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 InScore [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.

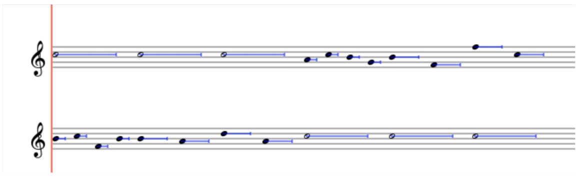

Figure 2. Same score after applying proportional notation. The default hold times indicated by the blue lines are set to 80% of the events' nominal duration.

```
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).

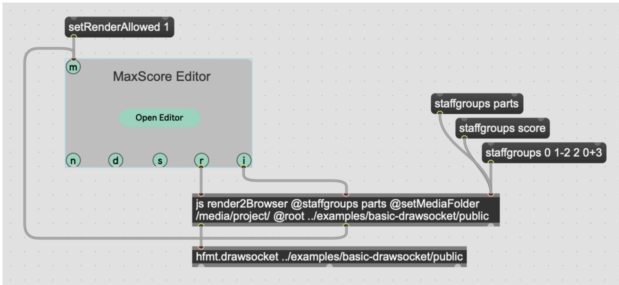

Figure 4. Max patch featuring the render2browser.js object as well as the hfmt.drawsocket abstraction containing the node.js script in charge of communicating with the clients.

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.

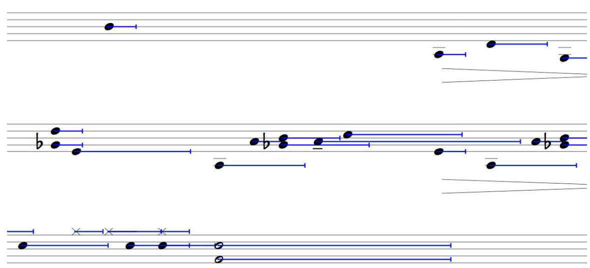

Figure 5. Excerpt from Raindrops Keep Falling (2018) for clarinet, cello, drum set and multimedia.

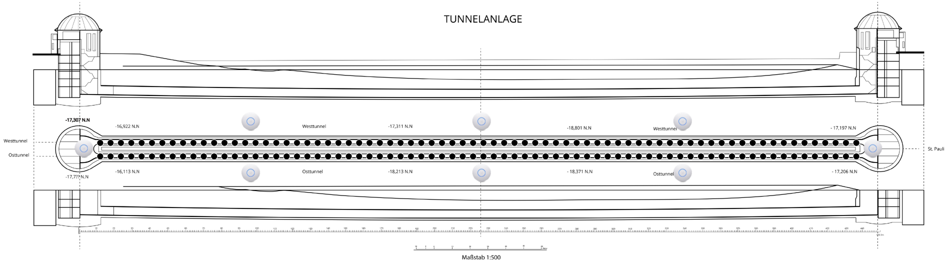

Figure 6. Cross section and top view of the Old Elbe Tunnel. Eight access points will be spaced at regular distances, each providing coverage for 18 players.

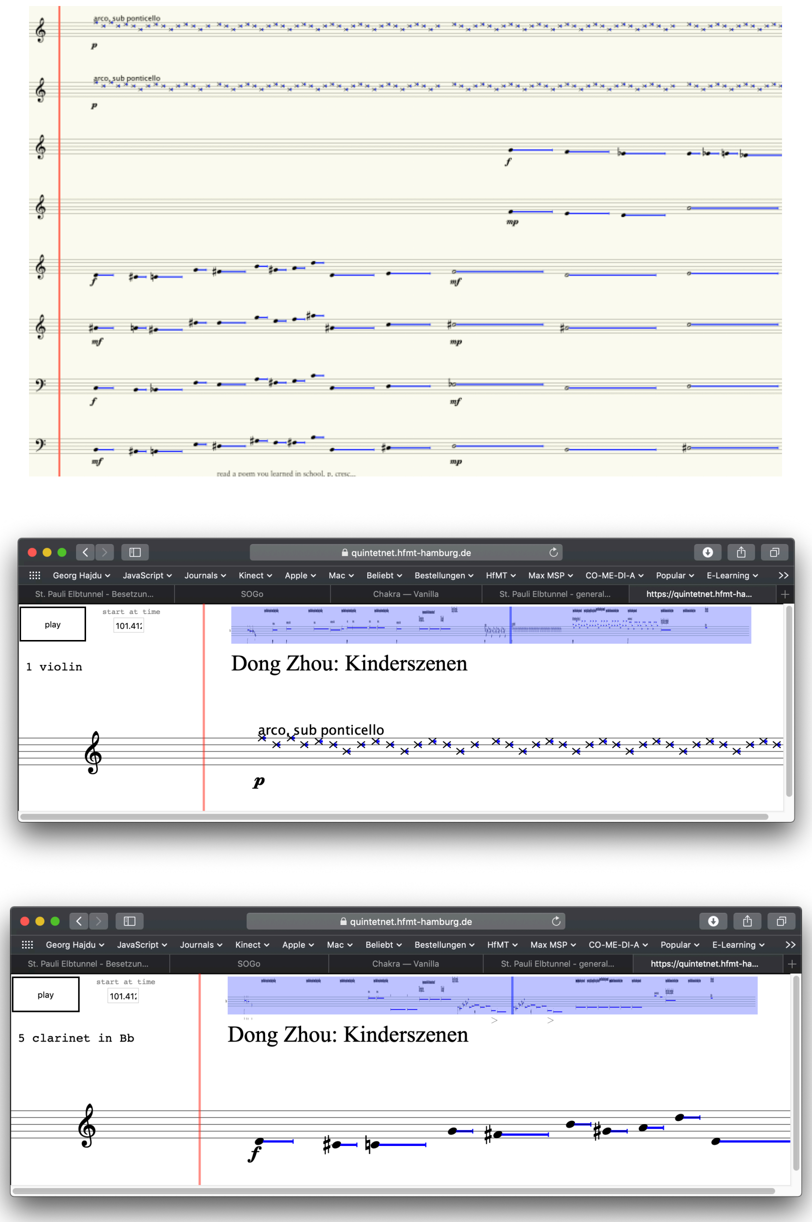

Figure 7. A score by HfMT alumna Dong Zhou showing 8 of 144 staves (top) rendered dynamically in separate browser windows (center and bottom). The staves are split by render2Browser.js. The interface elements developed for practicing at home are being added programmatically in a separate step while interface 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.