https://soundcloud.com/cassandra-miller-composer/guide-exaudi

[14] C. Miller. "Guide (2013)" [Online]. Available: https://cassandramiller.wordpress.com/2013/10/22/guide/

[15] E. Fischer-Lichte and I. J. Saskya, *The transformative power of performance: a new aesthetics*. New York: Routledge, 2008.

[16] C. Chen. "Adagio," *Vimeo*, December 8, 2016. [Video file]. Available: https://vimeo.com/194820531

[17] M. Baldwin. "Better Know A Weisslich: Carolyn Chen's Adagio" [Online]. Available: https://weisslich.com/2016/04/07/better-know-a-weisslich-carolyn-chen/

[18] C. Chen. "Adagio (2009)" [Online]. Available: https://walkingmango.files.wordpress.com/2014/08/adagio.pdf

[19] M. Baldwin. "BUZZED (2015)" [Online]. Available: https://michaelbaldwin.online/buzzed/

[16] cf. video scores e.g. Celeste Oram's *XEROX ROCK* (2015) http://celesteoram.com/XEROX-ROCK-2015-a-video-score [20]

[20] C. Oram. “XEROX ROCK (2015)” [Online]. Available: http://celesteoram.com/XEROX-ROCK-2015-a-video-score

[21] Cnock Cnock. “Gavin Bryars - 1, 2, 1-2-3-4,” *YouTube*, Dec. 19, 2013. [Video file]. Available: https://www.youtube.com/watch?v=ZiOKkB482bU

[22] Mode Records. “Gavin Bryars - The Marvellous Aphorisms of Gavin Bryars: The Early Years” [Online]. Available: http://www.moderecords.com/catalog/177bryars.html

[23] B. Brown. “Tomorrow, When I Grow Up | Becky Brown” [Online]. Available: http://becky-brown.org/work/tomorrow-when-i-grow-up

[24] A. Cassidy. “I, purples, spat blood, laugh of beautiful lips — Aaron Cassidy,” Apr. 21, 2017. [Online]. Available: http://aaroncassidy.com/product/i-purples/

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

[26] Improv Everywhere. “The Mp3 Experiments — Improv Everywhere” [Online]. Available: https://improveverywhere.com/missions/the-mp3-experiments/

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

[28] C. Kubisch. “Electrical Walks” [Online]. Available: http://www.christinakubisch.de/en/works/electrical_walks

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

[30] P. Oliveros, *Sonic Meditations*. USA: Smith Publications, 1974.

[31] G. Hajdu. “Research and Technology in the Opera Der Sprung” [Online]. Available: http://www.georghajdu.de/gh/fileadmin/material/articles/Research_and_Technology2.pdf

[32] Chris Lortie. “Incorporate (2018) - Ensemble Proton Bern,” *YouTube*, Oct. 6, 2018. [Video file]. Available: https://youtu.be/evfLyEUTFUc

[33] M. Applebaum. “Clicktrack,” for twelve players responding to individual clicktracks. Self-published, 2015. http://web.stanford.edu/~aplemk/scores/Clicktrack1of4-ForwardAndPartCovers.pdf

[34] Doug Westholm. “Click Track - Mark Applebaum,” *YouTube*, April 13, 2015. [Video file]. Available: https://www.youtube.com/watch?v=d9vrp8VYidY

[35] WEAREGRUPAT. “1984 IT’S O.K. by Jennifer Walshe,” *YouTube*, Oct. 4, 2016. [Video file]. Available: https://www.youtube.com/watch?v=peUox5Y1_9w

TOWARDS RESPONSIVE SCORING TECHNIQUES FOR NETWORKED MUSIC PERFORMANCES

Rebekah Wilson
STEIM
Amsterdam, The Netherlands
rebekah@steim.nl

ABSTRACT

The latent and unstable nature of networked performances, where the delayed transmission and uncertain, unstable, and compressed reception of transferred information demands scoring conceptualizations that consider the loss of the presence information traditionally expected by musicians when performing together in a shared space and time. The focus of this study is to develop electronic network-aware responsive scoring techniques that consider the primary constraints of networked music performances: i.e., latency, uncertainty, multilocated, and digital. Using machine-learning techniques to investigate and enhance digitally mediated presence technology, scoring possibilities are discussed that promote the experience of performing together while being remote from each other—connected via a public network and subject to latency. This study also looks at compositional and technical approaches to creating responsive scores for networked music performances using analysis of transferred sound as a means to generate and control metadata and symbolic notation.

1. INTRODUCTION

The pianist reaches out and strikes a few notes, and a phrase glitters across the piano. A moment later, from somewhere that cannot be seen, an echo arrives, yet it is not a perfect echo. We can hear clearly another piano—another pianist—out of sight with a slightly different timbre, a slightly different tuning. Our pianist responds to the notes she hears; she plays her own echo back, a slight, un-avoidable transformation. Again, notes arrive from else-where that mirror the sequence—but not exactly, a respon-sive spectral apparition. The pianists iterate notes and phrases that sometimes align and sometimes do not, result-ing in unexpected harmonies and timbres. Separated by distance and connected via a digital network, the pianists engage with each other using a score that responds to their actions; i.e., a performance takes place where synchroni-zation is impossible, but music happens.

With the Internet's extended reach and the development of transmission technologies over the last two decades, music has become reproducible in unimaginable ways, leading to explorations of performing music together over the network. However networked music performance has yet to enter the common-use arena. When connecting remotely, musical participants experience the loss of significant amounts of multisensory information that they rely on when playing music together in composite space. While we can simulate and transfer certain kinds of sensory data, such as high-quality audio and video, outside of dedicated networks, this transfer is subject to latency and high compression rates, which cause the loss of infor-mation and network instability. These factors cause musi-cians to perform in unfamiliar territory where traditional forms of scoring and time-keeping fail. We may instead look towards designing new forms of scores and interac-tion that are well-suited to the technical, latent nature of the networked performance environment, leading to the development of positive conditions with the aim of wider adoption of the medium.

While technical advances in intermachine music networking protocols such as MIDI, Open Sound Control (OSC), and Dante may be now ubiquitous, creating music together over the Internet is largely limited to collabora-tions involving the non-real-time transfer of files and per-forming together is extremely limited in practice. As Nevejan states, "sharing rhythm, without having to make any extra effort, creates an initial mutual sense of trust" [1]. The latency and unreliability of the Internet affects musicians' effortless ability to trust that they are, or even can be, in rhythm with each other. This loss of trust dis-rupts the very act of making music together. However, mu-sicians have a deep capacity to adapt to new technologies. Considering how the Internet is deeply embedded within society, and the "transcendence of time and place has been a human drive for centuries" [2], musicians and audiences will accept networked music as a platform for expression given a solution for the disruption of rhythm.

In this paper, I present an overview of the primary characteristics of networked music performance that must be considered when developing network-aware responsive scoring interfaces specifically designed for performances over the public Internet. I introduce the concept of time-based “musical agents” that act as both score and represen-tation of the performance environment, where musical agents are detected, analyzed, and transformed in real-time by software. My work F Not F (2019) is presented as an ongoing experiment of this concept into practice.

2. PRIMARY CHARACTERISTICS OF NET-WORKED MUSIC PERFORMANCES

The primary characteristics of networked music perfor-mances are defined by technology and the imposition of that technology on participants. The public Internet is a packet-switched network made resilient through its decen-tralized design, where data is broken up into pieces and transmitted along many possible paths to its destination where it is put back together. This design is subject to la-tency and instability: i.e., packets may arrive late or not at all, which leaves the receiving side unable to faithfully re-produce the original. Network latency cannot be removed because it is a property of the natural laws of physics: i.e., information cannot travel faster than the speed of light and latency cannot be reliably predetermined or fixed at a cer-tain value. For example, latency may change throughout the duration of a performance. Consequently, latency and transmission uncertainty are primary unavoidable charac-teristics of networked music performances.

The second primary characteristic of networked music performances is that participants are multilocated co-crea-tors. A networked music performance requires at least two participants, each inhabiting their own distinct acoustic space and influencing a musical result that is experienced only by them and their local audience. The projection of sound from one location to the other creates new acoustic challenges that must be considered from a technical stand-point; e.g., is each musician provided with a microphone that is mixed with as little feedback as possible or are room microphones used carefully to provide a reverberant acoustic effect? How the acoustic performance details are determined often depends on the technical resources avail-able from one venue to the next. The third primary charac-teristic of networked music performances is that presence is digitally mediated: i.e., music is performed over a digital network where the signal is certain to be digitally encoded, compressed, transmitted, and decoded. Sounds, images, and performative relationships are managed purely by the means of this digital transmission and in the process of this unavoidable operation, we afforded the means to further manipulate and modify transmissions towards new timbral and aesthetic experiences.

[1] Sarah Weaver defines “composite space” as performing in the same space at the same time [3], in contrast to “networked space,” where peo-ple are not in the same space, but are performing at the same time.

3. VITAL INFORMATION AND RHYTHM

Traditional music performances occur in composite spaces,1 i.e., spaces where people are together, here and now. In composite spaces, music is synchronized not only through music, but also through visual gestures and glances, a conductor’s nod, or the down-movement of a violinist’s bow. Musicians constantly exchange “relational information,” the “tacit, the personal, experiential, ethical, aesthetic” [4], and respond to that information for the du-ration of their performance. By contrast, networked music is mediated presence: i.e., a partial presence that lacks the full-sensory relational experience of real-life interactions. The performative relationships between two or more peo-ple when communicating over a network are mediated dig-itally; i.e., they cannot sense or exchange presence infor-mation without technology. The technology of transmit-ting information over the network is full of uncertainties; it is latent and unstable. However, it is also capable of transmitting what musicians need to perform together. Nevejan suggests that “when in trouble, one needs good information and good communication; i.e., one needs ‘vi-tal information’” [5], which is information that “supports survival for a specific person in a specific place at a spe-cific time” [5].

Transferring vital information in networked music per-formances is crucial in facilitating synchronization be-tween performers. In contrast to the traditional perfor-mance setting where ensemble musicians can rely on low latency audio and visual communication, when we connect remotely, we anticipate that the information that informs us about another’s presence will be limited. For example, when we video chat with a friend who is on a limited net-work, we anticipate glitches and network drop-outs that in-terrupt a smooth conversation and we accommodate ac-cordingly by perhaps using text chat or rescheduling the call for a later time. These interruptions result in synchro-nization difficulties where we easily misunderstand each other’s rhythmic intentions. In a musical performance, par-ticipants must consider what conditions can be controlled that limit these disruptions and still achieve a sense of shared time. Sharing time and rhythm is an important fac-tor because it creates performance intimacy between re-mote participants [6].

4. NETWORK-AWARE RESPONSIVE SCORING

A network-aware responsive scoring system is one that is in tune with both the conditions of the network and the mu-sical performance at the present moment in time. Respon-sive scores are graphical scores that in some way can be modified in real time to adapt to conditions, e.g., graphical scores that are connected to environmental input or that update a notation display as time passes. Responsive scores present information to the musician that reflect how musical relationships change over time in ways that static scores cannot, offering “ways to deal with time from a realistic standpoint" [7]. Responsive scores also afford mu-sicians to engage in the moment to change their actions and create a meaningful musical experience, where there "might be an urgency and a will to do what is required to effectuate that change" [8]. Combining responsive scores with techniques, such as telemetrics, machine learning, and statistics, responsive scores become network-aware: i.e., they can now adapt to latency and bandwidth changes as they occur, as well as responding to musical content from remote musicians as it is arriving over the network. Thus, the network-aware score can respond to and provide vital information about the performance environment to the musicians by addressing the primary constraints. A network-aware score must account for latency, it must acknowledge that no single location is the primary author or focus of a performance, and it must interface with digital network technology.

4.1 Existing network-aware interfaces

Network-aware and -responsive interfaces have been developed since networks have been in existence. For an excellent overview of historical real-time music systems for networked music, see Georg Hadju's "Automatic Composition and Notation in Network Music Environments" [9]. To date, network-aware scores have been primarily focused on transmitting and processing metadata or symbolic information or musical instructions for the performance itself due to limitations in compression technology and bandwidth. Hadju's Quintet.net transmits events between locations over the low-latency User Datagram Protocol (UDP), presenting musicians an interactive score of "certain notes or phrases to be played within time brackets" [10]. Extending this concept is Whalley's Graphic Networked Music Interactive Scoring System, where distributed clients communicate over OSC and map various musical parameters for interactive performances, with the common purpose of network-aware scores to "address practical timing problems in coordinating network-distributed participants following a score" [11]. Combining various approaches, the graphic and animated notation Decibel ScorePlayer system connects over a wide area network to transfer score instructions and permutations in real-time between remote participants [12].

With more recent developments and freely available high-quality audio transmission due to the growing ubiquity of broadband networks and concurrent software developments that offer high-quality, low-latency audio, we can now also analyze and modify the transmitted audio content itself, satisfying the network-aware requirement to consider the digitization and transmission of sound as an opportunity to seek new aesthetic approaches. Working directly with the transmitted audio, Ethan Cayko's toporhythmic research addresses "trans-chronotopic metricity" [13] by realigning decoded remote audio streams according to a telemetric-derived compensatory delay to achieve a fixed latency. This technical manipulation allows musicians to reliably perform precomposed rhythmic patterns while acknowledging the multiplicity of the distributed performance environment where each location experiences time—and therefore rhythm—differently.

4.2 Integrating network characteristics

Musicians are highly intimate with latency: i.e., the time it takes for a generated sound to reach the ears depends on the size of the performance environment and the distance from each other within that environment when performing in composite space where the response time is expected to be within some milliseconds. With the added network transmission time; however, response time is further in-creased. As Chris Chafe notes. "response time is variable depending on [the musician's] attention and what they hear, but it's way longer than the network delay" (private interview with the author, 2016). When a musician's re-sponse time is disrupted by network transmission, this la-tency extends musical opportunities with a unique dimen-sion because it affords further machine-processing latency that would otherwise be considered a disadvantage in a performance situation.

Network transmissions not only create response time de-lays and the inability to synchronize, but also cause con-gestion in networks, leading to unstable connections. Un-like in composite space, we cannot reliably repeat a musi-cal performance when connected over a network because of these unforeseen instabilities that may arise at any time. The musical score for remote participants must be de-signed for uncertainty and therefore be considered a guide to be followed rather than a score that can be reliably re-produced from one performance to the next given similar conditions.

4.3 Types of music representation

Hadju and Didkovsky state that in "current [networked music performance] environments one can typically differentiate three types of music representation being transmitted over the network" [14]. They go on to list the following:

1. Low-level audio, which can either be multichannel uncompressed or compressed (which increases latency, but allows audio transmission over consumer lines).
2. Mid-level performance data, which include note event or continuous-control messages generated by MIDI or alternate controllers.
3. High-level score data for symbolic representation of music.

I propose a fourth type of representation, i.e., the analysis and musical representation of digitally captured perfor-mances in audio format. With the aid of music infor-mation-retrieval methods and machine-learning methods, performative behavior patterns or "musical agents" can be detected in near-real time, processed locally, and transmit-ted over the network.

5. MUSICAL AGENTS

Hatten states that a “semiotic attribution of agency typically involves a sentient being that may set into action var-ious tools” [15]. Hatten introduces the idea of virtual agency, where a musical gesture or idea implies a creative force that generated that gesture or idea. The gestures then become agents of intent. When performing over a remote network where presence is mediated by technology, thesource of agency is hidden from us, yet we assume some-one, or something, created the sound we are hearing or at least triggered a process that set that sound into motion. Given we are communicating via technical means, we have data to detect and measure musical agents, and apply transformations and return them back to the remote space. By collecting and extracting these data, we can process and analyze these agents and expose new methods of operating within the constraints of the performance environment. Where natural human senses cannot detect the source of an action, technology can help us navigate this unknown territory through the transmission and generation of vital information and create new performance perspectives.

5.1 Detecting and manipulating musical agents

Musical agents serve to illuminate and respond to the primary characteristics of networked music performance and provide a channel for the transmission of vital information. This is achieved by classifying a musical gesture for each agent that can be documented, performed, modified, and transmitted. For example, the agent might be a series of pitch sequences in combination with certain loudness en-velopes and timbral fingerprints. Given a set of data, dis-tinct patterns can be stored and later recognized with the assistance of machine-learning applications. By detecting the transmission of the pattern, we know whether the agent has been transformed in some way, either by the musician or the network performance conditions.

5.1.1 Agent classification

Classifying a musical agent requires training on a certain pattern and consequently detecting that pattern using feature detection. Once the composer has decided which musical features [16] she wishes to detect, a convolutional neural network, which is “a set of filters that are trained to extract the most relevant features for detection from the received signal” [17] can be trained and deployed to ex-tract these features [18]. Where the agent is comprised of multiple components, several passes of sequence detection permits classification as a whole.

Machines are very good at classifying if they have been provided with sufficient and suitable training data, and the classification challenge matches the training data to an ad-equate degree. Deciding how to encode and train sound data so that it can be classified becomes an a~~esthetic deci-sion, as does choos~~ing the degree of accuracy and accepta-ble processing latency during a performance. Classification also requires significant attention to the data-capture method: e.g., when the input data changes, the result also changes. For music classification, this means that we must clearly define the classes and train as many different kinds of input as possible that reflect the classes under different conditions, i.e., with a variety of recording environments and instruments. As classification and pattern detection/ analyses improve, more complex decision-making tasks can take place. For now, the processing latency, classification errors, and statistical uncertainties due to insufficient data and processing time must be embraced when designing machine-learning score integrations.

5.1.2 Agent comparison

Once a musical agent has been detected, we can obtain fur-ther detail through comparison with ideal, statistical, and historical agents. This creates a layer of comprehension: i.e., not only have we detected agents, but we also have coherence though means and variance. We might also compare the real-time transmitted model with statistical analysis of randomness generated by a computer model to generate yet more layers of data. For example, detecting vertical note alignment patterns through a loudness-based chromagram [18] permits the composer to input a series of chromagrams to be matched with more nuance than simply detecting a series of pitches. With the additional factor of loudness, a variety of variation is allowed for in the com-position.

5.2 Data collection

Networked music performances inherently contain multiple forms of information that expose its past and current states, including those of digitization, encoding, transmission, and decoding. The aim of data collection is strongly task-oriented, where data is not inferred through analysis, but retrieved from existing sources of measured information. Data sources in networked music performances in-clude telemetric data from the network itself, machine data, and metadata. Data collected from these sources do not tend to require additional processing: i.e., data in its raw format tell us the state of the system at the time we requested it.

5.2.1 Telemetric data

Telemetric data are the information automatically derived from the information generated by machine protocols during the transmission process and are designed for remote monitoring of equipment and services, which allows the calculation of statistical parameters (usually means and variance). Real-time protocol telemetry tells us detailed in-formation about the transmission, including latency or the time it takes for the data to be transmitted between loca-tions, the average rate of packet loss, the rate of change or “jitter” in latency, the number of packets discarded or re-paired, and a number of other useful measurements, such as the cumulative number of packets lost. Using this information, we can keep the participants

[2] Quality of experience (QoS) is a set of network performance metrics (see RFC 6390, https://tools.ietf.org/html/rfc6390).

informed of the state of the performance environment by providing quality of experience,2 and we can use the data as variables in musical algorithms and processes, such as is demonstrated in section 8. Figure 1 shows a real-time matrix of extracted data presented in a format that allows participants to absorb the present status of the system over the last 60 seconds at a glance.

Property	-60s	-50s	-40s	-30s	-20s	-10s	NOW
currentDelayMs	10	10	11	10	10	11	9
jitterBufferMs	4	5	5	4	4	5	4
jitterReceived	19	20	23	18	17	27	20
RTT	19	19	19	19	19	18	19
packetsLost	29	29	29	29	29	29	29
packetsSent	6330	6380	6429	6480	6530	6581	6682
bytesSent	162138	163388	164613	165888	167138	168434	169754
bytesReceived	232019	233381	234743	236105	237267	238829	240191
requestsSent	202	203	204	205	206	207	208
audioInputLevel	12	5	12	9	30	91	29
audioOutputLevel	80	67	57	67	9	23	25

Figure 1. Historical real-time output statistics of a networked music performance.

5.2.2 Music information retrieval

Music information retrieval primarily takes one of four forms [19]: images, such as a score; symbolic, such as MIDI; metadata, such as the instrumentation or knowledge about the performance and its environment; and digital audio. In networked music, we might classify telemetry as metadata. Music information retrieval permits us to make decisions specifically pertaining to music and performance: i.e., a score provides instructions on how and when to emit sound. MIDI and similar note-event systems tells us what electronic signals have been emitted, and information about digital audio gives us performance cues and markers. Such information is highly useful in determining context, particularly when musicians are remote from one another.

5.3 Data extraction

In contrast to data collection, data extraction is a creative, analytic process where, instead of gathering readily exposed information, meaningful information is inferred using algorithms. The extraction itself becomes part of the composer's process as she identifies and implements suitable algorithms, selects which data streams to input, and determines how to present the output to the data processors. Two immediate methods in networked music performances are those of processing the audio signal itself via digital signal processing (DSP) and extracting information about the transmission through a time-geography analysis.

5.3.1 Digital signal processing

One of the methods for extracting data from sound is DSP. As opposed to metadata collection, DSP is an algorithmic process that demands consideration of a variety of factors; therefore, it is a creative practice. DSP techniques can extract vertical and horizontal spectral and intensity information [20]. From these, we can deduce musical descriptions from the sound, such as invariance in pitch, loudness, duration, and spatial position. Using machine-learning techniques, we can extract musical features, such as beat tracking, chromagrams, and instrument detection and separation [21]. With this information at hand, the composer can implement a system that, e.g., listens to pitch or melodic content and responds accordingly, or can synchronize events according to note onsets.

