REALTIME WEB TECHNOLOGIES IN THE NETWORKED PERFORMANCE ENVIRONMENT

Rob Canning

Department of Music
Goldsmiths, University of London, UK
rob@kiben.net

ABSTRACT

Current real-time web technologies are capable of providing composers of network based music with a new infrastructure for the distribution, control and synchronisation of the networked score. When these technologies are combined with an adherence to web standards it is possible to remove software and platform specific solutions. Cross-platform web browsers present the composer with a standardised and accessible environment where notational material can be presented to performers.

These technologies provide a useful set of tools and processes to facilitate networked performance. The use of standards compliant systems over proprietary ones promotes interoperability and future proofing and provides a platform for shared reseach. These systems can also reduce rehearsal/performance set-up time and complexities to a fraction of that currently experienced with more ad-hoc scenarios commonly implemented in this area of performance.

This paper argues for the feasibility of how various Web standards, including the Hypertext Markup Language Revision 5 (HTML5), Scalable Vector Graphics (SVG) and the ECMAScript programming language can combine with real-time web server technologies, NodeJS and WebSockets in one possible work-flow as implemented in the author's *NodeScore* system.

1. BACKGROUND

The pervasiveness of network enabled mobile devices, in the western cultural context, from smart phones to tablets and laptops, alongside widespread high speed network access, in recent years has led to an increase in creative musical experiments using these technologies[13]. There has been an exponential growth of the laptop ensembles, many following on the model established by the Princeton Laptop Orchestra (PLOrk)[17] followed now by the incorporation of the smart-phone in to the performance environment as with the Stanford Mobile Phone Orchestra (MoPhO). As with their predecessors (groups such as the League of Automatic Music Composers and The HUB[7]) these ensembles often create their own ad-hoc strategies and softwares for interaction. L2ORK for example, [14] have created their own fork of the Pure Data softare as a platform for both message passing, sound processing/synthesis and networking.

The Laptop Orchestra model of treating the computer as a meta-instrument, utilising its abilities as sound-processor and input for Human Interface Devices (HIDs) has a very different set of demands than for that of the networked score. Common practice amongst composers working with networked score systems involve the use of a generalised audio oriented data-flow programming language, with plug-ins to deal with networking (often Open Sound Control (OSC)) and the presentation of notation (MaxScore, Java Music Specification Language (JMSL), OpenMusic or Lilypond based). Visual programming languages such as Pure Data or MaxMSP are the lingua franca of many computer musicians, and when it comes to finding strategies for the presentation of notational material to instrumental musicians, composers will very often use these types of tools as they are familiar and flexible tools (for example in Gerhard Winkler's Real-time scores [18] or Georg Hajdu's *Quintet.net*[8].)

These systems can, and do work, but this kind of approach is hampered by problems. Multiple, non standard installations of expensive platform specific softwares and associated plug-ins (often requiring expensive hardware), difficulties with networking; Network Address Translation (NAT) transversal and the circumvention of firewall restrictions, Network Time Protocol (NTP) synchronisation between clients as well as issues surrounding the incorporation of middleware to enable communications with some HIDs, are examples of some common problems. It suffices to say that many of these configurations are non-trivial to implement and require a technician with considerable understanding of both the specific software and network protocols being used.[1] In this context modularisation is the best way forward and the separation of the score interface from tools designed for the creation of new electronic instruments is an important step.

[1] The GRid-ENabled Deployment for Laptop orchestras (GRENDL) project [3] aims to eliminate many of the problems associated with the practicalities of the performance and organisation of pieces for laptop orchestras.

2. WEB BROWSER AS PLATFORM FOR THE NETWORKED MUSIC SCORE

The ubiquity of web browsers and document formats has led to the development of web based applications (WebApps) that previously would have existed as platform specific, stand-alone applications. The WebApp boom has resulted in the development of a range of associated technologies, providing a rich set of software tools that can be re-purposed for the creation of networked 'screen scores'[11] or 'net scores'. The network is the pertinent feature, not the screen; the content pushed through the network is not limited to that which can be displayed, for example haptic or audio driven cueing systems can be imagined. The HTML5 standard presents a rich set of presentational possibilities from standard plain text and image to high definition video, multichannel audio, animated SVG and Web Graphics Library (WebGL) 3D environments. By combining these with CSS3 and ECMAScript a new level of control over and interaction with these media assets can be obtained. The standards compliant web browser is becoming a powerful cross platform, cross device, rich web client, and can harness this power without the need of third party plug-ins such as Shockwave, Flash or Java. The use of non-standards compliant technologies can result in a relatively short life span for platforms and compositions built upon them (Duckworth's shockwave based *PitchWeb*[4] component to the web based *Cathedral* composition which no longer functions in modern browsers [5]). The standards compliant browser provides an ideal platform to be inhabited by a wide range of rich media HTML5 based net scores. Mike Solomon provided an example of this at ICMC2011 with his browser based score for *Norman (age 1)* [1] which utilises JavaScript to animate an SVG created with the Lilypond engraving software.