5.4 Time-geography approach

Time and space are intertwined in networked music performances. Time-geography [22] is an investigation into spatiotemporal processes that tell us about "potential encounters between agents" [23] and how those agents are constrained by space and time: e.g., by capability, coupling, and authority [22], where capability is the constraints dictated by the system in which the agent exists, coupling is being together in a place and time, and authority tells us about the forces that the agent is subject to. With these topologies, we expose the interconnected nature of the performers with the musical environment including the transmission network, where time-geography's "two main tenets are that time and space are seen as resources and that the constraints which operate on human beings particularly in the physical environment, are the primary dictates of human experience" [24]. A time-geography approach detects how agents are moving through time and space, and "distinguishes between fixed and flexible activities based on their degree of pliability in space and time" [25]. By combining telemetric analysis with musical feature analysis, time-geography techniques can tell us about under what limitations the system is operating: e.g., if a pattern experiences deviation due to congestion occurring over the network, which creates ripple effects for each agent's response time and thereby influences the musicians' ability to perform.

5.5 Pattern recognition

Where situations can be described as abstract logical problems, machine-learning algorithms are better than humans at detecting patterns. Automated pattern recognition techniques can use training data to detect relationships to existing known sequences, or they use the sequence itself to describe relationships within that sequence as the time series occurs. Pattern recognition implies having a pattern to recognize. For example, in networked music performances, we would simply train a recurrent neural network, such as long short-term memory or bidirectional recurrent neural networks [17] depending on our requirements, with initial telemetric data giving a base reading of the environment before a performance begins. Once the neural network has been trained, we will output data that can be folded back into the score.

5.6 Sequence deviation

Statistical reporting of deviations from a sequence, where anomalies and detected and measured, can be applied to both high-level agent movements in time, and within the agent itself when investigation change in pitch or harmonic sequences. In networked music performances, this determines when musical agents are subject to variation, and can be applied in scoring situations to detect both the creative decisions made by musicians and evaluate transmission performance within the network.

5.7 Sequence prediction

To add yet more layers of interpretation to her application, the composer may choose to predict the next events in a given time-series. Where the objective of a predictive model is to estimate unknown variables for two-dimensional sequences, such as pitch prediction, note onsets, envelopes, and network telemetrics, statistical forecasting algorithms are most useful where, at least for short-term forecasting, machine-learning algorithms offer little benefit at the expense of complexity [26].

5.8 Composite models

Integrating multiple algorithms and processes allows the detection and generation of multiple types and layers of sequences, allowing a more-sophisticated nonisomorphic approach to working with digital interfaces. When a massive data cloud can be processed, this exposes information about music as it happens and delineates the spatiotemporal relationships between remote participants. Data analyses give the composer and musicians insight towards both creating an informed performance environment and considering future potential opportunities.

6. *F NOT F*: ANALYSIS OF A NETWORK-AWARE SCORE

F not F (2019) is a continuous research project that puts into practice the principles introduced in this study: i.e., using machine-learning and statistical techniques to develop a responsive network-aware score interface. By analyzing the performance conditions, the score reacts to time-based pattern synchronization where the intent is to react to trajectories and thresholds that must be met before making certain decisions. To achieve this, a series of musical agents are deployed, where musical patterns that are distinguishable by the machine are analyzed.

The title of *F not F* refers to Nevejan's work on presence, where participants share or do not share time, space, and action. They may be "here" or "not here," and they may be participating "now" or "not now." That is, perhaps the musicians are responding to each other in real time or they are performing alongside a recording [5]. The title *F not F* suggests the mathematical concept of a function where input relates to an output.

6.1 Version 1

The first iteration of *F not F* was a custom Python program and a fixed notated score created for a live musical experiment between two pianists located in Ghent, Belgium, and Rotterdam, The Netherlands, which was presented at the Orpheus Institute's March 2019 conference on Simulation and Computer Experimentation in Music and Sound Art. During the performance, the pianist's sound was continuously analyzed by musical feature extraction methods using the Aubio[3] and Librosa[4] libraries. The program ingested real-time data that was sent to a computer vision program built on TensorFlow[5] and Keras[6] with the aim to determine what musical phrase was currently being performed. The program had been trained prior with 16 musical phrases, recorded hundreds of times in a variety of conditions, and translated into chromagram images (Figure 2). The chromagram was selected over the spectrograph or other feature- visualization methods after the testing of all of Librosa's options determined that the chromagram was the most easily recognized image by the computer vision software.

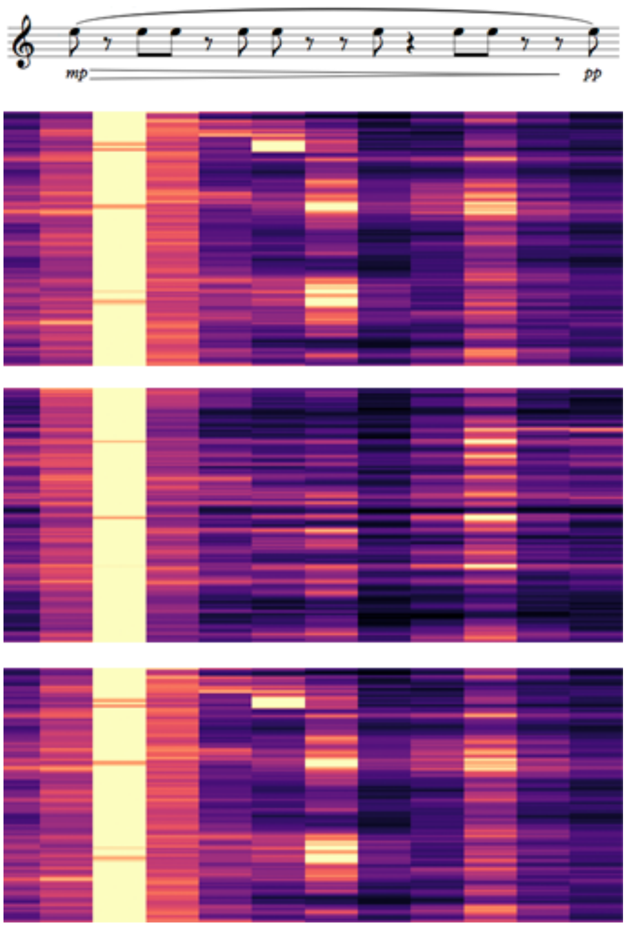


Figure 2. A musical phrase in *F not F* v. 1 and the chromagram representation of three recordings of that phrase.

These images served as training data to the program's machine-learning computer vision system. The performance consisted of the musicians playing a phrase a predetermined number of times, and then moving on to the next phrase. The discrete input of each pianist's mono-summed signal was recorded in real time by the program,

[3] Aubio is a C library and Python interface for the extraction of annotations from audio signals (see https://aubio.org).
[4] Librosa is a Python package for music and audio analysis for the purposes of music information retrieval (see https://librosa.github.io).
[5] TensorFlow is a platform for developing and training machine-learning models (see https://tensorflow.org).
[6] Keras is a high-level neural network Python API (see https://keras.io).

saved in 5-second wav files, one for each channel, and then immediately translated into chromagram images. The image was sent to the machine-learning program where the most-likely match was determined. This process took a few milliseconds. Once the program recognized a phrase, it triggered playback of samples from a bank selected according to the detected phrase. In all, a <6-second delay was experienced for the program to ingest and return the result. This delay informed the compositional structure where the musicians were instructed to repeat the musical patterns several times.

6.2 Version 2

F not F is a necessarily simplified implementation of the technical ideas discussed here; i.e., it is a constant work in development as the sophistication of tools improves. Machine learning is currently prone to error and ambiguity must be anticipated. Consequently, *F not F* is designed to embrace machine ambiguity while being structured overall for development of compositional narratives. Version 2 allows for a greater number of musicians and a variety of instruments to engage with musical agents. While version 1 was a static instructional score, a real-time score is generated for version 2, which can be viewed on a tablet connected to a central computer over a local or remote network. Like in the first version, an agent is a predefined microscore containing a distinct musical pattern where the rhythm and pitches can be easily detected by a machine. In version 2, agents become more sophisticated: i.e., a software program listens to each musician independently for the pattern and applies an interactive musical response. Table 1 lists events that are sent to the score once analysis has returned a result.

Analysis result	Action
A pattern has been partially detected	Trigger array of detected pattern string with threshold float value
A complete pattern has been detected	Trigger detected pattern string
The degree of synchronicity between any two data streams	Continuous stream of sync percentage (float array)
The degree of synchronicity between all streams (global synchronicity)	Continuous stream of sync percentage (float)
Telemetric data	Continuous stream of latency, packet loss, etc., to allow for synchronization calculations (e.g., mixed float, integer array)

Table 1. Analysis results and corresponding events

Version 2 of *F not F* is structured as a series of semi-notated precomposed instructions. For each series, musicians are presented with a series of notes. Figure 3 shows three examples of a note series that correspond to sections of a harmonic series, quantized to the semitone.

The musicians are instructed to improvise on the provided note series, playing notes in any order at a specified tempo and dynamic, but with deviating rhythms and timbres of their own choosing. For example, a resulting rhythm might sound something like the two rhythmic patterns shown in Figure 4.

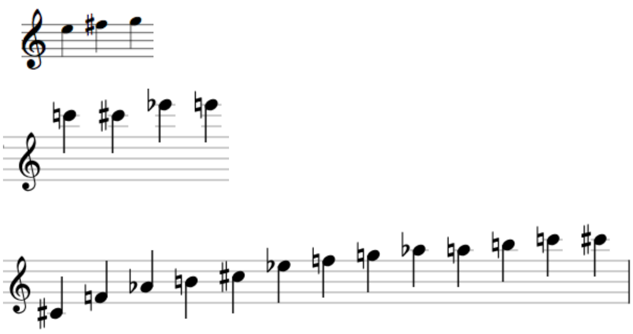


Figure 3. Three examples of a series: one for each musical agent.

Figure 4. Two musician-determined rhythmic pattern examples.

From this improvisation, all instruments in the ensemble combine to create a semi-chaotic timbral and rhythmic effect that continues for an indefinite period of time. The machine listens for a pattern it recognizes, which initially will be a partial pattern. Once this event occurs, the detected pattern becomes the focus: i.e., the "musical agent" that matches that pattern is initialized and the agent pattern begins to emerge in the score. As illustrated in Figure 5, the musicians begin to interleave the agent pattern (top sequence) with the chaos series (highlighted lower sequence). The score encourages the musicians to gradually play the agent pattern gradually more and more by flickering between the two layers, while the machine increases its reading of partial-pattern detection length.

Figure 5. Two examples of an agent pattern that would be animated and interlaced with its corresponding chaos series.

The musicians should now be starting to play the agent pattern more than not, which allows the machine to match complete instances of the agent pattern and calculate the level of horizontal rhythmic synchronicity between musicians, reading telemetric data to calculate for latency. Once synchronicity crosses a determined threshold, the next series is triggered: i.e., the previous agent pattern disappears and a new chaos pattern is shown to the musicians. This cycle can continue as long as the participants and composer wish, by supplying more or fewer patterns until the machine has completed its tasks.

An agent may also have a particular set of parameters that determine its musicality: e.g., it may respond to selected types of data streams or unique triggers specific to that agent to generate correlated effects using granular synthesis or other processing effects. In *F not F*, an agent may be responsive to the current global sync value in real time and react as outlined in Table 2.

NOT IN SYNC (<20%)	WITHIN THRESHOLD (>20% to <80%)	IN SYNC (>90%)
No effects	Clicks, noise bursts, sine tones	Agent sample triggers
	Increasing granular synthesis level	Granular synthesis immersion
		Countdown to next series

Table 2. Triggering electronic effects in *F not F*

When the musicians are not in sync, there are no effects. This decision was made so they can more easily seek synchronicity with each other. As their sync increases, so do the effects and granular synthesis levels. The intent of increasing and decreasing the intensity of electronic effects is to offer the musician the musical choice of whether she wishes to move towards or away from sync, depending on her musical intentions in accordance with the ensemble.

7. FUTURE WORK

The long-term goal of this research study is to develop advanced methods for detecting how musical agents are created and transmitted between networks. Learning how they interact and modified and transformed, whether by musician or machine, guides us towards understanding how participants experience and react to being in-time and in-rhythm with each other when performing remotely over a network. While being a disruptive factor in music, latency can be used for positive gains with machine interaction. The unavoidable delay can be well-utilized, where the transmission wait can be used to analyzed and generate performative interfaces for musical developments. Developing complex network-aware systems demands the intersection of real-time machine protocols with human-level research on mediated presence technology and theory and computer music studies. Collecting data from as many sources as possible at the time of creation and processing that data with algorithmic approaches allows us to create sophisticated sound processing applications and score creation for distance-aware composition structures while building strong musical relationships and transferring the vital information that musicians need when performing remotely. When working with sophisticated graphical interfaces, the transmission latency could be highly beneficial, e.g., when developing complex, immersive works using virtual and augmented reality tools.

Most machine-learning and statistical technologies to date are focused on genre classification or harmonic analysis. A real-time library for detecting and comparing sound features such as timbre, note onset and decay, and agent detection and comparison, e.g., would allow the composer to consider greater score and compositional complexities. As detection and analysis tools become more sophisticated, such as the real-time application of deep learning tools to recognize activities [27], composers will have the ability to model and detect musical features at fine resolution and great speed at their fingertips [28]. In addition, structural and decision-making events can be analyzed and cocreated by musicians and machines. In step with advances in analysis, innovations in digital scoring technologies increase the possibilities of representing this information with highly sophisticated responsive musical agents, which leads to more meaningful encounters between remote musicians and computing systems. As greater amounts of data become available, human perception tends towards perceiving greater meaning; i.e., we attribute a machine's complex recognizable patterns to mechanisms imbued with agency.

Ultimately, there are undiscovered depths of exploration in relation to performance and the machine, and our continued curiosity will certainly uncover many more aesthetic and technical applications. This research works towards the creation of a functional library of performance and scoring tools, while growing the portfolio of networked music performance works by this composer. She hopes to entice others to approach networked music performance as a source for novel and intriguing aesthetic opportunities.

8. REFERENCES

[1] C. Nevejan, Urban reflection: On the design of diverse engagement in the networking city of Amsterdam. [Online] Available: https://nevejan.org

[2] C. Nevejan and F. Brazier, "Granularity in Reciprocity," *AI & Society*, vol. 27, no. 1, 2012, pp. 129–147.

[3] P. Oliveros, S. Weaver, M. Dresser, J. Pitcher, J. Braasch, and C. Chafe, "Telematic Music: Six Perspectives," *Leonardo Music Journal*, vol. 19, no. 1, 2009, pp. 95–96.

[4] S. P. Gill, "Tacit Engagement: Betwixt and In-between," in *Tacit Engagement*, Springer Nature, 2015, pp. 127–160.

[5] C. Nevejan, *Presence and the Design of Trust*. Dissertation, University of Amsterdam, 2007.

[6] J. Croft, "Theses on Liveness," *Organised Sound*, vol. 12, no. 1, 2007, pp. 59–66.

[7] R.R. Smith, Composer Ryan Ross Smith lectures on his 'Study no. 47' at the New Music Gathering. [Online] Available: https://www.youtube.com/watch?v=p66Quh8zDl8

[8] S. Emmerson, M. Baalman, and O. Brandtsegg, "Instrumentality, Perception and Listening in Crossadaptive Performance," in *ICLI 2018 – International Conference on Live Interfaces (Porto)*, June 14–16, 2018, pp. 86-95.

[9] G. Hajdu, "Automatic Composition and Notation in Network Music Environments," *Proceedings of Sound and Music Computing Conference, SMC 2006 (Marseilles)*, 18–20 May, 2006.

[10] Quintet.net. [Online] Available: https://quintetnet.hfmt-hamburg.de/wiki/projects/quintetnet/Quintetnet.html

[11] I. Whalley, "GNMISS: A Scoring System for Internet2 Electroacoustic Music," *Organised Sound*, vol. 19, no. 3, 2014, pp. 244–259.

[12] S. James, B. Carey, X. Fu, and G. Hajdu, "Establishing Connectivity Between the Existing Networked Music Notation Packages Quintet.net, Decibel ScorePlayer and MaxScore," in *Proceedings of the International Conference on Technologies for Music Notation and Representation, TENOR'17 (A Coruña)*, 2017, p. 13.

[13] E. Cayko, *Rhythmic Topologies and the Manifold Nature of Network Music Performance*. Dissertation, University of Calgary, 2016.

[14] G. Hajdu and N. Didkovsky, "On the Evolution of Music Notation in Network Music Environments," *Contemporary Music Review*, vol. 28, no. 4–5, 2009, pp. 395–407.

[15] R. S. Hatten, *A Theory of Virtual Agency for Western Art Music*. Indiana University Press, 2018.

[16] G. Peeters, *A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project*. IRCAM, 2004.

[17] N. Farsad and A. Goldsmith, "Neural Network Detection of Data Sequences in Communication Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, 2018, pp. 5663–5678.

[18] T.L. Li, A.B. Chan, and A.H. Chun, "Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network," *Genre*, vol. 10, 2010, pp. 1x1.

[19] J.A. Burgoyne, I. Fujinaga, and J. S. Downie, "Music Information Retrieval," in S. Schreibman, R. Siemens, and J. Unsworth (Eds.), *A New Companion to Digital Humanities*, John Wiley & Sons, Ltd, 2015, pp. 213–228.

[20] M. Muller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal Processing for Music Analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, 2011, pp. 1088–1110.

[21] B. Srebrenik, Machine learning and music classification: A content-based filtering approach. [Online] Available: https://towardsdatascience.com/machine-learning-and-music-classification-a-content-based-filtering-approach-f2c4eb13bade

[22] T. Hägerstraand, "What About People in Regional Science?," *Papers in Regional Science*, vol. 24, no. 1, 1970, pp. 7–24.

[23] S. Winter and Z.-C. Yin, "The Elements of Probabilistic Time Geography," *GeoInformatica*, vol. 15, no. 3, 2011, pp. 417–434.

[24] N. J. Thrift, "An Introduction to Time-Geography," In *Geo Abstracts*. University of East Anglia, 1977.

[25] H. J. Miller, "A Measurement Theory for Time Geography," *Geographical Analysis*, vol. 37, no. 1, 2005, pp. 17–45.

[26] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward," *PLoS One*, vol. 13, no. 3, 2018, p. e0194889.

[27] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep Learning for Sensor-based Activity Recognition: A Survey," *Pattern Recognition Letters*, vol. 119, 2019, pp. 3–11.

[28] R. Fiebrink, "Machine Learning as Meta-Instrument: Human-Machine Partnerships Shaping Expressive Instrumental Creation," in Musical Instruments in the 21st Century, T. Bovermann, A. de Campo, H. Egermann, S.-I. Hardjowirogo, and S. Weinzierl, Eds. Singapore: Springer Singapore, 2017, pp. 137–151.

NETWORKED MUSIC PERFORMANCE IN THE ST. PAULI ELBE TUNNEL

Georg Hajdu
Hochschule für Musik und Theater
Hamburg, Germany
georg.hajdu@hfmt-hamburg.de

Rama Gottfried
Hochschule für Musik und Theater
Hamburg, Germany
rama.gottfried@hfmt-hamburg.de

ABSTRACT

In this paper we present a new distributed score display system currently under development at the Hochschule für Musik und Theater, Hamburg (HfMT). The project was initiated as part of a large-scale live performance in the St. Pauli Elbe Tunnel (sometimes also referred to as Old Elbe Tunnel), for 144 musicians spread out over the 864 meters of its two tubes. We describe here the background of this project and the current status of the technological and musical considerations required to achieve this event.

1. INTRODUCTION

In a presentation on his piece Music for a Wilderness Lake (1979), R. Murray Schafer referred to it as a "music for a place (personal communication)." This music greatly depends on the topology of the environment it is performed in. While cases of topologically informed music (with static and/or moving musicians) have already existed in since the 1500's with renewed interest in the 19th century (in his Memos Charles Ives tells the story of his father experimenting with marching bands walking towards each other and playing different tunes), it became more of a practice during the second part of the 20th century. Examples include Musik für ein Haus (Stockhausen, 1968), Eine Brise / Flüchtige Aktion für 111 Fahrräder (Kagel, 1996), music by Alvin Curran who created music performed in the Sydney harbor or on the river Thames, among other places.

One of the difficulties of such a practice is maintaining synchronicity, as the participants either act on their own or execute scores with little coordination amongst each other due to lack of visual and auditory cues from a central agent / conductor. Such difficulties are aggravated if the location of the performance is a virtual one such as in networked music performance (NMP). Recent developments in digital technologies have leveraged some if not all of these difficulties by providing audiovisual tools. They are either capable of creating a shared space by low-latency streaming or the exchange of control messages. While audio/video streaming has yielded excellent results since the early 2000s (e.g. CCRMA's software solution JackTrip [1]), systems that feature music notation in networked environments have been rare and only become more widespread since about 2010. Quintet.net [2], a networked multimedia performance environment developed by Georg Hajdu is an earlier example of software in which the interaction of musicians is facilitated by a visual notation layer. The Decibel Score Player is another practical solution available as a commercial solution since 2012 [3].

"Situative scoring" is a term coined by Sandeep Bhagwati [4] referring scores that "deliver time- and context-sensitive score information to musicians at the moment when it becomes relevant." According to Bhagwati, "reactive, interactive, locative scores have added new options to situative scoring [lately]." The members of the European Bridges Ensemble (in existence between 2005-2015) were widely focusing on scores, capable of capturing the specificities of a particular performance scenario [5]. Other approaches to live electronic scoring have been covered extensively in the December 2014 special issue of Organised Sound [6].

In this paper we are describing the musical and technological prerequisites for a project dubbed Symphony for a Tunnel which has been realized in Hamburg in May of 2019. The Old Elbe Tunnel is a remarkable landmark in the heart of the Hamburg port. Completed in 1911, it was considered a technological marvel at the time, connecting two neighborhoods below the Elbe river. Featuring two parallel tubes for pedestrians, cyclists and automobiles (each about 430 m in length) which are being carried down / up by sizeable lifts to / from the bottom 24m beneath the surface, it is also an extraordinary place for performances. Its Jugendstil half cylinders form a resonant body in which the sound of a single instrument carries over large distances with relatively little decay[1].

The Stage 2.0 grant within the Innovative Hochschule initiative of the Federal Ministry of Education and Research in Germany (BMBF) has laid the financial foundation for a musical project in the tunnel. The aim is to connect a large number of musicians via a network of connected devices delivering scores on time. We went through a number of scenarios until zooming in on the most practical solution: As the ideal spacing of individual musicians was determined by us to be around 5 to 6 m, it was a most welcome finding that dividing the total length of the tubes by 6 m yielded nearly 144, a highly divisible number carrying technical and compositional meaning. For instance, this

[1] In an experiment featuring a violin playing scales at mezzo forte dynamics, it could be heard at 50m distance as if it was still be played right next to the observer

number allowed us to define identical sub-groups consisting of 12 musicians each or to place 8 access points at regular distances between musicians (see Figure 6). An older, safer idea consisting of tablet computers connected to a wired Ethernet network was abandoned in favor of a Wi-Fi network as it would have required anywhere between 2.5 and 5 km of cables depending on the number of switches involved. Another serendipitous finding was that when Rama Gottfried joined the Stage 2.0 project in 2018, he had already been working on a node.js-based system for the Berlin Ensemble Mosaik and possessed the expertise to take on a project that would also take advantage of Cycling '74's recent effort to integrate node.js into their Max multimedia authoring environment.

2. FOUNDATIONS

A small number of software solutions are capable delivering scores in a networked environment, most notably In-Score [7], bach [8], the Decibel Score Player and MaxScore [9]. MaxScore emerged from the ongoing effort to provide a robust notation layer to the Quintet.net multimedia performance environment and has first been used in 2007 in a performance at the Budapest Kunsthalle. MaxScore went through several iterations and incarnations (e.g. as LiveScore bringing standard music notation to the Ableton Live Digital Audio Workstation). For clarity's sake we will now use the following nomenclature: MaxScore denotes the environment consisting of numerous objects, abstractions and scripts while the name MaxScore object refers to the Max Java object called com.algomusic.max.MaxScore. The MaxScore object is based on the Java Music Specification Language (JMSL), a Java-based music programming language developed by Nick Didkovsky and Phil Burk [10]. In contrast to other Max notation solutions it requires a canvas to which it draws to and receives mousing information from. This "division of labor" affords considerable flexibility as it allows the MaxScore object to render to various targets, such as Max drawing objects (namely lcd, jsui, jit.mgraphics) as well as Scalable Vector Graphics (SVG) and Portable Network Graphics (PNG) files, the latter via Max Jitter matrix export). Drawing commands, specific to the environment they are being executed in, can be defined as "rendered messages" and attached to notes, staves or measures.

In a 2018 TENOR paper, Hajdu and Didkovsky [11] describe how scores generated by MaxScore could be displayed in real-time on iPads and browsers via the Max Mira and MiraWeb systems. This approach relies on the Max fpic object which can be mirrored on handheld devices. However, having to create a PNG of the entire score of each time it changes (amounting to substantial document sizes) was found to significantly slow down the performance of the Mira/MiraWeb system. After testing various approaches, we found that we were able to leverage the JavaScript/HTML 5/SVG features built into modern browser for much faster results.

Our efforts thus led to the development of DRAWSOCKET, a Node.js-based solution allowing on-time delivery of scores that can be scaled and animated without loss of quality (due to the use of vector graphics).

3. DRAWSOCKET

The DRAWSOCKET system consists of a Node.js server running inside a Max abstraction called hfmt.drawsocket (see our companion paper [12]). Using the NodeForMax environment, the server functions to relays messages from a Max patch to client browsers, routed to individually addressable channels based on the browsers' URL. The server uses the URL as an OpenSoundControl (OSC) address prefix, which tags messages with a target destination. When new drawing commands are received by the server, the commands are routed to the client browsers via WebSockets. On receiving new messages from the server, the client parses the data and uses it to produce new display information in the browser, using SVG as the main display format. The system also provides tools for animation of graphic elements, as well as user interaction callbacks, among other features. With this system in place, MaxScore connects through the hfmt.drawsocket Max abstraction to the client browsers, and then renders its score for each remote location based on the OSC address prefix (Figure 4).

4. MAXSCORE

As mentioned in section 2., MaxScore possesses a fair amount of flexibility in terms of rendering to a wide array of targets. The JavaScript object render2Browser.js was created to facilitate the communication between the MaxScore object and the hfmt.drawsocket abstraction. The js object was designed with massive networked music performance in mind. Such performances pose enormous difficulties when distributing large scores with dozens of staves. In performances with Quintet.net, scores containing just a few staves were split into instructions to be reassembled by individual instances of the MaxScore object and rendered locally by the Clients. But doing the same with dozens of instances (potentially destabilizing the environment and introducing unwanted latency), we re- sorted to a different strategy by implementing the concept of multi-client rendering, treating the ensemble of clients like one single canvas. In MaxScore, nearly every rendering message contains indexes referring to the notation object it represents. Thanks to those indexes, render2Browser.js is capable of dynamically reroute a rendering message to targets set by the staffgroups attribute. This attribute can have the following values: score, parts or a list containing indexes (for individual staves), two indexes joined by hyphens (for a staff range) or any number of indexes joined by plusses (for arbitrary collections of staves), such as in this example:

staffgroups 0 1-2 2 0+3

In addition to splitting and routing messages, the object is also capable of respacing staves so that they always appear on top of the page. It does so by querying the MaxScore object during rendering to obtain crucial information about staff spacing and using this information to apply offsets to the y values of each message to be rendered.

Figure 3 shows a sample of rendering messages generated by the MaxScore object. Note that nearly every message is accompanied by indexes (in red) referring to the notation object they represent. The y coordinates (blue numbers) are remapped according to the current staffgroups setting.

Figure 1. A score with a random melody rendered in MaxScore's default layout.

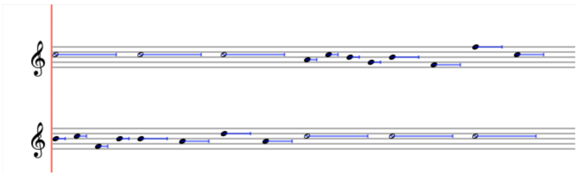


Figure 2. Same score after applying proportional notation. The default hold times indicated by the blue lines are set to 80% of the events' nominal duration.

```
tempoqtrequals 20. 21. 0.5 Measure 0. ...
tr 22. 75.959999 0.5 Staff 0. 0.
staffnumber1 0. 63. 0.5 Staff 0. 0.
timesig4 43. 57. 0.5 Staff 0. 0. ...
StaffLine 0. 0. 4. 0.5 20. 75. 300.660797 75. false ...
frgb 0 0 0
noteheadblack 83.620689 57. 0.5 Note 0. 0. 0. 0.
frgb 0 0 0
no accidental 75.555557 57. 0.5 Note 0. 0. 0. 0.
frgb 0 0 0
stem 76.620689 79. 0.5 Note 0. 0. 0. 0. STEM DOWN
RenderMessage staff 0 0 166. 13. 0.5
rendered Picster−Element[5] 175.3ocUOsnBBCCCzOk6dAg[...]
```

Figure 3. A sample of rendering messages generated by the MaxScore object. Each element is accompanied by an index (such as Note 0. 0. 0. 0.) which can be used to dynamically split the score into parts to be sent to individual targets.

The somewhat cryptical RenderMessage message contains a gzip'ed JSON object which, when deserialized, contains the code for graphical score elements such as line, rectangle, arc or image. The render2Browser.js object is also capable of animating any number of cursors moving across set of measures and staves.

To scroll the entire score horizontally, we created another JavaScript object called maxscore.proportionalNotation.js. It toggles between MaxScore's default score layout and its proportional representation by hiding rests, stems, beams and naturals and indicating the duration of a note by a line extending from a note (see Figures 1 and 2 as well as Figure 7). The length of a measure is calculated by obtaining its tempo and time signature values and taking a setTimeUnit (denoting pixels per second) attribute into consideration. The durational spacing base value of 0.385 has proven to be optimal for spatially representing the delta time between events [14]. The start message will cause a playhead to appear at the position given by the scoreLeftMargin attribute and instruct the browser to scroll the score (Figure 2).

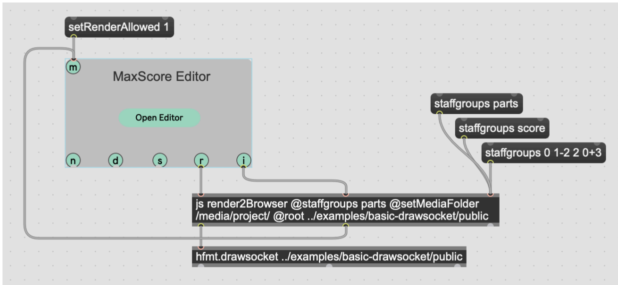


Figure 4. Max patch featuring the render2browser.js object as well as the hfmt.drawsocket abstraction containing the node.js script in charge of communicating with the clients.

5. SETUP IN THE ELBE TUNNEL

The performance system for the Elbtunnel performance consisted of a central server running the MaxScore / DRAWSOCKET framework, and eight Ubiquiti Unifi WIFI access points providing robust and redundant coverage inside the tunnel.

The APs were connected to the network by switches linked via fiberoptic cables to overcome the limitations of current Ethernet cables. Each of the 144 performers had an iPad mounted on a music stand, connected to the central server via a unique URL-OSC routing prefix (e.g. /1, /2, /3, ..., /144).

There were 4 concerts on May 25 and 26, 2019 with pieces written by established composers as well as HfMT multimedia students. The tunnel has been closed for traffic and the audience was allowed to freely move around during the performance.

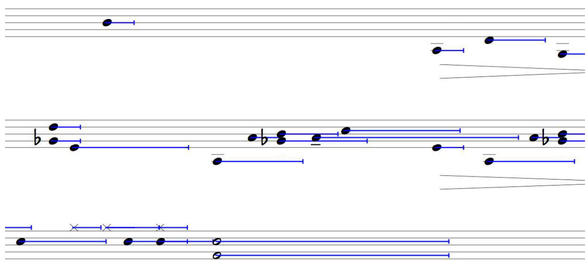


Figure 5. Excerpt from Raindrops Keep Falling (2018) for clarinet, cello, drum set and multimedia.

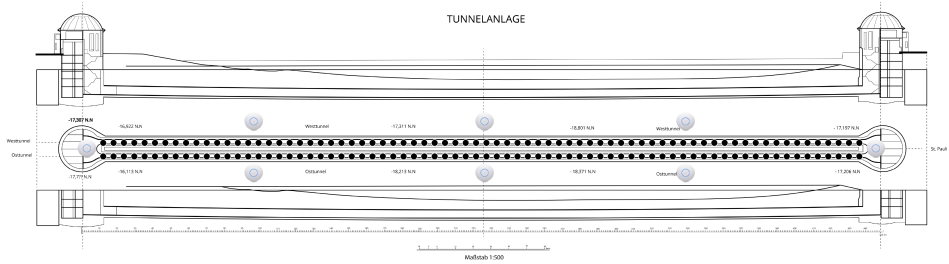


Figure 6. Cross section and top view of the Old Elbe Tunnel. Eight access points will be spaced at regular distances, each providing coverage for 18 players.

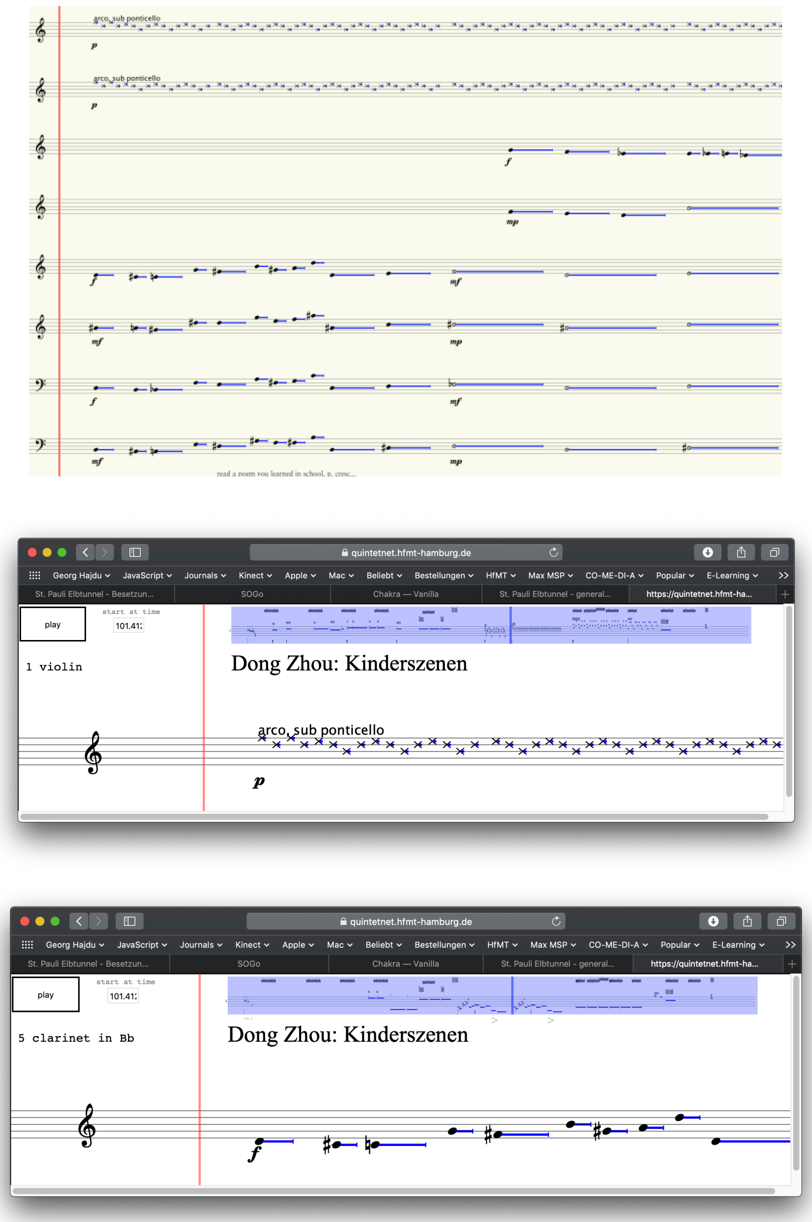


Figure 7. A score by HfMT alumna Dong Zhou showing 8 of 144 staves (top) rendered dynamically in separate browser windows (center and bottom). The staves are split by render2Browser.js. The interface elements developed for practicing at home are being added programmatically in a separate step while interface used in the actual performance differs in that the users can't start the performance by themselves and instead operate a button to indicate their readiness.

6. CASE STUDY

A performance at the December 2018 International Workshop on Computer Music and Audio Technologies (WOCMAT) in Hsinchu, Taiwan posed an excellent opportunity to test drive the software before it was tried in the tunnel. A new piece (Raindrops Keep Falling) by Georg Hajdu for clarinet, cello and percussion with multimedia consists of a transition between various rain samples and a late-1960's hit called Raindrops Keep Fallin' On My Head mediated by a Max Pluggo effect called Raindrops[2]. The piece features 12 different versions of the song found on the Internet. The HfMT graduate student and research assistant James Cheung arranged the songs in such manner that they all share the same tempo structure and key signature, allowing the seamless navigation between those versions. James also created an arrangement of the song for the aforementioned instrumentation to be performed simultaneously with the recording, which was further subject to processing. First, parts of the score were "whited out" by a probabilistic process so that more and more events were allowed to appear paralleling a similar process applied to the audio tracks. The whiting-out was achieved by a JavaScript object called maxscore.whiteout.js capable of applying a "whiteout" gradient to a given section (the name was inspired by Cat Hope's piece The Great White). Second, the score was turned into proportional notation, transmitted to the iPads of the performers via DRAWSOCKET and scrolled in synch with the audio. The system held up to its promise as a computer-based conducting system. The scrolling was fluid and the musicians stayed in tempo despite the tempo fluctuations in the audio track.

7. FUTURE WORK

The St. Pauli Elbe Tunnel project proved to be a successful test bed for situative performances of orchestral dimensions. This project will be further developed according to the following criteria:

1. The situative aspect: The system can be scaled to other outdoors and indoors settings with unique topologies. Once the system has proven its robustness, we will most likely see more real-time and interactive uses.

2. The assistive aspect: Allowing semi-professional and amateur musicians to participate in large-scale events with little prior orchestra experience. A transcription of Ligeti's Atmospheres into proportional notation, for instance, could leverage some of the difficulties reading the complex notation in rehearsals and concerts. The networked notation tool SmartVox by Jonathan Bell and Benjamin Matuszewski [13] has already yielded excellent results working with amateur and student choirs. Cat Hope's opera Speechless is another case where a networked notation tool (Decibel Score Player) has been used in large scale performance [15].

[2] It's also a tongue-in-cheek reference to the usual end-of-year weather pattern in Taiwan.

Acknowledgments

The authors would like to thank James Cheung and the other students of the Multimedia Composition master program at the HfMT for their detailed testing of the system which pushed the development of many new features and design considerations. We would also like to acknowledge the Federal Ministry of Education and Research in Germany (BMBF), for their support of this research through the Innovative Hochschule: Stage 2.0 initiative.

8. REFERENCES

[1] J.-P. Cáceres and C. Chafe, "Jacktrip: Under the hood of an engine for network audio," *Journal of New Music Research,* vol. 39, no. 3, pp. 183–187, 2010.

[2] G. Hajdu, "Quintet. net: An environment for composing and performing music on the internet," *Leonardo*, vol. 38, no. 1, 2005, pp. 23–30.

[3] A. Wyatt, C. Hope, L. Vickery and S. James, "Animated Music Notation on the iPad (Or: Music stands just weren't designed to support laptops)". *Proceedings of the 2013 ICMC Conference, Perth, WA.* 2013*,* pp. 201- 207.

[4] S. Bhagwati, "Vexations of ephemerality: extreme sight-reading in situative scores-for makers, performers, audiences," in *TENOR 2017: International Conference on Technologies for Music Notation and Representation,* 2017, pp. 161–166.

[5] G. Hajdu, K. Niggemann, A. Siska and A. Szigetvári, "Notation in the Context of Quintet.net Projects," in *Contemporary Music Review*, 29, 1, pp. 39 – 53, 2010.

[6] L. Wyse and Whalley, I. Editorial. *Organised Sound, 19*(3), 2014, pp. 213-214.

[7] D. Fober, S. Letz, Y. Orlarey, and F. Bevilacqua, "Programming interactive music scores with INScore," in *Proceedings of the Sound and Music Computing conference*, Stockholm, Sweden, 2013, pp. 185–190.

[8] A. Agostini and D. Ghisi, "A max library for musical notation and computer-aided composition," Computer Music Journal, vol. 39, no. 2, 2015, pp. 11–27.

[9] N. Didkovsky and G. Hajdu, "MaxScore: Music Notation in Max/MSP," in *Proceedings of the International Computer Music Conference*, 2008.

[10] N. Didkovsky and P. L. Burk. "Java Music Specification Language, an introduction and overview," in *Proceedings of the International Computer Music Conference,* 2001.

[11] N. Didkovsky and G. Hajdu, "Maxscore: Recent Developments," in *TENOR 2018: Proceedings of the International Conference on Technologies for*

Music Notation and Representation, 2018, pp. 138-146

[12] R. Gottfried and G. Hajdu, “Drawsocket: A Browser Based System For Networked Score Display,” in *TENOR 2019: Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2019.

[13] J. Bell and B. Matuszewski, “Smartvox–a web-based distributed media player as notation tool for choral practices,” in *TENOR 2017: Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2017, pp. 99–104.

[14] D. Blostein and L. Haken. "Justification of printed music," *Communications of the ACM* 34.3, 1991, pp. 88-99.

[15] 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 *TENOR 2018: Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2018, pp. 193-200.

UNLOCKING THE DECIBEL SCOREPLAYER

Aaron Wyatt
Monash University
Melbourne, Australia
aaron.wyatt@monash.edu

Lindsay Vickery
Edith Cowan University
Perth, Australia
l.vickery@ecu.edu.au

Stuart James
Edith Cowan University
Perth, Australia
s.james@ecu.edu.au

ABSTRACT

This paper discusses recent developments in the Decibel ScorePlayer project, including the introduction of a canvas scoring mode, python ScorePlayer externals, and enhancements to the ScoreCreator application. Firstly, the canvas scoring mode of the Decibel ScorePlayer app allows for other applications, such as Max, to send drawing commands to the ScorePlayer via OSC. Several examples of implementations of generative and animated notation scores are discussed and evaluated. An object model has been developed allowing for the creation of hierarchies of drawn elements. The object model defines a framework of commands that can be used to create and control these objects, and supporting examples describe the way in which scores can be developed to take advantage of this new scoring mode. Secondly, a python *scoreplayer-external* library has been developed, defining two python classes: *scorePlayerExternal* that makes a connection to the iPad, opening a UDP listening socket and letting the iPad know which port to send its replies to, and *scoreObject* which is responsible for creating and drawing objects populated on the canvas display window of the Decibel ScorePlayer. It acts as a wrapper to the raw OSC commands so that programming can be done using object-oriented paradigms. Thirdly, the ScoreCreator, an application developed for Mac OSX for automating the process of making scores for the Decibel ScorePlayer, has been expanded allowing for the defining of a range of score types and functionalities.