A number of uses of the web browser have been explored by composers, for example as a playback/display mechanism in the case of Solomon's work and as an interface to facilitate crowd sourcing for in Jason Freeman's *Graph Theory*, composers have also exploited their hypertextual navigational systems to create browser based open form scores[6][13].

3. THE REAL-TIME WEB

In the real-time web, the server can automatically push information to the browser (client) as it becomes available, without the need for a request. Current examples of this can be seen in Facebook and Twitter status updates, Ebay's bidding system and browser based live chat systems. This type of interaction with the web is fundamentally different from the old model, based on the one-page-at-a-time experience of HTML 1.0, which was inefficient with each change to the page necessitating a complete page refresh.

Although a web infused with these technologies is becoming more common today it is not a new technology[12]. In the mid-nineties Netscape Navigator introduced server-push technologies in two forms, a multi-part response MIME type called multipart/x-mixed-replace which can be used in Common Gateway Interface (CGI) scripts, and Java applets which allow a persistent connection via a raw Transmission Control Protocol (TCP) socket. Today, the most common way real-time features are incorporated in the web is by a collection of technologies working together; at the core are JavaScript and XML working to manipulating the Document Object Model (DOM), this approach is know as Asynchronous JavaScript and XML (AJAX). AJAX programming is based on an asynchronious sequence, the client makes a requests that is answered when a new server event occurs, this closes the connection and triggers a new request and so on. This type of implementation is known by a number of terms most commonly long polling or Comet programming. AJAX based long-polling and multipart responses are not true real-time, synchronious technologies, but are workarounds trying to circumvent problems such as browser timeouts (in the multipart response method), or are trying to emulate a persistent server-client connection through asynchronious sequences; they are also inefficient as each server event necessitates a client-server TCP handshake, slowing down the interaction.

Two current technologies are available which make synchronious connections possible; Bidirectional-streams Over Synchronous HTTP (BOSH) and WebSockets, the later becoming incorporated in the HTML5 standard[10]. For the demands of the networked score a synchronious, bidirectional connection between the server and clients is preferable. Bi-directionality provides the potential for non-hierarchical network systems and distrubuted and participatory style game pieces rather than the traditional conductor-musician hierarchy suggested by the unidirectional model, to encorporate these types of models is important as they are at the center of research in the field of network performance.

For *NodeScore*[2] (the networked score system in constant development by the author) the WebSockets protocol was chosen as the most efficient system with the greatest future potential though its incorporation in the HTML5 standard. The WebSockets Application Programming Interface (API) is a fully bidirectional, duplex protocol that communicates over a single TCP socket.1 It can be browser based and as such can use port 80, bi-passing firewall restrictions imposed by some servers and clients on non port 80 TCP connections.

4. NETWORKED PERFORMANCE AND BI-DIRECTIONAL SERVER INFRASTRUCTURE

NodeScore's primary goal is the communication of musical score material to, or between, human performers reading from displays connected over a network. It is a web based system but unlike much work [2][15][9] with web

[2] http://nodescore.kiben.net

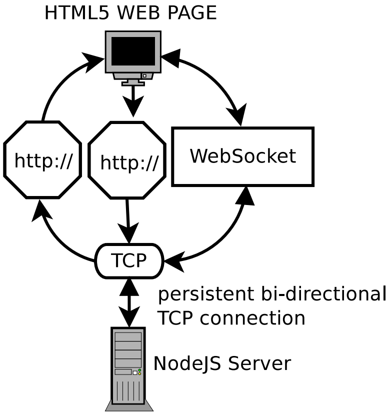

Figure 1: WebSockets

based Networked Music Performance (NMP) it does not attempt any synthesis, either distributed or cenralised.