1. INTRODUCTION

Upon its public release in 2013 [1], the Decibel Scoreplayer app offered a limited number of score presentation formats. At this early stage of development, it was considered a priority to implement robust networking capabilities which served the demands of live performances involving the synchronisation of multiple devices. The ScorePlayer app continued to be developed over some years, based on an iterative methodology, and tested in rehearsals and performances with Decibel, an Australian new music ensemble. The ScorePlayer's preeminent mode of functionality was the scrolling score format in which an image file could be synchronously moved from left to right over a prescribed period of time on multiple devices. In 2012, while the ScorePlayer was still being used by the Decibel ensemble, a ScoreCreator app was developed for Mac OSX to simplify the creation of scores utilising the scrolling score mode of the ScorePlayer, a user-friendly solution for those less tech savvy. As development of the app continued, further formats and functionalities were added [2, 3, 4, 5, 6, 7], however many of these enhancements were inaccessible to the public: and while some features were periodically made available in public releases of the app, they were not well documented and composers often could only access them by reverse engineering existing scores. This paper documents a range of developments intended both to expand the functionalities of the ScorePlayer app and to open the platform for experimentation to interested parties. These developments include expansion and support for non-linear score formats in the ScoreCreator app, implementation of a canvas scoring mode for the sandboxing of non-standard notations, and a Python library to allow for the control of the ScorePlayer via the command line, in addition to OSC-based visual programming.

2. THE CANVAS SCORING MODE

The Decibel ScorePlayer was initially conceived as a modular platform [1], assigning control of the user interface and networking functions to the main player window while drawing tasks were enacted by rendering modules that could be alternated to allow for the implementation of different score types. New score modalities were implemented, including scrolling in all planar directions, nonlinear, two dimensional [1], rhizomatic [5] and generative [4] scores, but with standalone or very restricted user control of their functionality. While it would perhaps be ideal for the platform to be open allowing users to create and compile their own rendering modules, iOS (like most tablet computing environments) forbids Apps that are distributed via the App Store to permit libraries to be dynamically loaded [8]. We attempted to overcome this limitation by allowing composers to directly control the drawing surface in the canvas scoring mode (publicly released in 2018) from an external application using commands sent via the Open Sound Control (OSC) protocol [10, 2, 7]. This arrangement permits composers to develop, prototype, and even distribute new score paradigms and idiosyncratic realtime score generation without the need for any code to be accepted into the App Store. The public release of canvas mode implements six different types of objects:

- Layers - consist of a simple rectangular region on the canvas which is empty when first created, into which a flat colour or an image can be loaded;
- Scrollers - that allow for a larger image to be scrolled horizontally at a set rate through a fixed viewing window;
- Text - using iPad supported fonts;
- Glyphs - a special case of the text object used as a convenient shortcut to display symbols from the bravura music font (Steinberg Media Technologies GmbH. 2018);
- Staves – creating a five-line staff of a defined size;
- Lines - drawn within the coordinate space of their parent object. (These are the only type of object that cannot be used as a container for other objects.)

An external device can send drawing commands via OSC to any networked canvas score. Each connected score, for example on a set of iPads, has an OSC address starting `/Renderer/Command`, followed by the name of the object to be manipulated, and the command to be sent to that object: for example, to add a layer to the score's initial canvas, the command "`/Renderer/Command/canvas/addLayer`" would be sent. This command is followed by arguments defining the name of the layer to be created, the part for it to be assigned to (or 0 for the layer to be placed on all parts) and then the objects' display coordinate data. A full list of drawing commands can be found at http://www.psi-borg.org/canvas.html.

Any application that can send and receive OSC packets can be used to control the ScorePlayer in canvas scoring mode. Examples of patches written in Max were included in a previous paper [6] and can also function within Ableton's Live using MaxforLive.

A library has also been developed to allow externals to be quickly and easily written in python, and this has been released under the LGPL via the Python Package Index, or PyPI for short [10]. Both of these solutions use Bonjour [11] for service discovery, and are able to find any iPads running the ScorePlayer on the local network, as long as multicast traffic is not blocked. In cases where such traffic is blocked, manual connection is still possible, but an understanding of IP addressing is required.

Canvas scores, like previous ScorePlayer files, consist of a standard zip file with its extension changed to dsz [1] which must be bundled with all of the image resources needed by the score, in jpg or png format, as well as xml files [12] that define the score's metadata and settings for any additional options. This file is then imported into the ScorePlayer via Apple AirDrop, iTunes' file sharing feature or downloaded directly from a web server using either a URL or a QR code that represents a URL. The main xml file that defines the score is named opus.xml [1]. For canvas mode, the <scoretype> tag in this file is set to "Canvas" and duration may be set either to zero or any number of seconds. If zero (or negative), the ScorePlayer only displays the Reset button to the user, and the navigation bar and status bar remain visible. If set to a positive value, then the Play button is also displayed, and pressing it sends the usual /Control/Play command over the network, starts the clock, and hides the navigation bar until the specified time limit is reached. The obvious advantage of this is that it makes a larger drawing surface available, even if a composer has no intention of making use of the timing functions of the ScorePlayer. (The size of the canvas is 1024x768 when in landscape mode, and the status and navigation bar clip reduces this height by 70 pixels). The opus.xml file may also point to an XML preferences file. This file (typically "prefs.xml") is currently used to define the number of available parts that can be drawn to. Like previous ScorePlayer files, parts are accessed by swiping up or down on the screen. In a canvas score this file is also used to define the <clearonreset> setting, which determines whether the canvas is cleared by pressing the Reset button in the ScorePlayer. If undefined this feature is by default set to yes.

2.1 Affordances of Canvas Mode

The canvas scoring mode provides for masking and changes in opacity in addition to a number of forms of object animation. Since new objects are placed in the foreground, existing objects can be modified by placing masks in front of them. The opacity of all objects can be controlled using the `setOpacity` command. Making a parent layer translucent will also affect the opacity of any child layers. Alternatively, the opacity can be animated using the `fade` command, which additionally allows for the duration of the opacity change to be specified. David Kim-Boyle discusses using both masking and opacity generated in Max as compositional tools in his *tunings* (2006) for Cello and Computer [13].

"Flipcard" style animation, of which André Vida's *Vidatone* series is an example [14, 15] can be achieved by repeatedly using a `loadImage` command to load a sequence of evolving images into a layer. All objects can also be moved continuously or discontinuously in any planar direction on the screen. The `move` command sets the position of an object and animates the transition over a period of time specified in seconds by a duration argument, while the `setPosition` command can be used to change the position of an object instantly. The scroller object is most efficient for moving large images. It allows for a larger image to be scrolled horizontally at a set rate through a fixed viewing window. The viewing window itself is defined by the usual position and size coordinates that are passed to the object on creation. A few additional parameters, also passed at creation, are used to define the behaviour of the scroller while animated. The `scrollerWidth` parameter determines the width of the layer to be scrolled through the viewing window, while the `scrollerSpeed` defines the rate at which this occurs in pixels per second. Setting a negative value for `scrollerSpeed` causes the scroller content to move backwards, to the right of screen. These values can be changed at any time, and the scroller can be set in motion or stopped using the start and stop commands.

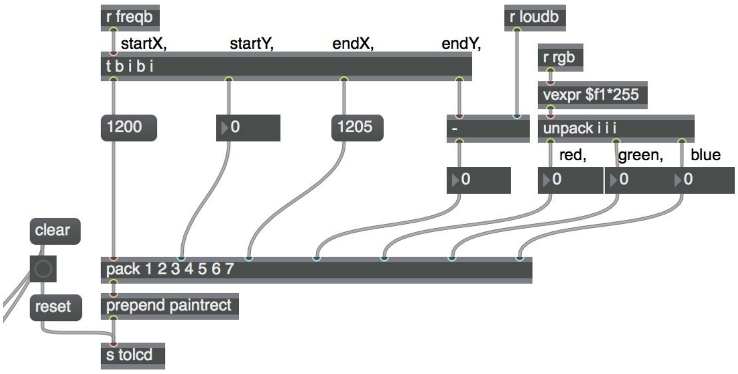


Figure 1. Score generation in the Max-only version of the *Lyrebird Environment Player* using the LCD object.

3. EXAMPLES OF IMPLEMENTATIONS OF WORKS IN CANVAS MODE

Two generative scores by Lindsay Vickery, *Lyrebird Environment Player* (2014) and *The Semantics of Redaction* (2014) originally implemented in Max were adapted to utilise the canvas scoring mode. In these works audio files, of field recordings and speech respectively, are analysed in order to generate a score [4, 5]. In both works the intent was to emphasise the capability of generating a score permitting interaction by performers with recent and locally recorded sonic environments or a current topical speech in "near real-time". The data for generating the score in these works is derived from the `sigmund~` object in *Lyrebird*. Mapping of the audio analysis to score elements is shown in Table 1.

The works presented different challenges for implementation in the canvas scoring mode. In the case *of Lyrebird*, the principal parameter of interest was the network-rate/draw-rate of the OSC connection. In the original Max-only version, *Lyrebird* drew coloured rectangles to an LCD object at an average rate of between 21 and 22 messages per second These `paintrect` messages, as shown in Figure 1, are driven by the output of a `sigmund~` object with an FFT analysis window size of 2048 audio samples.

Updating the score to support the canvas scoring mode on the iPad required translation of the LCD drawing commands. Whilst the original version used a single draw command, the new implementation requires four messages: an `addLayer` command, followed by a `setColour, move,` and `remove` command. As the sequence of these commands is crucial in procedural programming they are sequentially executed by a trigger object (Figure 2). The sending of the remove command is delayed by 12 seconds to allow the move animation for each rectangle to complete. Each rectangle is named according to a variable number '`line%s`' with a replaceable suffix of any number between 0 and 500, as in this case with a draw rate of 21 or 22 rectangles per second, up to 259 separate rectangles might appear at any time over a scroll time of 12 seconds.

Spectral Descriptor	**Graphical specification**
Frequency	rectangle vertical position
Amplitude	rectangle size
Brightness	rectangle hue
Noisiness	rectangle saturation
Bark Scale deviation	rectangle luminance

Table 1. Spectral Descriptor to Graphical specification mapping in *Lyrebird.*

To measure the responsiveness of this scoring mode, the Max application was stress tested whilst overdrive mode was enabled. The timing and quantity of packets sent from Max to the iPad were analysed during these tests. Between 84 and 88 renderer commands were consistently generated every second over a total time of 60 seconds. The networked iPad also comfortably drew at this rate. At this stage it is assumed by this that as a scoring paradigm, this method would prove to be consistently reliable.

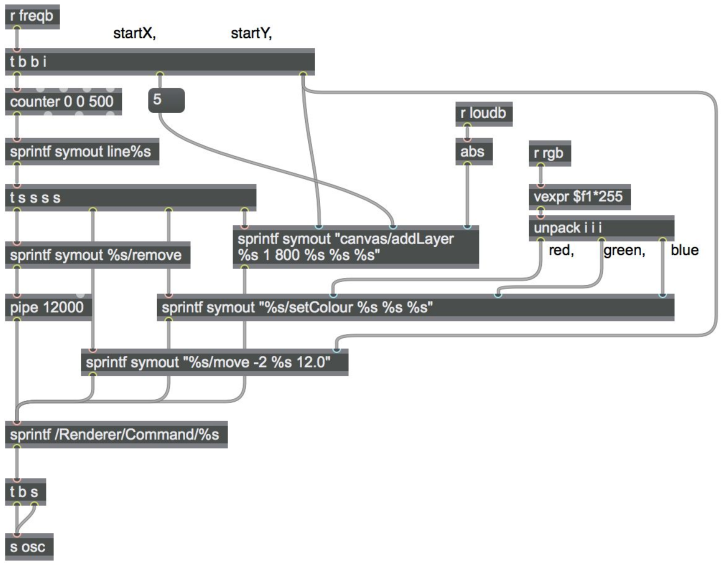


Figure 2. *Lyrebird Environment Player* canvas score mode version.

There are some minor differences between both rendered versions of the score, LCD versus the iPad canvas scoring mode in the ScorePlayer app (Figure 3), however as the sizing of the canvas window in the LCD and on the iPad are known, and both systems use standard RGB color space to define their colors, on the whole the scores translate reasonably closely with the added benefit of the smooth graphical rendering capabilities of the core animation graphics rendering and animation infrastructure available on iOS devices.

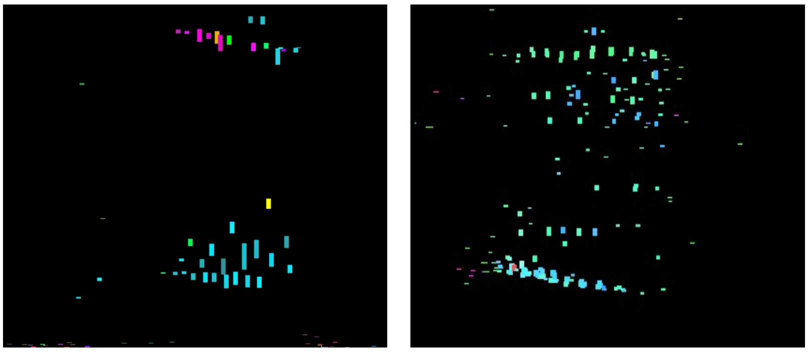


Figure 3. Comparison between *Lyrebird* ScorePlayer canvas score (left) and LCD (right) score.

In the case of *The Semantics of Redaction* five categories of score elements needed to be drawn using separate processes: score setup, noteheads, note stems, graphical symbols and text (indicating the beginning of each section). In each section the audio data is mapped in five varied "modes" (Table 2).

The score setup includes a black line in the centre of the score that functions as a "beam" for all of the generated noteheads/stems and a "playhead" giving the position at which a note is executed. These scored objects are created as part of the initialisation routine as children of a layer object using the commands:

```
/Renderer/Command/canvas/addLayer sor 0 0 0
1024 698
/Renderer/Command/sor/setColour 255 255 255
/Renderer/Command/sor/addLayer beam 0 0 348
1024 6
/Renderer/Command/beam/setColour 0 0 0
/Renderer/Command/sor/addLayer playhead 0 70 0
4 197
/Renderer/Command/playhead/setColour 255 255 0
```

The data for generating the score in these works is derived from the `analyzer~` object in *The Semantics of Redaction*. Mapping of the audio analysis to score elements is shown in Table 3.

The rate of draw commands in *The Semantics of Redaction* are considerably slower than in *Lyrebird*, however, there are a greater variety of draw commands in the work, ranging from rectangular colored noteheads with thin black stems connected to the central beam, to graphical notations, and text. Each of these processes draws on the same process as seen in Figure 2, except the `line%s/setColour` command is substituted for a `%s/loadImage` command when loading png image files. Text includes a number of further variables, and therefore requires more than four commands: `addText`, `setFontSize`, `setColour`, `setFont`, `setText`, `move`, and `remove`.

The iPad copes well with the realtime drawing commands over the network, provided the UDP OSC packets are received. Again because of discrepancy between the iPad screen/LCD dimensions, the score is more horizontally compressed in the Canvas mode implementation (Figure 4).

Spectral Descriptor	Generated graphical specification
Frequency	notehead vertical position and hue
Amplitude	notehead size
Brightness	notehead colour saturation
Noisiness	notehead colour luminance

Table 3. Notehead and graphical symbol drawing behaviour and audio playback behaviour in *The Semantics of Redaction*.

3.1 Flipcard Style Animation

An experiment to reproduce animated notation using "Flipcard" style animation was conducted with Ryan Ross Smith's work *Study No. 55* (2016). The work was a challenging candidate, requiring colour full-screen frames (1920 × 1080px on the iPad) with an average size of 180KB. 500 frames of the 13-minute work (2.6% of the total) were rendered and bundled into a .dsz file of 46MB. A Max patch was made using the canvas `loadImage` command to draw and then remove successive image layers (Figure 5). The optimal removal rate for images was shown to be 1.2 times the draw rate. The draw command rate was tested across a dedicated wireless network between a 2018 Macbook Pro running High Sierra and an

	Note head drawing behaviour				graphical symbols				Audio Playback			
Mode	triggered by detected "attacks"	noteheads extended horizontally	continuous steps drawn between noteheads	interrupted by black rectangles (redactangles)	graphical symbols are superimposed	draws black rectangles (redactangles)	noteheads become increasingly transparent	playback audio normal	playback audio muted	interrupted by redaction "bleeps"	audio superimposed at varying playback speeds.	continually pitch-shifted downwards
Opening/Body	x							x				
Commentary	x	x			x			x				
Interlude	x		x		x				x			
Redactio	x	,		x	x	x				x	x	
Closing	x				x		x					x

Table 2. Notehead and graphical symbol drawing b and audio playback behaviour in *The Semantics of Redaction*.

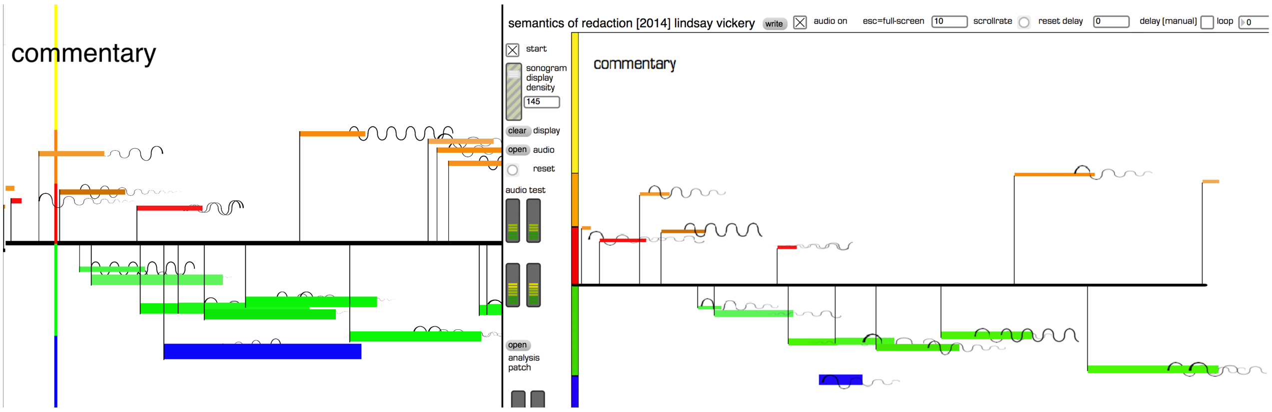


Figure 4. Comparison between *The Semantics of Redaction* ScorePlayer canvas score (left) and Max/LCD (right) score.

iPad Pro running OS12.2. Draw commands were successfully communicated at a rate of 15ms (approximately 66FPS) and demonstrated to be stable at a rate of 41ms (approximately 24FPS) the frame rate of Smith's original video. A second test using an iPad 2 crashed the Scoreplayer application at rates of 4FPS.

Although the results were encouraging, they suggest that flipcard style animation in canvas scoring mode is more suited to the implementation of smaller, shorter frame animations, perhaps in combination with other, less size and CPU intensive, processes. (A dsz bundle combining all 18 thousand frames of *Study No. 55* is estimated to be a more unwieldy 880MB in size).

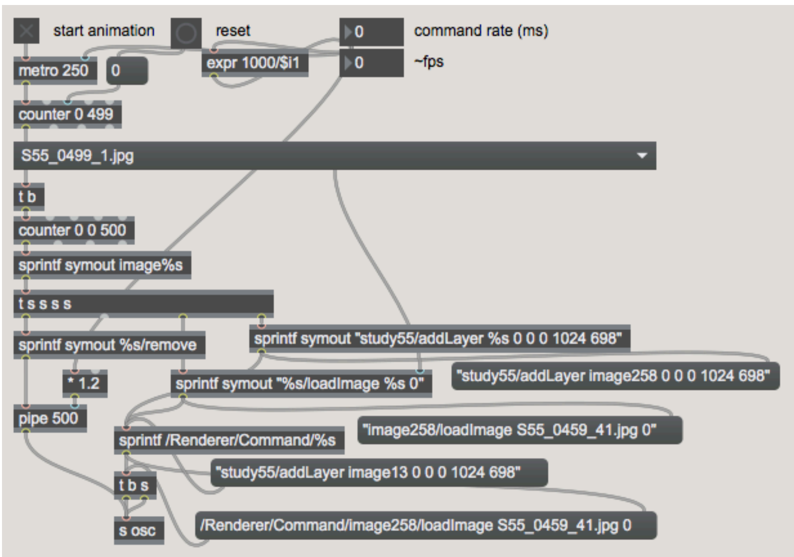


Figure 5. Max patch using the canvas loadImage command to draw and then remove successive image layers for "flipcard" style animation.

4. PYTHON SCOREPLAYER EXTERNALS

First released in 1991 and created by Guido van Rossum, Python is generally recognized as an interpreted, high-level and general-purpose programming language. There are several advantages to using Python as an alternative to Max for driving the Decibel ScorePlayer's canvas mode features on the iPad, particularly due to it being open-source and based on a community development model. As a result, it is freely available on most platforms, and has a convenient distribution mechanism for libraries in the Python Package Index, or PyPI [10]. It also has a plethora of existing libraries, including ones specific to service discovery [17] and OSC [18], which greatly simplified the development of our own scripts. These scripts have since been developed into the *scoreplayer-external* library which is now available on PyPI following its initial release and presentation at the Australasian Computer Music Conference in December 2018. The Python *scoreplayer-external* library defines two python classes: *scorePlayerExternala* and *scoreObject*.

The first class is used to make a connection to the iPad, opening a UDP listening socket and letting the iPad know which port to send its replies to. The second object is designed to represent the objects of music notation that populate the canvas display window. The *scorePlayerExternal* class also encapsulates a Bonjour service browser that can be used to find an iPad to connect to. This design allows for the process of finding and connecting to an iPad to be done in very few lines of code, allowing the composer the freedom to focus almost entirely on drawing commands.

4.1 Connecting to the Decibel ScorePlayer using the scorePlayerExternal class

The following example code demonstrates how a link can be established between a Python script and the ScorePlayer app using the *scorePlayerExternal* class.

```
 #!/usr/bin/env python
 from scoreplayer_external import scorePlayerEx-
ternal
 from threading import Event
```

```
finished = Event()external
= scorePlayerExternal() ex-
ternal.selectServer()canvas
=    external.connect(onCon-
nect)
finished.wait()     exter-
nal.shutdown()
external.shutdown()
```

Once all of the necessary libraries have been imported, a new *scorePlayerExternal* object can be created (and in this case is assigned the name external). Running the `selectServer()` method on this object prints the list of available servers to the console, and prompts the user to either select one or refresh the list to check for any new servers that might have become available. Once the user makes a choice, the `connect(connectionHandler)` method can be called. This sends a registration message to the iPad server, letting it know of our listening port, and on return of a confirmation message it runs the function passed as the `connectionHandler` argument. This function (not shown here) should contain the drawing commands that the composer wants to use to render their score. The connect function also returns an instance of a *scoreObject* that is a reference to our canvas. It is by calling the methods of this returned object that new objects can be added to the score.

4.2 Creating Python-driven scores using the scoreObject class

The *scoreObject* class encapsulates the sending of the raw OSC commands used to manipulate objects on the ScorePlayer canvas, and allows these remote objects to be treated as if they were Python objects. Instances of the *scoreObject* class should not be created directly by the user, but should instead be stored and used when returned from the various methods that add objects to the canvas. As outlined earlier, for example, the initial connect method returns a *scoreObject* that represents the canvas. Other objects of varying types can be added to this by calling the `addLayer`, `addScroller`, `addText`, `addGlyph`, `addStave`, or `addLine` methods of the canvas. This is demonstrated in the following code example (which assumes that the initial canvas object has been stored to the variable canvas):

```
rdef onConnect():
canvas.clear()
stave = canvas.addLayer('stave', 0, 0, 0,
1000, 800)
stave.loadImage('Stave.png', 1)
score = canvas.addScroller('score', 0, 90, 0,
1000, 800, 0, 60) score.loadImage('Score.png',
1)
score.start()
global finished
finished = True
```

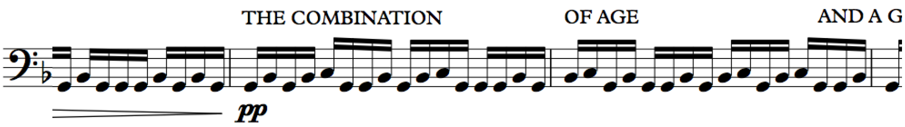


Figure 6. A layer object on the canvas that contains a sublayer.

The addLayer method is used first to add a layer named 'stave' to the canvas. A separate scroller layer is added named 'score', and the script then proceeds to scroll the score. (Figure 6).

There are some methods that are common to all objects. `setColour` can be used to change the colour of any object. (Whether this command applies to the foreground or background colour is dependent on the type of object.) All objects apart from the canvas can also have their opacity set, and all bar the canvas or line objects can be moved within the coordinate space of their parent object, either instantly or animated over a specified time frame using the `move` command. Additionally, any object (except for lines) can be used to hold other objects, allowing for the creation of complex hierarchies of objects that can be manipulated as one with relatively few commands. Score objects also have commands that are specific to their type listed in the Figure 7 tree diagram.

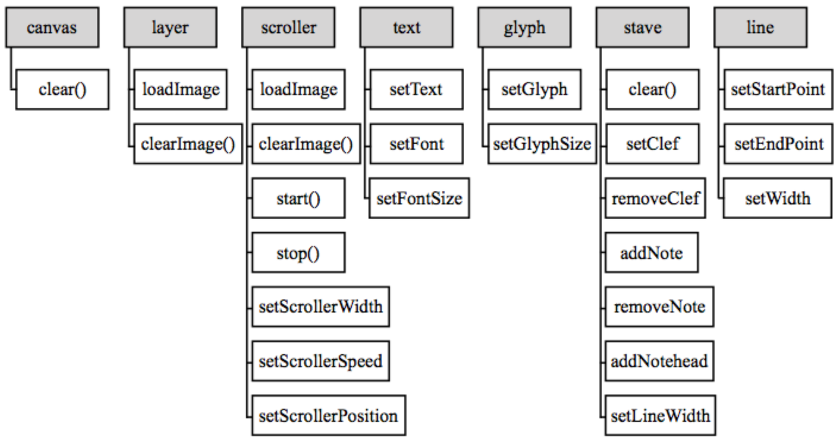


Figure 7. A tree diagram outlining the commands specific to canvas score object type.

Using the type specific commands of the stave object, for example, Common Practice notation can be added to a score. The following code example adds a stave to the canvas and then draws a sequence of clefs and a randomly populated sequence of stemless note-heads, the output of which can be seen below it in Figure 8.

```
def onConnect():
canvas.clear()
stave = canvas.addStave('stave', 0, 10, 100,
1100, 72, 2)
pitches = ('C4', 'C#4', 'C#+4', 'D4', 'E4',
'F4', 'F#4', 'G-4', 'A#+4', 'B4')
stave.setClef('treble', 40)
stave.move(-1000, 100, 10)
for i in range(0, 20):
note = stave.addNotehead (pitches[random.rand-
int(0, 9)], (i*50)+100, 1)
```

```
global finished
finished = True
```

Since notes are drawn on the stave using pitch notation, a clef needs to be defined before any notes can be added. To avoid this, note drawing could also be achieved, albeit in a much more manual way, with either the `addGlyph` command or the `addText` command, with the font set to the Bravura font [16]. These commands also allow the composer to override the default geometries used by the stave object. With all of these varied objects and commands, composers can create a hybrid system of notation combining not only Common Practice and graphic notation, but also animated notation paradigms using both the scroller object and the more general animated movement and fade commands.

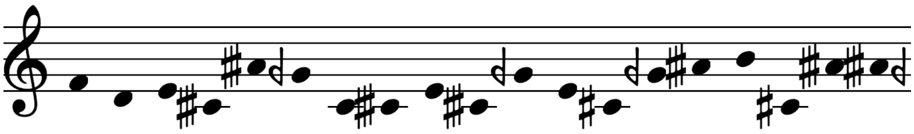


Figure 8. A python script responsible for creating a stave, generating a sequence of 20 pitches randomly gathered from a user-specified list, and scrolling the system off screen over 10 seconds.

5. ENHANCEMENTS TO THE SCORECREATOR APPLICATION

Early versions of the ScoreCreator consisted simply of a form window that allowed for the entry of the data that would populate the relevant xml tags of the score's opus file. When the "create score" button was clicked, this data was written out to the xml file, and this was zipped together with any image resources that the composer had specified. While much more convenient for a composer than editing a raw xml file, this still did not present much of an idea of what the score would look like in the player during the creation process. Refining the settings of a score could potentially require a lot of movement back and forth between the two apps. The ScoreCreator was also only capable of creating scrolling scores, and didn't offer access to some of the more advanced features of these.

ScoreCreator version 0.5.1 addresses both of these concerns. The creation of Talking board [1] style scores has now been added as an option, and work is currently underway to add Slideshow, Flash card and Canvas mode scores. The interface has also been improved to allow for more advanced preferences to be selected. To take scrolling scores as an example, it is now possible to customize the playhead, either by changing its colour or using an image in place of the usual line, and it is possible to make vertical as well as the standard horizontal scrolling scores (Figure 9). For vertical scores, the score can either be set to scroll from top to bottom or bottom to top. Most importantly, the new version has a preview window, so that composers will be able to view what their score will look like once imported into the player. While this preview isn't animated, the composer can still scroll through the image to be used, seeing how it will be affected by any scaling, and how much of it will be visible on the iPad's screen at any one time in both the landscape and portrait orientations of the device. When any settings that change the display of the score are altered in the main window, this is reflected instantly in the preview window.

Additionally, the ScoreCreator now handles images in a more intelligent way. It has always been possible to use the app to create a scrolling score from a series of equally sized tiles, but now it can search for and detect if such an image set exists upon selection of the first image. After prompting the user about the discovery, it can then automatically adjust the settings if desired. (It can also adjust settings in the other direction if the required number of images are not found.) On creation of a score, it will also check to see if any images are too large for display on the iPad, and will offer to scale or tile the original image as appropriate. (It won't do this if the user has manually done their

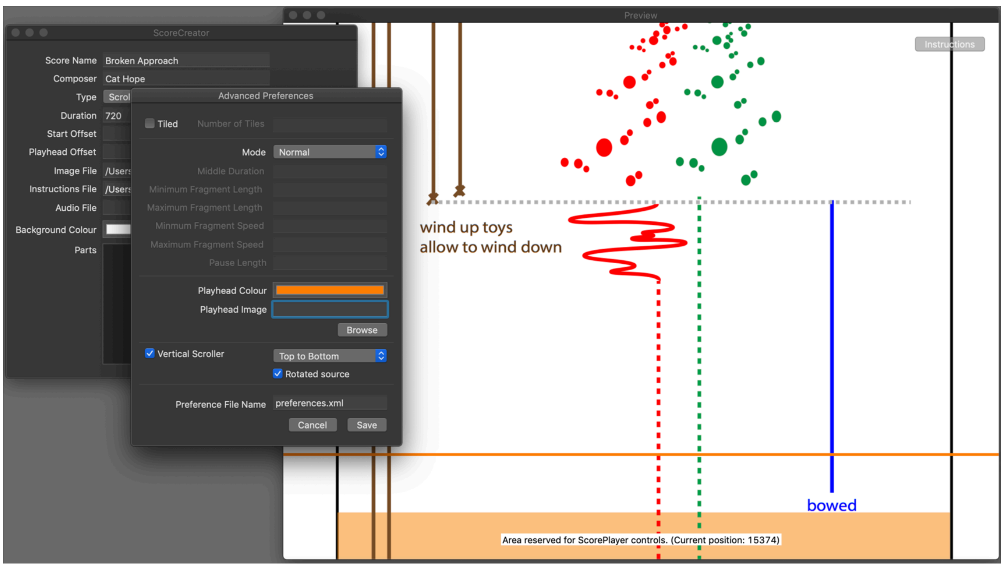


Figure 9. The updated ScoreCreator, showing the advanced preferences window for scrolling scores. The preview window can also be prominently seen, showing how the score will look as a vertical scroller.

own tiling, and will trust them to know what they are doing in that case.) The result of all of these changes is that the composer doesn't need to know quite so much about how the ScorePlayer handles their images internally, and can trust that their score will appear in the player exactly as it does in the preview window, leaving them to focus on the task of composition.

6. CONCLUSIONS

The advancements in the Decibel ScorePlayer project described above are aimed at opening the platform for experimentation by composers and others. It is hoped enhancements in the ScoreCreator app will allow for engagement with a range of novel score formats, and make them accessible to novice users with minimal programming experience. The development of the canvas mode now supports drawing commands sufficient to afford significant experimentation with new score formats and presentation methodologies. The implementations of scores discussed in this paper indicate that the canvas mode will support quite draw command intensive scores in a robust manner. The development of a Python library for the creation of external control apps forms something of a bridge between these two approaches, facilitating control of the canvas scoring mode and other ScorePlayer functions with minimal programming experience via a freely available and accessible language.

7. REFERENCES

[1] A. Wyatt, and C. Hope, "Animated Music Notation on the iPad (Or: Music stands just weren't designed to support laptops)," in Proceedings of the 2013 International Computer Music Conference (ICMC2013), Perth, 2013, pp. 201-207.

[2] C. Hope, A.Wyatt, and L. Vickery, "The Decibel ScorePlayer: New Developments and Improved Functionality, " in Proceedings of the 2015 International Computer Music Conference (ICMC2015),Denton, 2015, pp. 314-317.

[3] C. Hope, A. Wyatt, and L. Vickery, "The Decibel ScorePlayer - A digital tool for reading graphic notation" in Proceedings of the International Conference on New Tools for Music Notation and Representation (TENOR'15), Paris, 2015, pp. 59-70.

[4] L. Vickery and S. James, "Tectonic: a networked, generative and interactive, conducting environment for iPad," in Proceedings of the 2016 International Computer Music Conference (ICMC2016), Utrecht, 2016, pp. 542-547.

[5] L. Vickery, "Rhizomatic approaches to screen-based music notation" in Proceedings of the 2016 New Interfaces for Musical Expression Conference (NIME2016), Brisbane, 2016, pp. 394-400.

[6] S. James, C. Hope, L. Vickery, A. Wyatt, B. Carey, X. Fu, and G. Hadju, "Establishing connectivity between the existing networked music notation packages Quintet.net, Decibel ScorePlayer and MaxScore," in Proceedings of the International Conference on New Tools for Music Notation and Representation (TENOR'17). A Coruña, 2017, pp. 171-183.

[7] A. Wyatt, L. Vickery, and S. James, "The Canvas Mode: Rapid Prototyping For The Decibel Scoreplayer," in Proceedings of the Australasian Computer Music Conference 2018 (ACMC2018), Perth, 2018, pp. 99-108.

[8] D. DeVille, "Dynamic linking on iOS." Last modified April 2, 2014. http://ddeville.me/2014/04/dynamic-linking

[9] M. Wright, "The Open Sound Control 1.0 Specification." Last modified March 26, 2002. http://opensoundcontrol.org/spec-1_0

[10] Python Software Foundation. "Python Package Index." Accessed October 26, 2018. https://pypi.org

[11] Apple Inc. "Bonjour Overview." Last Modified April 23 2018. https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/NetServices/Introduction.html.

[12] WC3. "Extensible Markup Language (XML) 1.0 (Fifth Edition)." Last modified February 7, 2013. https://www.w3.org/TR/xml/.

[13] D. Kim-Boyle, "Real time generation of open form scores." Proceedings of Digital Art Weeks, Zurich. 2006, pp. 3-14.

[14] A. Vida, Accessed October 26, 2018. http://www.andrevida.com

[15] R. Smith, A practical and theoretical framework for understanding contemporary animated scoring practices. PhD Diss. Rensselaer Polytechnic Institute, 2016.

[16] Steinberg Media Technologies GmbH. "SMuFL: Standard Music Font Layout." Accessed October 26, 2018. https://www.smufl.org/fonts/

[17] Python Software Foundation. "Zeroconf 0.23.0." Accessed June 1, 2019. https://pypi.org/project/zeroconf/

[18] Python Software Foundation. "Python-OSC 1.7.0." Accessed June 1, 2019. https://pypi.org/project/python-osc/

The Monthly Acid Pattern – an accessible notation system for Acid House collaboration

Dylan Davis
Swinburne University
Melbourne, Australia
ddavis@swin.edu.au

ABSTRACT

Scholars continue to investigate how communities are built by sharing histories, norms and values within the 'third spaces' enabled by social media technologies. How can these third spaces be harnessed to explore collaborative and experimental compositional practices, and, in turn, what can practising music as a 'shareable' culture reveal about community building as it shifts to digital platforms. The Monthly Acid Pattern Group used the compositional schematic of the Acid Pattern and a particular analogue synthesizer, the Roland TB-303, as the basis for the sustained production of interpretative works using online collaborative and publishing platforms over a four-year period. This project contributed to further understanding of how the practices and cultures of music composition are shared and can build community. With the use of an emerging online platform (SoundCloud), the research project made an innovative contribution to methodologies of documenting, and enabling, the interpretative practices of an online community as it emerged. The works that were created demonstrated new possibilities for accessible modes of notation, instrumentality and compositions within digital music cultures.

1. INTRODUCTION

An Acid Pattern is the melodic element of an electro acoustic composition in the Acid House musical genre. The Monthly Acid Pattern Group posted an original Acid Pattern each month and members of the community created their own renditions of each month's pattern.

The Monthly Acid Pattern Group was community of the third sector [1], collaborative music project based around the Acid House music genre, the Roland TB-303 monophonic synthesizer and the visual representation of the music notation for this instrument. This community had a measure of commitment to a set of shared values, norms, meanings, history and identity, in short to a particular culture[2].

The aim of the research was to explore of the notation system used for the Monthly Acid Pattern. Each month's Acid Pattern and its particular visual representation was the starting point for a range of reinterpretations and renditions. In addition, this research observed the building and development a community of the third space as defined by Rose [3] and Etzioni [2]. This community building and development occurred on the SoundCloud[1] platform within its defined group structures functionality. This group featured allowed a shared space to share music and forum space for group discussion. SoundCloud hosting of music also allows for users to comment the original pattern and on particular users' renditions. SoundCloud has since removed its' group functionality.

2. RESEARCH CONTRIBUTION

Acid House Music has been identified as a strong contributor to youth cultural identity but the mechanism of production has not been studied in any depth [4]. This research has two main contributions to the field. A unique system of the musical notation based specifically for the melodic component of Acid House music and the instrument which defined the Acid House sound, the Roland TB-303 instrument and varied body of new music created utilizing this system. Secondly whilst Acid House has a strong place based cultural identity [5] an unexpected output of this research was the development of a global community of musicians that came together because of a particular shared cultural bond, this community building gives support to Rose's [1] definition of the community of the third space and also gives weight to Young's [6] argument that community has no need to be face to face.

3. METHOD

3.1 Data Sources

The data that was analysed consisted of the monthly acid patterns submitted by various musicians over a four year period from 2011 until 2015. These acid patterns were a series of graphical representations of the TB-303 melodic contribution to the electro-acoustical composition. In addition, data was gathered from each month's recordings of

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

the original composition and the range of renditions made from each month's Acid Pattern. Further data was gathered from the feedback and comments of the pattern writers and contributors to the recordings on SoundCloud.

3.2 Data Analysis

Content Analysis, described as a flexible method of text analysis [7] was used to analyze the data. Content Analysis has been used in many studies [8] and the number of studies reporting the use of content analysis has steadily increased since 1991 [9]. Content Analysis can be used to analyze a range of text data, such as text in audio, print, or electronic format. This text data can be obtained from a range of sources; interviews, focus groups, observations, articles, books and manuals[9]. As stated in previous section the data analyzed was drawn from the Monthly Acid Patterns submitted to SoundCloud by each musician and the renditions of those patterns by contributing musicians. The collation of this data assisted in understanding the themes and activities involved [8], [10]. The technique of immersion in the data helped identify key themes and supported the development of the initial coding scheme for the data. These codes were organized into related categories such as clarity, communication, notation and other variables [11] These approaches to coding incorporated both conventional content analysis to develop codes and direct content analysis techniques to identify variables [9] It is recommended to have multiple researchers working on coding [12] to ensure reliability of the coding process. To ensure the coding schemas that were developed for data analysis were reliable, they were checked and refined throughout the coding process with the research peers.
The data was used for several distinct purposes. Firstly, to identify the variables in approaches to device specific musical notation. Secondly, the data gave insights into how online co-creation gave rise to a community of musical practice amongst geographically diverse group of musicians.

Thirdly, these renditions presented a narrative of each months Acid Pattern, the activities that occurred and the processes and practices used within those activities. The narrative identified key points in each month and these were mapped to the stages of the groups processes where possible. The narrative of each month gave greater insight into the suitability of the Acid Pattern as a notation system and the community's development and growth.

4. RESULTS

4.1 The TB-303

To program the TB-303's sequencer the musician needed to know a number of variables, pattern length, pitch data, pitch modification and time data. To communicate this information as an Acid Pattern the musician had to be able to describe the pattern length, n-16 steps, with the ability to chain up to 4 patterns. The pitch data, which note on which step. The pitch modifications, such as transpose up, transpose down, slide and accent. In addition, the musician also had to be able to communicate the time data for each step, whole note, tie or rest. Finally, the musician had to communicate the Acid Pattern using an accessible notation system because the majority of Acid Pattern Group's community did not have sight reading or ear playing skills. As demonstrated in Figure 1 the TB-303 utilizes both traditional Western notational elements and more accessible elements. The pitches are described using letters, as are the pitch modification such as slide, accent, transpose down and up. The time data uses symbols and notion for note, tie and rest.

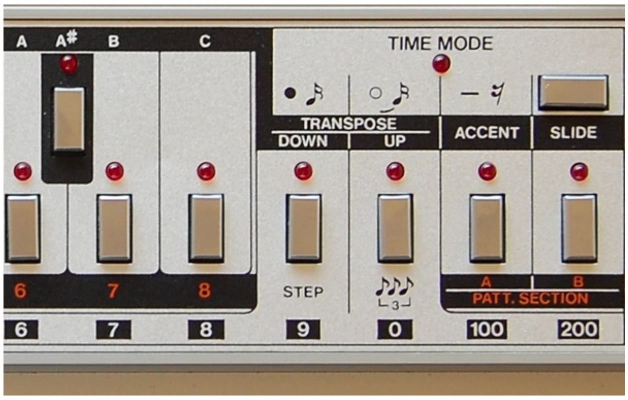


Figure 1. The TB303.

4.2 The Acid Patterns

The Acid Pattern Group utilized a predefined pattern sheet to clearly communicate the pitch data, pitch modifications and time data necessary to successfully program the pattern into the TB-303 as illustrated in Figure 2. This allowed each musician to communicate which note, the particular pitch modification on each note, transpose, accent or slide and also the timing data, note, tie or rest. In the pattern sheet the time data uses only the symbol form from the TB-303, rather than the notional explanation. Due to the specific way in which the TB-303 is programmed the pitch data and time data are entered separately. The pattern sheet also allowed the musicians to enter further instruction regarding each pattern and for retrieval purposes denote which memory slot on the TB-303 they had written the pattern.