NodeScore uses a server written using the NodeJS framework, a server-side environment for the creation of web applications, primarily web servers. It is built on Chrome's JavaScript runtime (V8), the programs for NodeJS are written in JavaScript. NodeJs allows the use of a common language on both server and client sides of the application.[3]

The combination of WebSockets, NodeJS, JavaScript and HTML5 is a flexible, standards compliant base on which to build net scores. The bi-directionality of the connections established between the multiple clients and the server allows inter-browser communications and control. In the context of the net score this opens a number of possibilities; one browser can act as a control mechanism over other connected browsers in a traditional one to many leadership hierarchy, where the ensemble director can, via their web interface, control the content displayed on any one of the connected performers' net scores.2

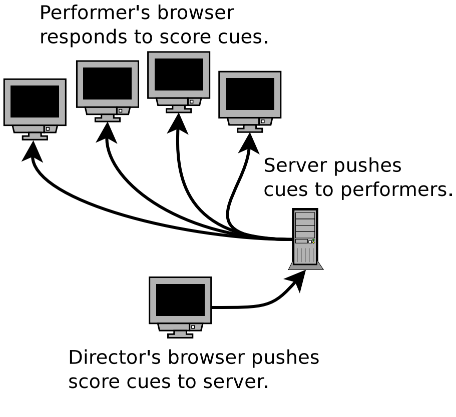

Figure 2: Traditional hierarchical, one-to-many "Directed" Model

The ensemble director's web based control interface sends data down a WebSocket to the server. This data contains an index determining who should receive the message and an index indicating which notational material should be displayed. The server filters that information and channels it back down the relevant WebSocket to the target performers interface where the material is then displayed. There is no delay while waiting for TCP handshakes to be completed as the connections between all involved are persistent, the only delay involved is the network latency which is minimal, especially over Local Area Network (LAN). As all clients are connected to a central server and messages between clients pass through that server then latency maybe normalised through a calculation of client Internet Control Message Protocol (ICMP) echo request response times with outgoing messages staggered accordingly.

The hierarchical model is not the one only available, any client has the ability to send messages to any other client (via the server).3 Nor does the server have to be aquiescent in its relay of messages within this system but can be given an identity, deciding which messages it will or won't pass or interfering with the message content. Placing a rougue agent such as a disobidient server or a "bot" within the network is an example of how creative experimentation intersects with available technologies to create unpredictibility within the channels of communication in the quest for novelty. The framework must allow for the potentialities of these more democratic or experimental structures of interaction. Rebelo has suggested three distinct relationships that may be use to categorise networked performance: projected, directed and distributed "dramaturgies". Within these dramaturgues are explored the relationship between "notions of authorship, collaboration, structure and content"[16] and it is crucial that any framework that may be usefull is capable of both synchronious and asynchronious event handling within all of these multi-nodal relationships.

The ability to exert a level of control over the relationship between the material presented and the musicians (improvising or performing within polyvalent structures) gives the composer the opportunity to experiment with the network itself. To compose with network configurations makes it possible to explore how musical materials and performers can interact to yield novel results. This process allows reflection on the emergent properties of the network configuration employed and raises questions on how emergent properties may be identified and categorised within these systems.

Most ensemble performance can be considered in the context of the network, as they explore channels of communication through listening and response between a number of agents. In the context of the western tradition of notated art music, the score can act as a barrier to inter-group communication, as performers are isolated by an individual instrumental "part". The authority on the relationships and dynamics of the system lies with the holder of the full score, the conductor, inter-ensemble communication are often reduced to the level of nuance.

Other features currently built in to the server system include a "server-push" based stopwatch and a chat client

[3] Implementations using the Apache web-server with WebSockets implemented through PHP are also possible.

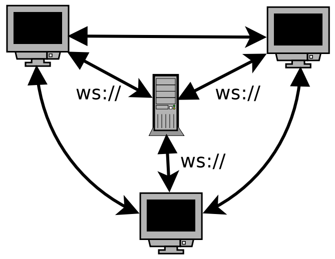

Figure 3: Non-Heirarchical multinodal implementation.

allowing all performers involved to synchronise and send message to one another. The audio and video stream can be hosted on a separate layer, and need not be integrated into the net score's server infrastructure as it is important to keep the system modular allowing different technologies to be mixed and matched as appropriate.

With real-time web processes a platform can be created to allow for a type of score that can be distributed, synchronised and controlled across a network in new and more democratic ways. Using standards compliant technologies to achieve this creates a shared platform for research and a free and accessible medium for its dissemination.