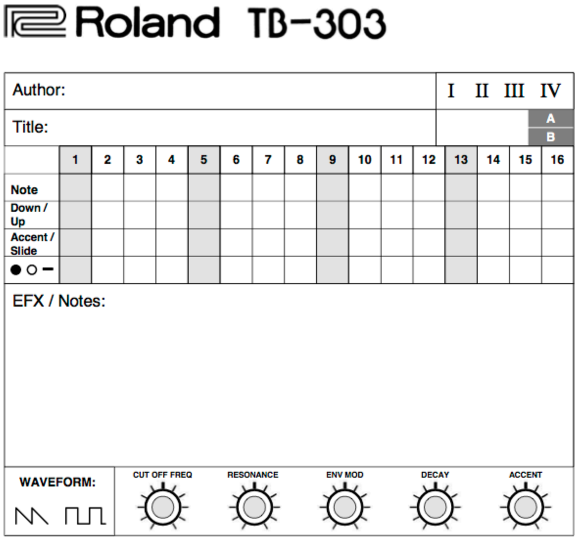
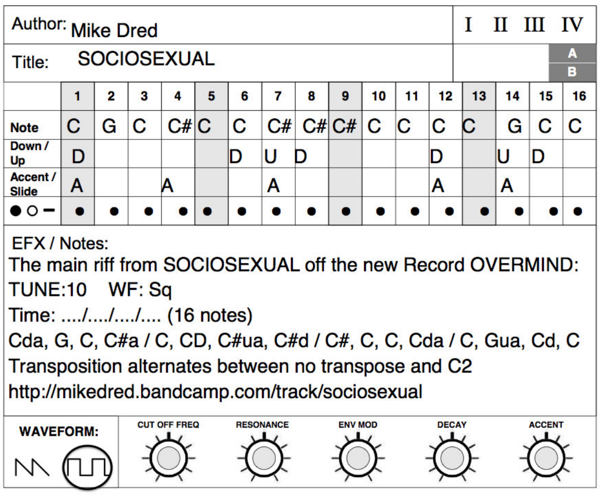


Figure 2. Blank TB303 pattern sheet.

As demonstrated in Figure 3, Mike Dred utilizes the pattern sheet to communicate to other group members the pitch data for each step of the pattern in the form of the note and the pitch modifications, in this case transpose down an octave, transpose up an octave and utilize the accent. This particular pattern's time data is straightforward with no ties or rests. The use of notes section allows the musician to further communicate information regarding the pattern and how to perform it, such as using the note transpose function on the TB-303 whilst the pattern is playing.



Author: Mike Dred — I II III IV

Title: SOCIOSEXUAL — A / B

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Note	C	G	C	C#	C	C	C#	C#	C#	C	C	C	C	G	C	C
Down / Up	D					D	U	D				D		U	D	
Accent / Slide	A			A			A					A		A		
● ○ –	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

EFX / Notes:
The main riff from SOCIOSEXUAL off the new Record OVERMIND:
TUNE:10 WF: Sq
Time: ..../..../..../.... (16 notes)
Cda, G, C, C#a / C, CD, C#ua, C#d / C#, C, C, Cda / C, Gua, Cd, C
Transposition alternates between no transpose and C2
http://mikedred.bandcamp.com/track/sociosexual

WAVEFORM: — CUT OFF FREQ — RESONANCE — ENV MOD — DECAY — ACCENT

Figure 3. Mike Dred's acid pattern sheet for Sociosexual February 2014. *https://acidpattern.bandcamp.com* (2014)

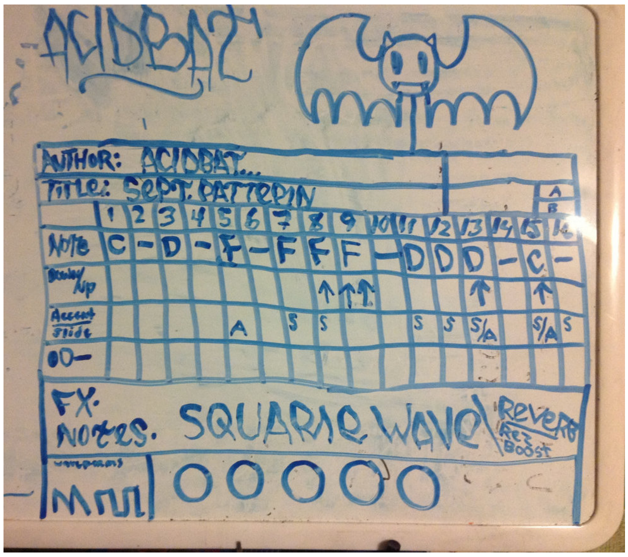


Figure 4. Acid Bat's acid pattern sheet for September 2014. *https://acidpattern.bandcamp.com* (2014)

As documented in Figure 4, musicians also created their own pattern sheets utilizing whatever notation tools they had on hand, but these bespoke pattern sheets still communicate the required information. Much like the bespoke pattern documented in Figure 5 both musicians eschew the group pattern sheet for a DIY version that uses very similar grid representation to display the month's acid pattern.

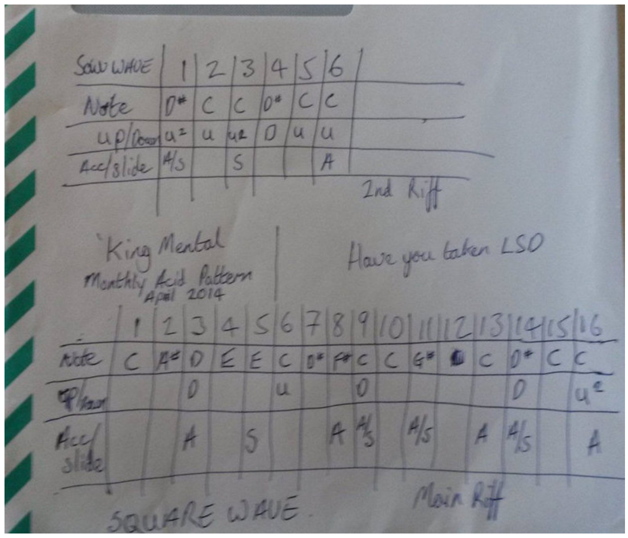


Figure 5. 'King Mental's acid pattern sheet for April 2014. *https://acidpattern.bandcamp.com* (2014)

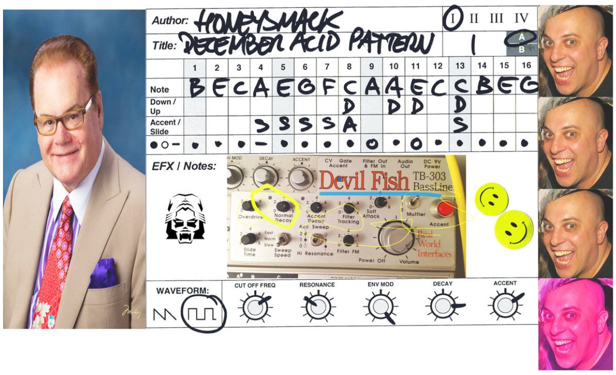


Figure 6. Honeysmack's acid pattern sheet for December 2014. *https://acidpattern.bandcamp.com* (2014)

In addition to bespoke pattern sheets, musicians would also customize the group pattern sheets with elements of their personal brand, as demonstrated in Figure 6. In addition to the personal branding, Figure 6 also demonstrates the use of ties and rests in the time data section of the pattern sheet.

Another common element across the majority of the acid patterns was the pattern length, very few patterns were more than 1 bar, though some such as documented Figure 2 had transposed variations.

4.3 Online Community

SoundCloud's interactive comments allowed musicians to interact directly with the pattern writer and the pattern creators to interact with those creating renditions of their patterns. Comments were for the most part positive, and related to various aspect of the pattern and its character. In addition, there were comments around the inspiration behind the pattern, the types of accompanying machines used and other comments of technical nature.

5. FINDINGS

The Acid Pattern demonstrates an example of accessible musical notation system that is genre specific, Acid House music and machine specific, for the TB303 sequencer. The Acid Pattern notation removes the need for sight reading skills, which can cause recognition errors with musicians [13]. The pitch data was limited to single notes because the TB-303 is a monophonic synthesizer and transcribed using the letters as notes that appear on the TB-303 as shown in Figure 1. The pitch data modification such as slide, accent whilst part of traditional Western musical notation had to be communicated in an accessible way that the participating musicians could understand, and in keeping with the design of the TB-303. The same consideration had to be given to up and down transpose modification. The specific style of the TB-303's sequencer which separates the entry of time and pitch data meant any user of the TB-303 could enter the pitch data and then the time data and get the correct pattern. The TB-303 is not played in the traditional sense, patterns such as those shown in Figures 3, 4, 5, 6 are programed into it and then the sequencer is started, there is no need for the musicians to audiate [14] the pattern from the notation and if musicians are unable to transcribe

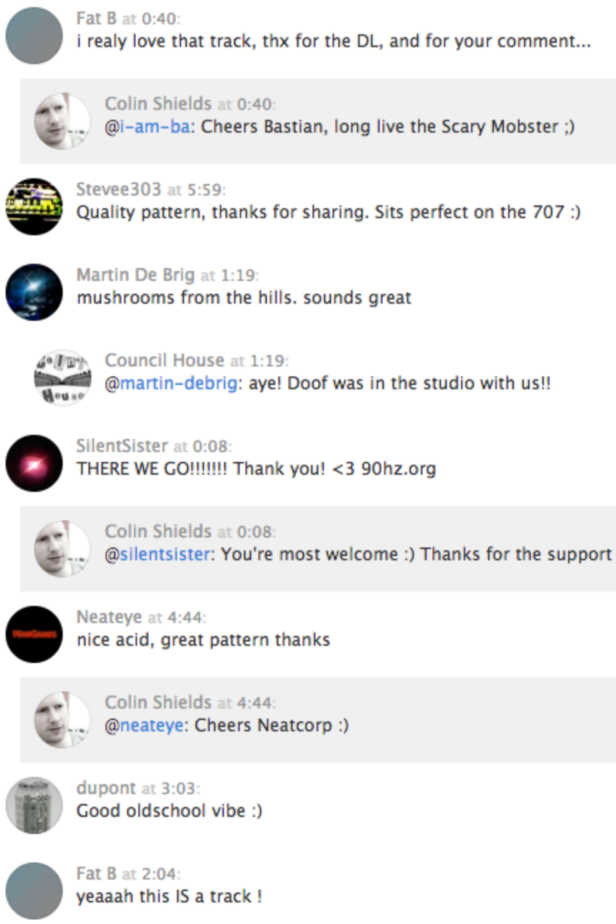


Figure 7. SoundCloud comments on July Acid Pattern 2011 : The Acid North (Moody B & Colin Shields) - Acid in the Hills. *https://soundcloud.com/colin-shields/july-acid-pattern-the-acid* (2011)

the pattern from listening are also able to input the correct pattern using this notation system.

The Acid Pattern Group's success can be seen in the creation of original music, one original acid score each month and a large number of renditions each month for a four year period. From 2011 to 2015 there were at least ten to twenty renditions of the original pattern each month with their own style of accompaniment such as drums, samples and other synthesized sounds. A collection of two years' worth of music created by the Acid Pattern Group's members is available via the Bandcamp website [2].

The ability of a globally distributed group of musicians to create such a large number of works within a monthly deadline, and consistently for a number of years can be ascribed to number of factors. Firstly, the acid pattern notation was highly accessible to this diverse group of musicians from around the world and allowed each musician to work on their piece easily. Secondly, the shared cultural

2 https://acidpattern.bandcamp.com

identity of the musicians around Acid House music [4] brought this group of musicians together and as the group started creating music this brought new members into the group. Finally, the nature of the SoundCloud group allowed this globally distributed group of musicians to become a community which created, shared, collaborated, commented and advised on each other's compositions. As demonstrated in Figure 7, the original track for each month and the renditions would have numerous comments and feedback. In additional the conversational aspect of this system allowed for interactions between each month's pattern creator and those undertaking renditions. These ongoing conversation between members of the group, created networks and formed relationships between this geographically diverse community. This community distributed globally would never meet face to face, they were connected by interest, activity and a shared purpose[6], mediated through SoundCloud, it was a community of the third space bounded by this shared culture and identity [1].

6. CONCLUSIONS

In this paper, it has been demonstrated that the Acid Pattern sheet as an accessible notation system for the creation of Acid House music utilizing the TB-303 and its many clones has been a successful form of notation. The sheet has been used by many individual musicians from around the world since 2010. Whilst the success of the group is not solely attributable to the design of the notation sheet itself, there are also strong cultural, community and identity influences that drive this group. Since the demise of SoundCloud groups functionality, the Acid Pattern Group as continued in somewhat low key form utilizing the Facebook social media platform. The current iteration of the group has 334 members and has been running since 2017, with a fresh pattern and a number of renditions each month.

Whilst the Acid Pattern Group is ongoing, the use of this Acid Pattern sheets as a notation system continues to be utilized by a range of musicians, outside of the group as a way of documenting, archiving, sharing and writing acid house music in their professional practice.

Acknowledgments

The author would like to acknowledge the contribution made by the musicians submitting patterns each month over the course of this project and the musicians that created their own renditions of each months' acid pattern. The author would also like to acknowledge the Roland Corporation for creating the TB303 and Phuture for creating Acid House music.

7. REFERENCES

[1] N. Rose, "5 Community," *Powers Free. reframing Polit. thought*, pp. 167–196, 2010.

[2] A. Etzioni, "Positive aspects of community and the dangers of fragmentation," *Dev. Change*, vol. 27, no. 2, pp. 301–314, 1996.

[3] N. S. Rose, "Powers of freedom: reframing political thought," in *Powers of freedom: reframing political thought*, .

[4] D. Hesmondhalgh, "The British Dance Music Industry : A Case Study of Independent Cultural Production," *Br. J. Sociol.*, vol. 49, no. 2, pp. 234–251, 1998.

[5] H. C. Rietveld and others, *This is our house: house music, cultural spaces and technologies.* Ashgate Publishing Ltd., 1998.

[6] I. M. Young, "The Ideal of Community and the Politics of Difference," *Soc. Theory Pract.*, vol. 12, no. 1, pp. 1–26, 1986.

[7] S. Cavanagh, "Content analysis: concepts, methods and applications," *Nurse Res.*, vol. 4, no. 3, pp. 5–13, 1997.

[8] S. Elo and H. Kyngäs, "The qualitative content analysis process," *J. Adv. Nurs.*, vol. 62, no. 1, pp. 107–115, 2008.

[9] H.-F. Hsieh and S. E. Shannon, "Three approaches to qualitative content analysis.," *Qual. Health Res.*, vol. 15, no. 9, pp. 1277–88, Nov. 2005.

[10] R. Tesch, *Qualitative research : analysis types and software tools*. New York : The Falmer Press, 1989; New York : The Falmer Press, 1989, 1990.

[11] M. Q. Patton, *Qualitative research and evaluation methods*, 3rd ed.. Thousand Oaks, Calif. : Sage, c2002.; Thousand Oaks, Calif. : Sage, c2002., 2002.

[12] S. Elo, M. Kääriäinen, O. Kanste, T. Polkki, K. Utriainen, and H. Kyngas, "Qualitative Content Analysis: A Focus on Trustworthiness," *SAGE Open*, vol. 4, no. 1, pp. 1–10, 2014.

[13] J. A. Sloboda, "Visual perception of musical notation: registering pitch symbols in memory.," *Q. J. Exp. Psychol.*, vol. 28, no. 1, pp. 1–16, 1976.

[14] E. Gordon, *Learning sequences in music: Skill, content, and patterns: A music learning theory*. Gia Publications, 2003.

PERFORMING THE COMPOSITIONAL ACT WITH BOUNCY CASTLES, SOAP AND SHH

Andy Ingamells
Royal Birmingham Conservatoire
(Birmingham City University)
Birmingham, UK
andy.ingamells@gmail.com

ABSTRACT

My practice-based research has led to a rethinking of the relationships between composer, performer and listener in my own creative work through an interpretation of a diagram by experimental composer George Brecht. Through the reconfiguration of this diagram I have developed a framework in which the act of composition can be performed via the activities of 'reading', 'performance' and 'playing', with the focus on an expanded notion of traditional score-reading that makes the act of reading manifest onstage as part of the physical theatricality of musical performance. This approach can be used as a site for further experimentation by other interdisciplinary creative practitioners.

1. INTRODUCTION

In Western classical music the performers are most likely to read notation onstage. This is true even in the case of famous examples in the history of experimental music, a tradition "characterized by its radical opposition to and questioning of institutionalized modes of composition, performance, and aesthetics" [1]. When interviewed about performing John Cage's *4'33"*, a piece which has no notated sounds and requires the performer(s) to sit in silence onstage, pianist David Tudor stated: "I was looking at the first movement and I was turning pages because I was reading the score in time" [2, p. 86]. Tudor's statement shows that even a piece with no notes to play can still have something for the performer to read. In my own experience of watching live performances of experimental music I want to know what is written on the pages in front of the players, especially in the case of graphic scores such as *Treatise* by Cornelius Cardew [3] (figure 1) and *Kandinsky Studies* by Deborah Pritchard [4] (figure 2) where the visual content of the score is as compelling as its sounding result.

However, in performances of such pieces the visually interesting score is hidden from the view of the audience. The content of the score is only communicated via sound, so that the graphic score may as well be written in traditional musical notation or completely ignored. Composer and writer G Douglas Barrett suggests that Manfred Werder's piece *2010* specifically alludes to this "hiddenness of the score in performance, its physical absence from the view of the audience" whilst describing scores themselves as laying "along the edges of the musical frame" [5, p. 57]. Scores and notation are neither the music itself, nor are they completely outside the music. They remain unseen by the listener during performance and translated by the performer via sound. As a practitioner interested in the theatrical and visual aspects of musical performance, I want this act of reading to be demonstrated to the audience directly. I want to explore the relationship between composer-performer, notation and audience by making the act of score-reading part of the theatre of musical performance itself.

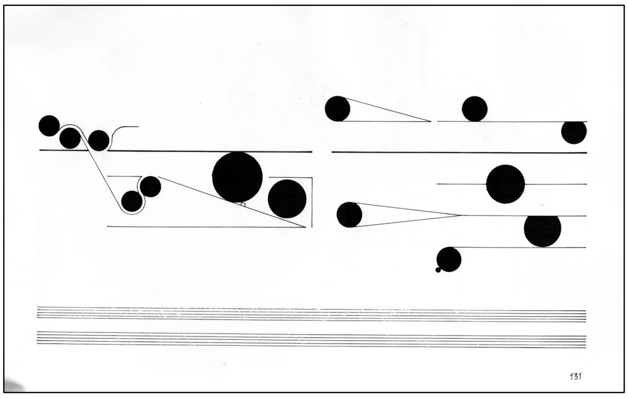


Figure 1. *Treatise* (1967) by Cornelius Cardew [3, p. 131].

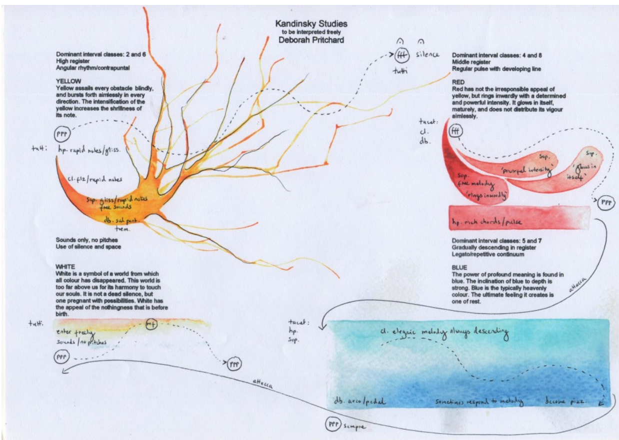


Figure 2. *Kandinsky Studies* (2016) by Deborah Pritchard [4].

Musicologist Adam Harper, in his 2009 blog post *What is a composer*, describes how the act of reading is an under-explored area of musical practice: "[Text-based formats] allow the act of reading the text to occur at the level of performance as well as listening, so that the two activities begin to merge. This is a rich and currently pretty esoteric area of aesthetic possibility, with few listeners appreciating music in this way" [6]. Harper is referring specifically to written text, but in this paper I take a broader view of 'text' to mean anything that can be read onstage as part of a musical performance; both words and notation in the broadest sense. Musical scores are compelling documents in which a great deal of information can be encoded and displayed. The merging of reading and performance allows for such information to be communicated with an audience, and for this area of experience to be opened up to people who may not be able to read written music. This has the potential to remove the need for spoken explanation of the works prior to, or following, performance, allowing for the possibility of an enhanced aesthetic experience of the music and compositional act.

2. RELATIONALITY

Musicologists will often discuss the relationships between notes and sounds, but musical situations are made up of a multitude of different relationships. Barrett calls for a focus upon these relationships in *After Sound*, suggesting that music is "inherently premised upon structures of collaboration and social relationality" [5, p. 134], whilst percussionist Greg Stuart states that experimental music in particular "attempts to radically rethink the relationship between composition, performance and listening" [7]. A creative practice rooted in experimental music is thus in a strong position to challenge notational conventions in performance by reconsidering such relationships. This is affirmed by Christopher Small who, in his study of African-American music-making *Music of the Common Tongue*, remarks that: "Genuine musical innovation, as we have seen, is a matter not just of new sounds or techniques but of new forms of relationship" [8, p. 319].

Small questions the relationship between notation and the creative compositional act within classical music: "It is only in the classical tradition that notation has taken over as the medium through which the very act of creation takes place" [8, p. 43]. However, I posit that notation can be reframed as an integral part of the theatre of a musical performance, bringing the creative compositional act onstage, without recourse to the discourse of free improvisation. By doing so the composer can undertake the act of creation live as a performer, blurring the roles of composer and performer to become composer-performer.

3. PRACTICE-BASED RESEARCH

I am an experimental musician who develops unusual methods of composition that blur the line between composer and performer, and conduct practice-based research into how the legacy of certain types of experimental music from the 1960s can lead to the formation of new approaches to composition through the creation of original works. My practice-based approach is best summarised by the following statement from Graeme Sullivan's *Art Practice as Research: Inquiry in Visual Arts*: "An artwork is a form of individual inquiry … Art practice is theoretically robust, ideas-based, purposeful, and strategic, and it makes use of forms and methods that are connected to, but distinct from, traditional systems of inquiry" [9, p. 244]. My inquiry is into my own practice as a composer-performer, and is conducted through that practice. Alternative approaches to musical composition and notation can arise through direct reflection on and reaction to the creative outcomes of this practice.

3.1 Shh

An example of how I have approached the integration of scores into the theatricality of a musical performance in my own creative practice is a 1-minute piece for solo vocalist and audio backing track called *Shh*. In *Shh* I intervene in a radio broadcast by repeating a single vocal sound to change the meaning of a phrase. I achieve this by cropping together all the occurrences of the 'Hit Music' identifier on British commercial radio station CapitalFM during a typical hour-long broadcast to make a 1-minute audio track, then saying 'shh' every time the identifier appears in order to blur the two sounds and create the phrase 'shit music'. The piece was originally conceived as a video work with a close-up shot of my lips producing the sound, but is now often performed live.[1] The meaning of this piece is ambiguous: the audience could infer that I am saying pop music is shit, or alternatively that my own performance is shit. When performed alongside other pieces in a concert setting it could look like I am trying to say that the other music in the concert is shit. I prefer to maintain this ambiguity: if the audience are questioning what I mean then I consider them to be active, rather than passive, listeners.

The vocal sound 'shh' is ordinarily used to tell people to be quiet. Classical concert audiences are conventionally supposed to sit quietly during a performance, and people might tell each other to 'shh' if they fail to abide by this convention. This sound is thus a by-product of musical performances, an aspect of the 'extramusical' which is foregrounded in this piece. The only live musical action that takes place in the performance is the synchronisation of the 'shh' sound with the beginning of the phrase 'Hit Music' on the audio track. Traditional score-reading produces a synchronised sound-image of written music, and I see performing acts of synchronisation as an expansion of this traditional practice. The pre-recorded track serves the purpose of an audio score that is used to indicate when the 'shh' sound should be made. In my performance I follow this audio track in the same way that a score of written musical notes would be followed. What I realised from *Shh* was that the score, in this case a backing track, can be the

[1] Original video work of *Shh*: https://vimeo.com/230430041

source of aesthetic experience in the performance. The score is the primary musical material, and my reading of the score through the synchronisation of the 'shh' sound serves as the reason to play it.

3.2 Waschen

A development of this idea is demonstrated in another solo performance entitled *Waschen.*[2] Here the score material can be read by both the performer and audience simultaneously, with the audience being able to follow the score-reading process in a very direct way. Information is communicated to the audience both sonically and visually in a way that goes beyond the purely sonic experience of *Shh*. I perform *Waschen* as a solo, untrained singer, and read my body in a mirror, with the image of my body serving as musical notation to be read in real time. The word 'waschen' is written all across my body with thick black marker pen. The words are written in a seemingly arbitrary fashion with no formal pattern, but serve the purpose of written music notation as something to be read onstage during a performance. As I perform the piece I take a bar of soap and wash the words from my body whilst reading aloud the words in a mirror. When I touch a particular part of a word I sing it, reading my body as though it is a musical stave, shown in diagrammatic form in figure 3. The higher up my body the word is written, the higher I sing, and the lower down it is written, the lower I sing. When the ink is dark I sing *forte*, and as the ink fades I *diminuendo*. When the ink has completely disappeared, no sound comes out of my mouth. This process should become clear to the audience as the piece progresses.

The text on my body is not just a score for reading, but forms an essential theatrical part of the performance itself. The act of reading and responding to the written text musically can be followed by the audience, and through this process the act of reading onstage is made manifest. The audience are not required to join in because this would be overly didactic, but instead it is possible for them to remember the process of the piece and recreate it themselves. In this way the piece is akin to a brief Fluxus text score that can be easily memorised and recreated, such as Alison Knowles' *Nivea Cream Piece* (1962, figure 4) [10] or George Brecht's *Flute Solo* (1962) [11, p. 24] which contains just two words: disassembling, assembling.

4. FLUXUS

Fluxus was a loose international grouping of composers, poets and artists who began working with indeterminate methods of art-making in the late 1950s. Some met at John Cage's composition class at the New School for Social Research in New York, although Cage himself is not considered a member of the group. Fluxus composers created much of their work through text scores that describe musical or performative actions in brief poetic terms and engage the performer in a straightforward act of reading (see figure 4).

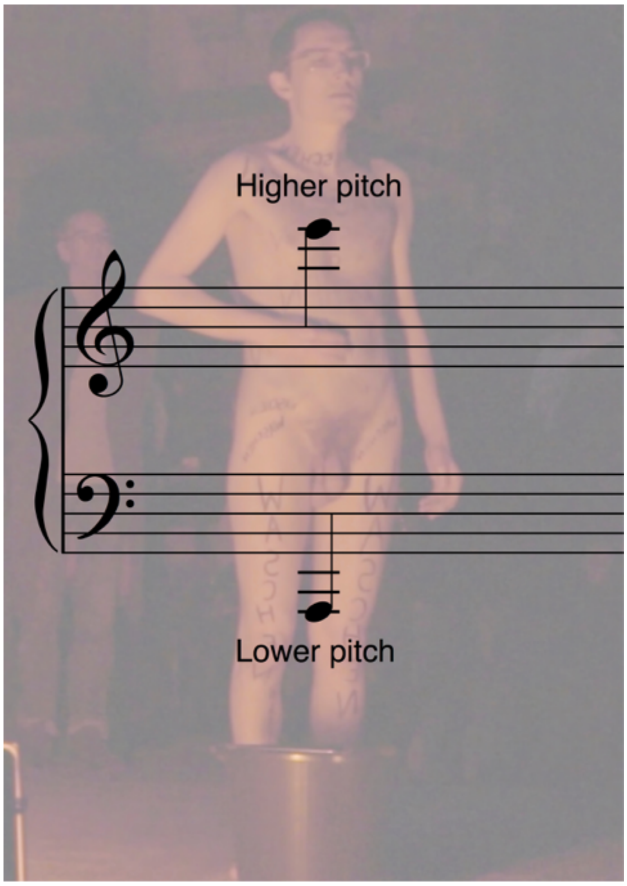


Figure 3. Screenshot from video documentation for *Waschen* (2015) showing how I read my body as a score.

#3 —
Nivea Cream Piece (November, 1962) — for Oscar Williams

First performer comes on stage with a bottle of hand cream, labeled "Nivea Cream" if none is available. He pours the cream onto his hands, and massages them in front of the microphone. Other performers enter, one by one, and do the same thing. Then they join together in front of the microphone to make a mass of massaging hands. They leave in the reverse of the order they entered, on a signal from the first performer.

Premiered November 25th, 1962 at Alle Scenen Theater, Copenhagen, at Fluxus Festival.

Figure 4. Text score by Fluxus artist Alison Knowles [10].

Fluxus works were irreverent towards the institutionalised musical establishment of the day, and the indeterminate nature of such works led to a blurring of boundaries between artistic disciplines, which practitioners termed 'intermedia'. Fluxus had a significant influence on much conceptual and performance art, and still resonates today – particularly in relational and participatory practices that focus on the experience of the viewer. It is the experience of the viewer that I suggest will be enhanced by an expanded approach to score-reading within musical performance. Art historian Hannah Higgins summarises: "Through the overlapping of touch, taste, smell, sound, or speech, Fluxus intermedia works have, at some level, the principle of directness, non-mediation, and unprocessed experience at their core" [12, p. 73]. Higgins' description

[2] Video documentation of *Waschen*: https://vimeo.com/230645019

of Fluxus work as being based in “directness, non-mediation, and unprocessed experience” can be used as a description of practice-based research, and is how I approach my own compositional-performance activity. Although this paper is in itself a mediation between the reader and musical works, my use of a first-person account of the creative outcomes reflects the directness of a Fluxus-informed practice.

5. GEORGE BRECHT DIAGRAM

Fluxus offers a clear reference point through which to rethink the relationships between composer-performer, notation and audience in the form of a diagram by one of its leading composers, George Brecht. Brecht codified all the relationships within a musical performance in the diagram that he drew in his notebook in 1959, shown in figure 5, illustrating how each role is related to every other role. His diagram shows the figure of the composer being related to the other protagonists through the activities of composition, criticism and improvisation. I am concerned with the figure of the composer-performer, someone who integrates both roles at the same time in order to bring the creative compositional act onstage. Brecht’s original pentagonal diagram can thus be reconfigured to put composer and performer together, making composer-performer, with the relationships rearranged accordingly. This results in a square with an intersection in the middle, shown in figure 6.

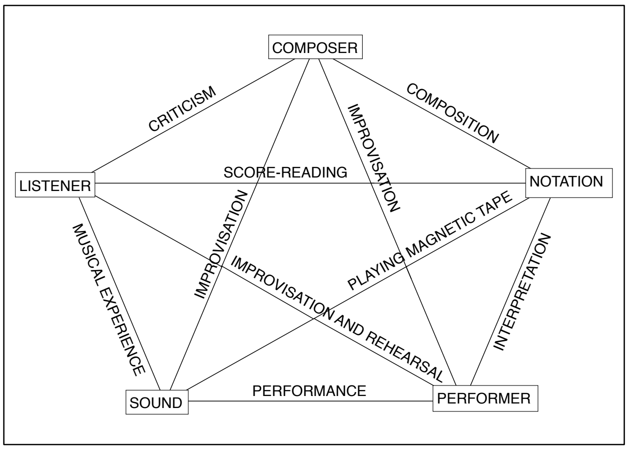


Figure 5. Computer-processed version of George Brecht’s original hand-drawn diagram from 1959 [13, p. 127].

In the middle of this new diagram there is a meeting point where improvisation, performance and score-reading collide. These terms can be used as a starting point to better define the activity of a composer-performer. Through reconfiguring Brecht’s diagram, a conception of the composer-performer is revealed as someone who combines the activities of improvisation, performance and score-reading into a fully integrated practice, shown in a triangular frame in which musical works can be plotted according to how much of each characteristic they embody (figure 7). These characteristics were already occurring independently within my creative work, and the reconfiguration of Brecht’s diagram suggested that the focus on the performance of score-reading should be combined with an improvisatory approach if the creative compositional act was to be successfully performed onstage.

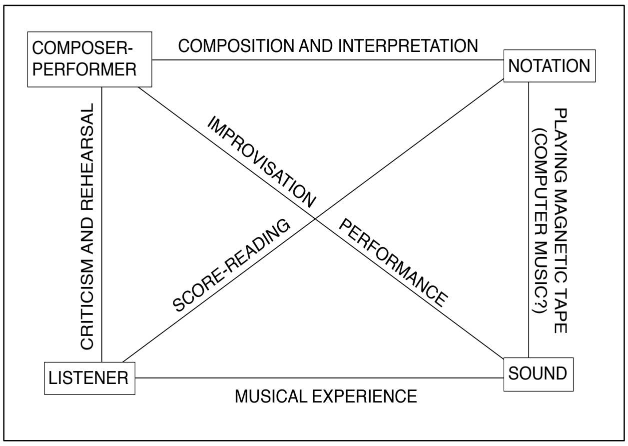


Figure 6. Square-shaped rearrangement of Brecht’s original diagram.

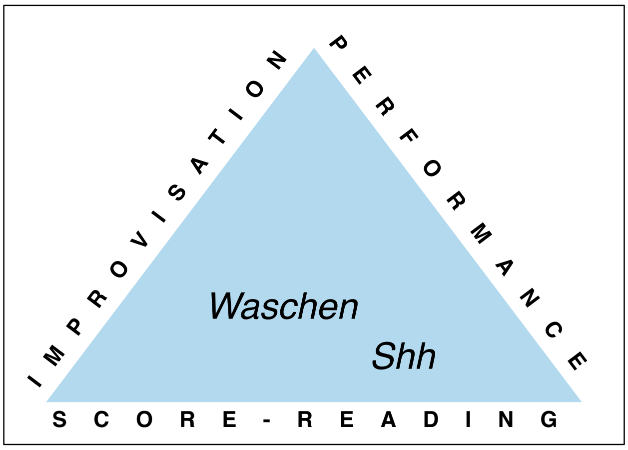


Figure 7. Triangular frame with *Shh* and *Waschen* plotted according to how much of each characteristic they embody.

Prior to reconfiguring Brecht’s diagram I considered the non-written parts of my work as being ‘indeterminate’, coming from my experience as a practitioner of experimental music. ‘Indeterminacy’ was the term favoured by John Cage for pieces that gave the performers a significant degree of choice in what they played. However, the three terms emerging from the diagram suggested that the familiar activity of improvisation could be more relevant to the conception of the composer-performer who performs the creative compositional act onstage. Small remarks that: “Composition and performance are [...] part of a single act which Europeans call improvisation but others call, simply, playing” [8, p. 46]. This provides a better term for improvisation in my work, one that more clearly defines how I use it: ‘playing’. I offer a blend of the meanings of the verb ‘playing’ in my work, which refers to both playing music or just playing by engaging in an activity for the sake of enjoyment.

6. MAKE EACH FACE A LIVING NOTE

To experiment with a more playful approach to reading in performance I developed a new piece entitled *Make each face a living note.* This piece was developed for other musicians to perform, rather than for me to perform myself. So even though the three terms arose from my thinking about being a composer-performer, they were now applied to the more conventional situation of a composer creating a performance for other musicians.

In *Make each face a living note* a large white bouncy castle is presented as though it is a musical score that is to be read and performed live in a participatory outdoor performance by an assembled group of brass players. The people bouncing on the bouncy castle, who could be considered the 'audience', are read as musical notation, and a 5-line musical stave is held in front of them using thick coloured rope. Passers-by are invited to become a unique part of an ever-changing musical score by bouncing on the castle as the musicians interpret their heads as musical notes in real-time. The event begins with a soloist, and gradually the other musicians join until the piece finishes with a full ensemble playing together, whilst two performers holding the 5-line stave move it up and down to suggest that the players should alter their register. By presenting the audience as the notation to be read by the players in real time the piece challenges the traditions of what a musical and choreographic performance can be.

In video and audio documentation of the piece the people on the bouncy castle can be seen and heard reacting to the instrumentalists by bouncing higher to encourage the playing of higher pitches or seat-dropping to elicit sudden low notes.[3] I noticed that I did this myself when I joined in on the castle, participating as an audience member. During the performance the people on the bouncy castle have the opportunity to develop a personal connection with the players in front of them, which is in contrast to interactive sound installations such as *LINES* (2016) by Anders Lind [14], which provide a situation akin to a large musical instrument that can be played by an audience in their own time. Although such interactive sound installations offer a highly satisfying musical experience, there is no connection to a live human musician, and it is this connection that marks out musical performance as distinct from sound installation in terms of the type of interactivity available.

A score for the piece, shown in figure 7, was made retroactively and is presented in the style of a children's colouring sheet with empty speech bubbles that leave room for different groups of musicians to significantly reinterpret the performance. Only the framework of bouncers being read as notation is maintained, with suggestions as to how

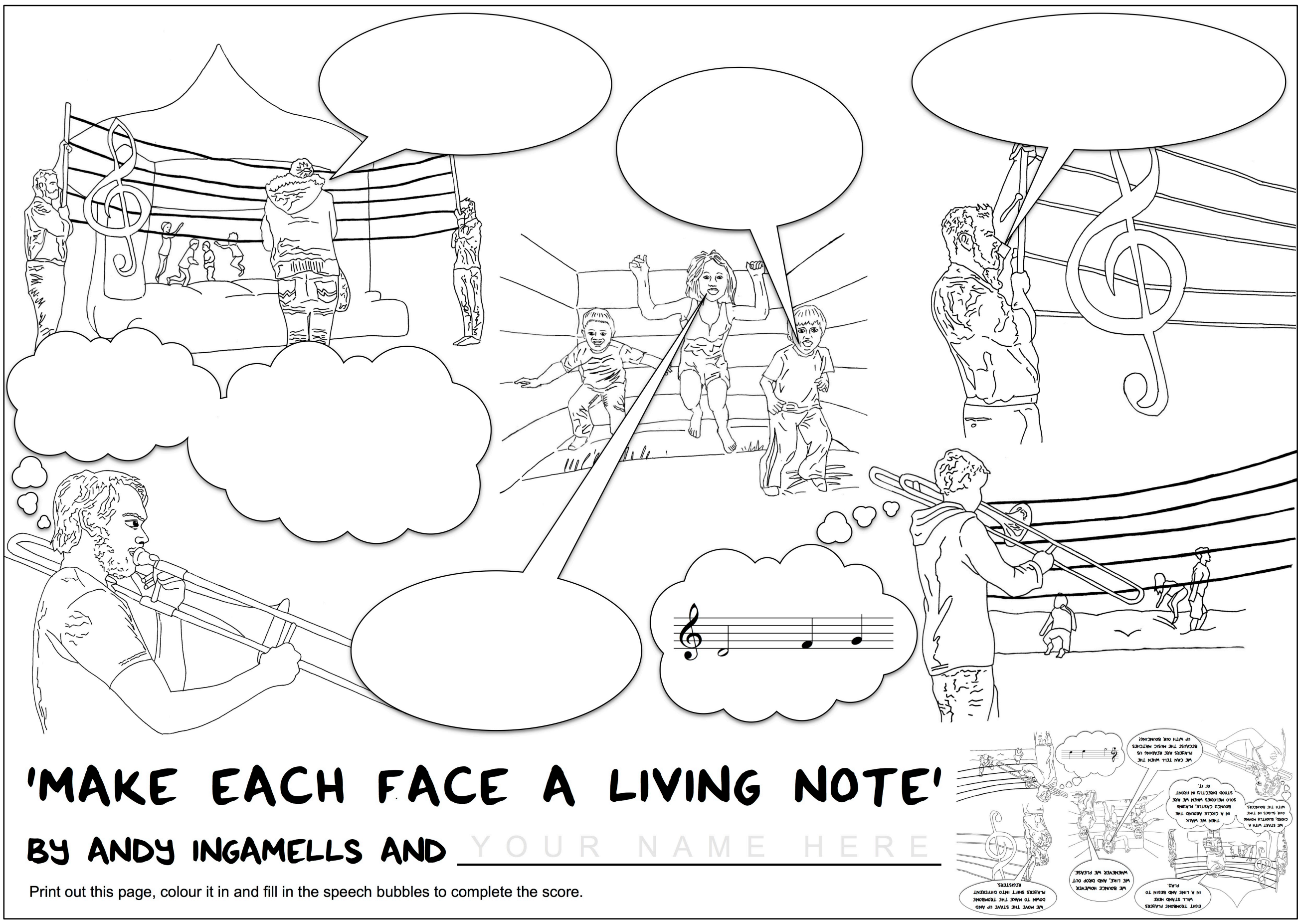


Figure 8. Open score for *Make each face a living note*, created following the first performance (2018).

[3] Video extract showing a small part of the first performance of *Make each face a living note*: https://vimeo.com/299722995

the piece can be staged given in the bottom right-hand corner based upon the outcomes of the first performance. In this way the colouring sheet serves as an outline for a situation in which musical sounds may occur, but does not prescribe any precise musical activity such as melody, harmony or rhythm beyond a very brief example.

7. CONCLUSIONS

The creative works presented in this paper are practical examples of ways in which notation can be reframed to become an integral part of the physical theatre of a musical performance, rather than something hidden behind a music stand or completely removed through memorization. This act of reframing is presented as part of a process of reimagining relationships within musical performances, rather than in the context of technical innovation or novelty. In these works, the act of reading is integral to the theatre of musical performance. When explored alone this approach has the potential to give insight into the compositional process, as in *Waschen*, and when combined with a playful improvisatory approach it can enable audiences to interact directly with the musical outcome, as in *Make each face a living note*. Making the act of reading manifest within performance establishes the possibility of notation being enjoyed for its performative aesthetics, as in *Shh*. An interdisciplinary compositional approach such this, which blurs the boundary between the roles of composer, performer and listener, can expand the creative possibilities of classical music performance in unusual and unexpected ways. Such a practice engenders active relationships between composer, performer and listener, rather than passive ones. It can also open up the myriad possibilities that notated music offers to people who might not be classically-trained and have little access to the aesthetic experience of reading musical notation. If, as Christopher Small says, notation is the medium through which creativity takes place in classical music, then the act of making, reading and interpreting notation should be celebrated directly in performance.

8. REFERENCES

[1] C. Sun. (2012, Jul. 10). "Experimental music", *Grove Music Online*. [Online] Available: https://doi.org/10.1093/gmo/9781561592630.article.A2224296.

[2] P. Dickinson (Ed.), *CageTalk: Dialogues with and about John Cage*, 2nd Edition. Rochester: University Rochester Press, 2014.

[3] C. Cardew, *Treatise*. London: Hinrichsen Edition Ltd, 1967.

[4] D. Pritchard. (2016). "Kandinsky Studies", *Creative: Deborah Pritchard on her piece for Sonic Visions*. [Online] Available: http://www.thehermesexperiment.com/creative-deborah-pritchard-on-her-piece-for-sonic-visions/.