5. CONCLUSION

Network enabled scores need a unique set of tools and processes for both their creation and distribution. The real-time web combined with standards compliant document formats, provide a framework within which these tools and processes may be developed. These will be of use to both the laptop/mobile ensemble as well as the instrumental ensemble reading from the screen and will provide a flexible layer, working alongside sound processing and streaming technologies for the facilitation of telematic and LAN based networked performance.

The primary concern of *NodeScore* is to provide a framework for fast, low latency web-based rich media score delivery, control and synchronisation. This is achieved through the use of an infrastructure that aspires towards longevity and cross platform accessibility in a rapidly changing ecosystem of devices and internet technologies. Compositions using this system can explore synchronised non-linear/hypertextual ensemble performance over the network or director led improvised systems; it is however, is not limited to these experimental structures and may be used for more traditional page based or scrolling types screen scores in the context of networked performance.

Whether or not the browser will prove an enticing environment for the composer to present net scores depends largely on the development of standards compliant media types and associated authoring tools. With the recent developments in HTML5, CSS3 and ECMAScript combined with compliant media such as the animated SVG, the webm video codec and the WebGL 3D engine it does look to be a promising platform. To combine and control these media in real-time across a network shall be an important part of the process.

6. REFERENCES

[1] M. Akkermann, "ICMC 2011 Review: Concert 10, 4th August 2011," *Array, Journal of The International Computer Music Association*, p. 64, 2011.

[2] A. Barbosa, M. Kaltenbrunner, and G. Geiger, "Interface decoupled applications for geographically displaced collaboration in music," in *Proceedings of the International Computer Music Conference*, 2003.

[3] S. Beck, C. Branton, S. Maddineni, B. Ullmer, and S. Jha, "Grendl: Grid enabled deployment for laptop orchestras," *Society for Electro-Acoustic Music in the United States 2011 National Conference.*, 2011.

[4] W. Duckworth, "Making music on the web," *Leonardo Music Journal*, vol. 9, p. 13–18, 1999.

[5] ——. (2011, February) Pitchweb @ONLINE. [Online]. Available: http://pitchweb.org

[6] J. Freeman, "Web-based collaboration, live musical performance and open-form scores," *International Journal of Performance Arts and Digital Media*, vol. 6, p. 149–170, 2010.

[7] S. Gresham-Lancaster, "The aesthetics and history of the hub: The effects of changing technology on network computer music," *Leonardo Music Journal*, vol. 8, pp. 39–44, 1998.

[8] G. Hajdu, "Real-time composition and notation in network music environments," *Proceedings of the International Computer Music Conference*, 2008.

[9] J. Herrera. (2009) The horgie: Collaborative online synthesizer. [Online]. Available: https://ccrma.stanford.edu/courses/220c.../Herrera_Horgie.pdf

[10] I. Hickson. (2012) The websocket api w3c candidate recommendation 08 december 2011. [Online]. Available: http://www.w3.org/TR/websockets/

[11] C. Hope and V. Lindsay, "Screen scores: New media music manuscripts," *Proceedings of the 2011 International Computer Music Conference (ICMC)*, 2011.

[12] J. E. Kendall and K. E. Kendall, "Information delivery systems: an exploration of web pull and push technologies," 1999. [Online]. Available: http://aisel.aisnet.org/cais/vol1/iss1/14

[13] D. Kim-Boyle, "Real time generation of open form scores," in *Proceedings of Digital Art Weeks, ETH Zurich*, 2006.

[14] T. Martin, E. Standley, and M. Matthews, “Introducing l 2 ork : Linux laptop orchestra,” *Interfaces*, no. Nime, pp. 170–173, 2010.

[15] C. Ramakrishnan, J. Freeman, and K. Varnik, “The architecture of the auracle: a real-time, distributed, collaborative instrument,” in *Proceedings of the Conference on New Interfaces for Musical Expression*, 2004.

[16] P. Rebelo, “Dramaturgy in the network,” *Contemporary Music Review*, vol. 28, pp. 287–393, 2009.

[17] D. Trueman, P. Cook, S. Smallwood, and G. Wang, “Plork: Princeton laptop orchestra, year 1,” *Proceedings of the 2006 International Computer Music Conference (ICMC)*, 2006.

[18] G. E. Winkler, “The realtime-score. a missing-link in computer-music performance,” *Sound and Music Computing*, 2004.