[5] G. D. Barrett, *After Sound: Toward a Critical Music*. London: Bloomsbury, 2016.

[6] A. Harper. (2009, Dec. 17). "What is a [classical] Composer?"*, Rouges Foam.* [Online]. Available: http://rougesfoam.blogspot.ie/2009/12/what-is-classical-composer.html.

[7] G. Hare. (2015, Oct.). "Experimental music concerts return to Columbia", *University of South Carolina*. [Online] Available: http://www.sc.edu/uofsc/announcements/2015/ experimental_music_concerts.php#.Wblgu63MyRu.

[8] C. Small, *Music of the Common Tongue*, 2nd Edition. London: Calder Publications Ltd, 1994.

[9] G. Sullivan, *Art Practice as Research: Inquiry in Visual Arts*, 2nd Edition. London: SAGE Publications Ltd, 2010.

[10] Tenser, M. (Ed.). (2004). *By Alison Knowles*. [Online] Available: http://artype.de/Sammlung/pdf/knowles_by.pdf.

[11] K. Friedman, O. Smith, L. Sawchyn (Eds.). (2002). *The Fluxus Performance Workbook*. [Online]. Available: https://www.deluxxe.com/beat/fluxus-workbook.pdf.

[12] H. Higgins, "Dead Mannequin Walking: Fluxus and the Politics of Reception", in A. Jones, A. Heathfield (Eds.), *Perform, Repeat, Record*. Bristol: Intellect, 2012.

[13] D. Daniels (Ed.), *George Brecht – Notebooks I-II-III*. Cologne: Verlag der Buchhandlung Walther König, 1991.

[14] A. Lind. (2016, Sept.). "LINES – Interactive Sound Art Exhibition", *Cycling74.* [Online]. Available: https://cycling74.com/projects/lines-interactive-sound-art-exhibition/

PALACE64: IMPOSSIBLE ROLLER COASTERS AS SCORES FOR EXPERIMENTAL MUSIC PERFORMANCE

Luke Nickel
Independent
Bristol, UK
lukejnickel@gmail.com

ABSTRACT

In *Palace64*, a major new multidisciplinary project combining video and chamber ensemble, I examine the ways in which the domains of music composition and virtual roller coaster design might influence one another. First, I briefly discuss existing artistic projects relating to roller coaster design. Second, I present my own early artistic explorations combining music and virtual roller coasters. Finally, I discuss *Palace64*. In this project, I create a new medium of score that combines oral transmissions describing imaginary impossible roller coasters with videos created using innovative 3D roller coaster design software (NoLimits 2). Using strategies pioneered by Éliane Radigue and Jennifer Walshe for interpreting imagined images and paths as musical material, I develop ways in which performers can "read" these impossible roller coasters—remembered and virtual—as scores. Ultimately through this project my goal is to create and demonstrate a hybrid artwork that exists not only as a score to facilitate the performance of experimental music, but also as a conceptual theme park ride that traverses the boundaries of possibility and impossibility in a region that marries the digital with an embodied human experience of risk and pleasure. This paper is intended as an accompaniment to the performance of *Palace64* by Decibel New Music Ensemble.

1. AERIAL VIEW

Roller coasters are powerful icons, looming in our subconscious minds and permeating popular culture. [1] The domains of virtual roller coaster design and music composition have been knotted together for most of my life. As there have been few examinations of virtual roller coasters as an artistic material, I will begin this paper by providing a brief survey of major artworks on the topic. Then, I will return to a few personal *loci* that highlight the potentially fruitful bond between roller coasters and music. Finally, I will conclude with a discussion of my current project, *Palace64*. Specifically, I will examine the way that impossible roller coasters—both imagined and virtual—can be read as musical scores by combining strategies of mapping with the broader interpretational possibilities of verbal scores.

2. ARTISTIC CONTEXT

Artistically, roller coasters are often explored in terms of their relationship to danger and death. Most notably, Julijonas Urbonas' *Euthanasia Coaster* presents a hypothetical theme park attraction designed to slowly kill passengers using extreme g-forces. [2] Similarly, YouTube user No1WillWatchThis has created an extensive portfolio of rides that highlight the technical limits of *Roller Coaster Tycoon 2*, particularly as related to duration and death. [1]

Till Nowak's *The Centrifuge Brain Project* shows a fantastical futuristic theme park where the monstrous scale of the rides is intended to stimulate brain activity. Nowak ascribes a narrative to his project, creating a fictional future context for his attractions. [3] By contrast, Urbonas describes his work as a "design fiction" that

> "does not say anything itself about the settings (historical moment in time, geographic location), ethics, institutionalising, legal issues, etc. Presenting itself in such a minimalistic way, it reveals itself as a script proposal (of the usage) or as a McGuffin object for your own story". [2]

I am compelled by this idea of a ride concept that allows the audience to drape their own narrative over the work.

Real-world theme park designers are currently exploring combinations of the virtual and the real. Roller coasters that integrate VR headsets have been shown to increase people's desire to ride roller coasters and visit theme parks. [4, 5] Roller coasters are now being draped with *virtual* theming, mixing the possible with the impossible.

3. PERSONAL CONTEXT

In my experience, roller coasters are not only a site of collective cultural nostalgia but also of personal creative nostalgia. I began designing roller coasters when I was eleven years old. At the time, I lived in Manitoba, a province where the only permanent theme park was intended for children. I interfaced creatively with simulated roller coasters before I ever had the privilege of riding any real ones. I began designing roller coasters using Roller Coaster Tycoon (1999), but quickly graduated to a more

[1] YouTube user channel No1WillWatchThis. Available: https://www.youtube.com/watch?v=CFVm5R_dxoo

complex 3D simulator, NoLimits (2001). I improved my technical skills by taking part in a cycle of feedback and development made possible by various online forums and cyber peers.

After a few short years, I abandoned roller coaster design in favour of composing music. I began learning to compose music in much the same way I had learned to create roller coasters: by collaborating with other online users and iteratively testing concepts.

In 2017, after a fifteen-year hiatus in designing roller coasters, I began to wonder whether aspects of simulated roller coasters could be mapped onto or translated into musical material. During a brief residency at Arnolfini in Bristol, UK in April 2017, I tried to build a realistic simulated roller coaster—defined by my memories of design as a young teenager and by current designers' maxims found on online forums[2]. I quickly realized that the same community of users was still active after fifteen years, rendering the standards of realism and sophistication much higher than I could achieve.

At the end of my residency, I asked double bassist Dominic Lash to experiment with performing my (failed) realistic roller coaster design as a score. We identified potential variables to be mapped onto each other. For example, a roller coaster's height, speed, and g-forces might correspond to a musician's pitch, volume, and bow-pressure respectively. The roller coaster's predictable physical layout can be seen in Figure 1. The slowly undulating rises and falls resulted in a ridiculous glissando played on the double bass.

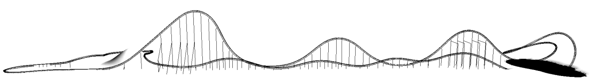


Figure 1. My realistic roller coaster created using NoLimits 2.

Dominic Lash asked me whether it would be possible to make a roller coaster that defied realistic building specifications. With a few clicks, I created a complex tangle of track, seen in Figure 2. Lash commented that this new impossible roller coaster was filled with musical potential. Unknowingly, I had created a surface texture, rather than a discrete entity that was traceable by the eye.

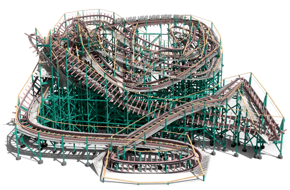


Figure 2. My first impossible roller coaster created using NoLimits 2.

[2] In addition to the Steam community, popular forums include https://nolimitscentral.com/forum https://www.coastercrazy.com/forums/ and https://coasterforce.com/forums/forums/nolimits-projects.8/

This interaction highlights the creative potential made possible through cross-talk between the domains of music and roller coasters. What once seemed like an undesirable result—an impossible roller coaster—became a site for expanded possibilities.

4. ROLLER COASTERS, MUSIC, AND SURFACES

4.1 Surface Intensity

Perhaps because of their relative newness, roller coasters lack the rigorous critical analysis of other disciplines, such as music. Many analyses consist of either reports of physical parameters—speed, height and g-forces—or prose descriptions of a rider's experience. Industry specialist and theorist Jeremy Thompson has attempted to create a system for roller coaster analysis that represents roller coaster elements using graphic notation designed to represent time, gesture, and intensity. He writes that this system is ultimately interpretive—an art, rather than a science. [7] In Figure 3, seen below, roller coaster elements are notated by letter in order to group similar elements together. The Y axis represents intensity, while small lines above the letters indicate "inflections" such as inversions, curvature, and hills.

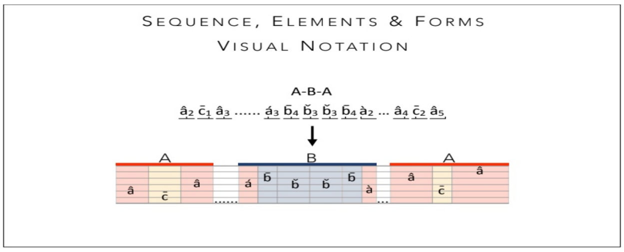


Figure 3. Roller coaster sequencing theory, credited to Jeremy Thompson / Thinkwell Group.

The intensity of roller coasters as mapped by Thompson may look similar to graphic representations of the form of many pieces in the *classical* music canon; however, gesture and intensity are concepts that have changed vastly over the course of learned eurological composition[3]. There exist many pieces that dwell at a similar intensity for their entire duration—such as Satie's *Vexations*—or musical practices outside this eurological tradition that relate to intensity in completely different ways. Absent of the demands of realism, virtual roller coasters can absorb some of this mutability. What might a non-gestural roller coaster look like?

4.2 Surface Tension

When designing impossible roller coasters, I continually return to the figure of a knot. The knot fascinates me be-

[3] Here I refer to Sandeep Bhagwati's observation that conceptual traditions often called Western Art Music are found in practices not dependent on the geographical location or lineage of the participants. [8]

cause it is an entanglement of threads. Tim Ingold categorises lines as representations or etched marks and threads as malleable materials. [8] The knotted roller coaster becomes a surface because it transforms the lines of a track—demarcations of force—into threads to be woven together. The roller coaster's clear path becomes intangible.

Much like Dominic Lash, I think an impossibly woven tangle has more potential for musical interpretation than the hills, valleys, and loose knots of real roller coasters. A surface has more variables than a line, such as density, depth, weave, direction, orientation, and texture.

4.3 Warp and Weave

In *NoLimits 2*, it is not practical to manually create a densely woven track. The editor interface consists of vertices and curves, rather than lines, and each vertex relates to the one before in increasingly complex ways. My initial explorations into hand-building tangles resulted in medium-density tracks that still somehow look realistic, as seen in Figure 4.

Figure 4. Aerial screenshot of *Moonriver32*, my first completely hand-built and themed tangle track.

Using advanced features in the program, it is possible to export and import track splines as CSV files. The resultant data consists of nine parameters, including most notably the spline's X,Y, and Z coordinates. Table 1 shows the CSV of a five-meter section of straight track.

No.	PosX	PosY	PosZ
1	0	0	0
2	0	0	-1
3	0	0	-2
4	0	0	-3
5	0	0	-4

Table 1. CSV X,Y, and Z values for a five-meter section of straight track.

In this table, I have omitted the other six columns for ease of presentation, which relate to variables such as lateral banking of the track. NoLimits allows the user to choose how often the coordinates are plotted. In this case, each point represents one meter. Figure 5 shows the corresponding track in the NoLimits editor.

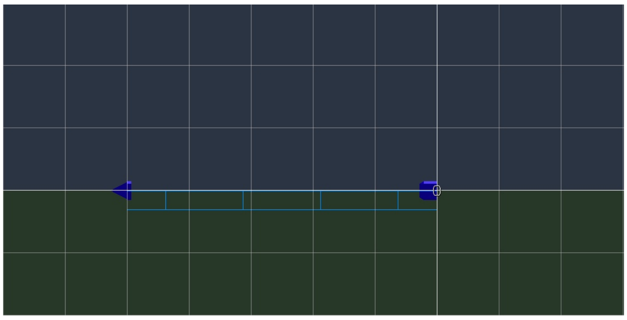


Figure 5. A five-meter section of straight track in NoLimits 2 editor, side view.

This import/export process enables users to create an element in NoLimits, export it as numerical data, manipulate it, and import it again as track. Other designers generally use this feature to create track in custom-built software that allows for more realistic physics or mathematically precise building. I am more interested in the capabilities of this feature in terms of algorithmic generation.

The following figures are two examples of tracks generated with different formulae. Figure 6 shows a track where I randomized every XYZ coordinate within a range of 150 meters.

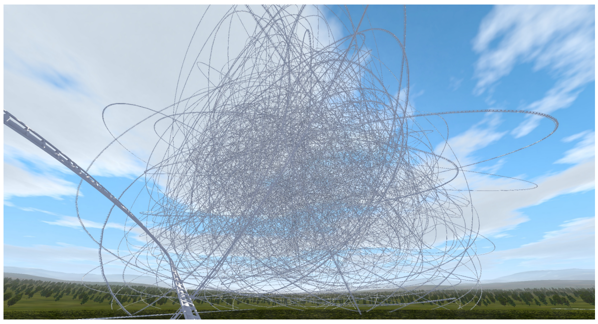


Figure 6. Track consisting of points randomized within 150 meters.

Figure 7. demonstrates a track where I hand-built a hill, then used simple successive random deviations within five meters to gradually transform the original element over many iterations.

Figure 7. Track created with a formula for random iterative deviation.

The result is an upward spiralling knot of track that comes much closer to a woven surface than any of my hand-built attempts.

5. IMPOSSIBLE ROLLER COASTERS AS SCORES: MEMORY AND VIRTUAL REPRESENTATION

In my own use of impossible roller coasters as scores, I seek to unite senses of nostalgia and possibility. Like Urbonas, I aim to create a design fiction—a roller coaster separated from the real world and full of possibilities for draped narratives.

The medium of the score for *Palace64* is twofold. The first is a private set of oral transmissions. The second is a video that serves as both an accompaniment and a score. My reasons for folding these types of transmissions together is partially conceptual—inspired by a roller coaster's connection to physics and collective memory—and partially practical. Decibel New Music Ensemble, with whom I am making the piece, is an ensemble uniquely suited to performing both orally-transmitted and video scores. Not only have they pioneered new animated score-reading technology through the creation of the Decibel ScorePlayer app, but they have also commissioned and performed a major orally-transmitted work by Éliane Radigue. My own work also traverses these two modes of transmission, making this piece the ideal opportunity to combine them.

Palace64 has three stages of development. In the first stage, I transmit verbal instructions of myself speaking about an impossible roller coaster to individual members of the ensemble.[4] I describe parameters such as the roller coaster's height, texture, speed, and direction. For example:

> *The roller coaster is a knotted spiral to heaven. It starts on the ground, but forms a tangled nest of tracks climbing upward. Cars travel via magnetic propulsion systems up the spiral, grinding and colliding with metal the whole way. This roller coaster isn't safe for humans. This roller coaster carries no-one. The speed is breakneck, but slows as the cars approach the summit of the spiral.*

Each member of the ensemble receives the same information. This first stage is an exercise in visualisation. Musicians build an impossible roller coaster in their mind. This process is similar to Éliane Radigue's collaborative compositional method utilized in her *Occam Ocean* series. Radigue asks musicians to visualise personally-familiar bodies of water to guide their performances, not only in terms of formal development but also in terms of timbre and technique. [11]

In the second stage of development, I transmit verbal instructions of myself speaking to convey the relationship between characteristics of the imagined roller coasters and musical parameters. Jennifer Walshe's *THIS IS WHY PEOPLE O.D. ON PILLS/AND JUMP FROM THE GOLDENGATE BRIDGE* points to certain strategies to avoid obvious mappings between physical paths and music. In the score, Walshe asks performers to learn to skateboard, and to memorize a favourite skateboarding path. In the performance, the musician(s) visualize skating along this path and, using one pitch as a drone, convey the surface texture, speed of skating, and shifts in muscle as timbral variations. [12] The performer(s) must use embodied memory and mapping to transform a single line of drone music.

In my piece, I take a similar approach. I ask the performers of Decibel to map their breath and bow lengths to their imagined roller coaster tracks' torsion, amplitude to speed, brightness to height, and noise-to-pitch ratio to the surface density of track.

In the third stage, I create an algorithmically-generated impossible roller coaster that focuses on knots, tangles, and surfaces. I drape this track with audiovisual theming, such as distorted sound effects from the NoLimits simulator and photogrammetrically-scanned 3D objects evoking memory and nostalgia. I film this roller coaster using screen capture technology and edit together a fixed video that serves both as score and accompaniment to the ensemble.

I ask the musicians to blend their orally-transmitted and embodied imaginings of impossible roller coasters with a reading of my video. Specifically, the musicians respond to different types of footage as cues to change their relationship to the video. While the video shows on-ride footage, the performers respond to the roller coaster on screen with the mapping strategies discussed earlier. This results in a homogenous ensemble texture. When the screen shows fly-by shots or abstractions of the footage, the performers return to their own remembered roller coasters, resulting in a more heterogeneous texture. I allow the video to oscillate between representation and surface[5], score and accompaniment.

6. CONCLUSION

Roller coasters are complex sites of nostalgia and creation. Their tracks are draped with layers of physical anticipation

[4] For a more detailed account of my oral composition process, please refer to publications by Jennie Gottschalk [9] and Tim Rutherford-Johnson [10].

[5] Playing videos as scores creates an exceptional relationship with the way musicians view screens. Schonig describes the tendency of film theory to link camera movement to an embodied notion of viewing—the camera represents our eye. He uses a number of examples to show film where instead the video becomes a surface to be read, pointing to a twofoldness of the mobile frame. [13] Video scores, both animated and made up of camera footage, require musicians to traverse this duality. These scores at once require musicians to "see" what is represented on the screen, but also to see it as representational and to translate it to embodied gesture.

and danger, maps (both as objects to be mapped and themselves a map of force and trajectory), and narratives (personal and imposed by theme park designers). Along the winding tracks are various loci, both birthed and forged—individual and collective—making roller coasters a productive tool for remembering and for imagining new futures. Roller coasters may even be an essential tool for teaching us about our own humanity. [13]

The domains of music composition and roller coasters have much to learn from each other. Further directions for this project might include: impossible roller coasters focusing on figures other than the knot; virtual roller coasters that become musical instruments through links to programs such as MAX/MSP; and tangential artworks, such as custom soundtracks for real-world roller coasters and 3D-printed sculptures.

Acknowledgments

In researching and creating this project, I acknowledge the support of the Canada Council for the Arts.

7. REFERENCES

[1] D. Anderson, "Sign, Space, and Story: Roller Coasters and the Evolution of a Thrill," *The Journal of Popular Culture*, vol. 33, no. 2, 1999, pp. 1–22.

[2] J. Urbonas, "Designing Death: G-design, Fatal Aesthetics, and Social Science Fiction" originally published in *The Edge of Our Thinking,* edited by Florian A. Schmidt. Royal College of Art, London, 2012, pp 32-46. Accessed online. Available: http://julijonasurbonas.lt/designing-death/

[3] C. Jobson. The Centrifuge Brain Project. [Online] Available: https://www.thisiscolossal.com/2013/02/the-centrifuge-brain-project-a-documentary-about-impossible-amusement-rides/

[4] W. Wei, R. Qi, and L. Zhang, "Effects of virtual reality on theme park visitors' experience and behaviors: A presence perspective," *Tourism Management*, vol. 71, 2019, pp. 282–293.

[5] T. Jung, M. C. tom Dieck, P. Rauschnabel, M. Ascenção, P. Tuominen, and T. Moilanen, "Functional, Hedonic or Social? Exploring Antecedents and Consequences of Virtual Reality Rollercoaster Usage," in *Augmented Reality and Virtual Reality: Empowering Human, Place and Business*, T. Jung and M. C. tom Dieck, Eds. Cham: Springer International Publishing, pp. 247–258, 2018.

[6] J. Lanza, "Female Rollercoasters: (And Other Virtual Vortices)," *Performing Arts Journal*, vol. 14, no. 2, 1992, pp. 51–63.

[7] J. K. Thompson, "Roller Coaster Sequencing Theory," *Roller Coaster Philosophy*, 10-Nov-2012. [Online]. Available: https://www.rollercoasterphilosophy.com/philosophy/sequencing-theory/.

[8] S. Bhagwati, "Vexations of Ephemerality," in *Proceedings of the International Conference on New Tools for Music Notation and Representation (TENOR'17)*, Montreal, 2017.

[9] J. Gottschalk, *Experimental Music Since 1970*. New York ; London: Bloomsbury Academic, 2016.

[10] T. Rutherford-Johnson, "Unsettling Scores," *The Wire*, 403, Sep. 2017, pp30-33.

[11] L. Nickel, "Occam Notions: Collaboration and the Performer's Perspective in Éliane Radigue's Occam Ocean," *Tempo*, vol. 70, no. 275, pp. 22–35, 2016.

[12] J. Walshe, *THIS IS WHY PEOPLE O.D. ON PILLS/AND JUMP FROM THE GOLDENGATE BRIDGE.* Unpublished, 2004. Available: https://www.cmc.ie/music/why-people-od-pillsand-jump-golden-gate-bridge

[13] D. Anderson and M. Burt, "Screaming on a Ride to Nowhere: What Roller Coasters Teach Us About Being Human," in *Entertainment Values: How do we Assess Entertainment and Why does it Matter?*, S. Harrington, Ed. London: Palgrave Macmillan UK, 2017, pp. 41–58.