AUTOCONDUCTOR – SYNCHRONIZING GRAPHIC SCORES, MULTI-CHANNEL SOUND AND FULL-HD PROJECTION

Lothar Felten
Freelance programmer
lothar.felten@gmail.com

Christian Klinkenberg
Composer
klinkenbergcomposition@gmail.com

ABSTRACT

In this paper, a networked score display system which synchronizes graphical notation, precise pitch notation and video with multi-channel audio is presented.
The software was purpose built for the project *The Glacier - Opera 2.0* composed by Christian Klinkenberg.

1. INTRODUCTION

Since 2016 the composer Christian Klinkenberg has been using a graphic video score for his compositions. This was the case in *Das Kreuz der Verlobten* and *The Leaves That Hung But Never Grew*. Unfortunately, the synchronization of the video score was insufficient in both cases. This was the trigger to develop a software to meet the special needs.

1.1 The previous projects

Concerning the project, The Glacier - Opera 2.0 [1] , the composer Christian Klinkenberg and the programmer Lothar Felten have jointly developed the software.
Since the first opera *Das Kreuz der Verlobten*, Christian Klinkenberg has worked with video notation. All musicians played an individual video file on their computer. The scrolling of the frames of the graphic score from right to left was performed with the transition Slide-Push in the software Adobe Premiere[1]. The conductor gave the signal and everyone pressed Start or Play on their video player. Real synchronicity was therefore by no means provided.
For the composition *The Leaves That Hung but Never Grew* [2] , a public domain software called *syncplay* [3] was used. Lothar installed the software on a server, and the musicians had to install a client version on their device. There were clients for all standard operating systems. The devices brought by the musicians where running iOS, Android and Windows, while the syncplay-server was running Linux. The advantage of syncplay was that the rehearsal work was much more comfortable than without this synchronization. As soon as a client jumped at a particular scene, all computers jumped with it. The disadvantage of this software was that the videos are only synced at start, no intermediate synchronization is applied. After the length of the entire piece, the videoscore was offset by up to 2 seconds between the slowest and the fastest device.
So, the project autoconductor was started.

1.2 Available solutions

There are multiple solutions available that provide means for scrolling through an image-based score in sync over the networks, such as *Decibel ScorePlayer* [4] [2][3] or *Quintet.net* [5] [4]. Two hard requirements for the composer's works are synchronization with a continuous video stream for the audience and the possibility to show a common graphical notation and a distinct precise notation per instrument on one screen for each musician.
Most available solutions also require a significant amount of configuration on the part of each musician. A notable exception is the *SmartVox* [6] [5] project that uses a web-based approach and hence supports a wide variety of platforms.

1.3 Goals for the new project

Parallel to the video scores, a video in mp4 with 1080p and 7.1 sound should be projected for the audience. The multi-channel sound was important because, among other things, a click-track was used, which of course should not be heard on the soundtrack of the audience.
The scores for the musicians consisted of 2 parts:

- The upper part consists of the graphic score, which remains the same for all musicians.
- The lower quarter consists of a pitch notation on staves[6].

Because of the microtonal compositional aspects, in addition to the standard notation with cent deviations[7], tablature notation and other alternative notation systems such as for the Bohlen-Pierce clarinets, the "*Mller-Hajdu notation* [7] [8][9] was used. This score was notated in four colors in the software Finale and later converted into a png-picture. The four colors are needed to link the notes to the graphic. When selecting the colors, care always had to be taken to ensure that they remained easily distinguishable from the

[1] http://www.operatheglacier.com
[2] http://www.kl-ex.com/en/projects/the-leaves-that-hung-but-never-grew/
[3] http:https://syncplay.pl
[4] http://decibelnewmusic.com/decibel-scoreplayer/
[5] http://quintet.net/
[6] https://www.bachproject.net/2016/09/07/the-smartvox-project/
[7] http://www.noralouisemuller.de/Muller_BP-Notation.pdf

background, which was also colored[10].
A precise time positioning should simplify the rehearsal work. So, it is possible to jump from one position to another.

2. TECHNICAL REQUIREMENTS

The technical requirements for *The Glacier Opera 2.0* can be summarized as a simple, lightweight tool to provide an audiovisual stream for the audience in synchronicity with the complete scores for all musicians.
Looking at the details, the complexity of the task becomes clear: The video does not only provide a stream with stereo sound for the audience but also includes multiple click-tracks for the drummer and the conductor. Also, supporting a wide variety of devices to show all the different scores in sync with the video over a wired or wireless network with an accuracy high enough to play music together is another aspect which had a significant impact on the design decisions.

2.1 Prior experience

The first collaboration was a year earlier for an interactive musical narrative in the microtonal system *Bohlen-Pierce* [8] with graphical scores. In the successful project *The Leaves That Hung but Never Grew*, we gathered initial experience with a setup where the audience was invited to play along with the musicians was gathered: At the start of the performance, everyone was invited to use their smartphone to connect to a wireless network and to use the browser to produce their own sounds. A large screen showed the images illustrating the fairy tale and included a superimposed score in the form of graphical notation. Symbols in 7 different colors, which should represent a certain musical content[11] of the graphical score were also used on the user interface shown on the smartphones of the participating listeners. This first project showed that its feasible to provide a responsive, low latency interface using browser-based technology over a wireless network. The same techniques were used as the foundation for the *autoconductor* project.

2.2 Client side

From the start it was clear that no dedicated hardware would be used the musicians would bring in their own devices and with it a wide range of hardware and operating systems. The lowest common denominator for the entire range of mobile and portable devices is probably a web browser.
An *app* would have been an option, but it would have required a cross-platform framework to support at least Windows, Android, iOS and Linux and needs to be installed beforehand.
A web-based application on the other hand, should run on every platform supporting a modern browser without any installation. Ideally, it also behaves in the same way on all systems. Also, there is little to no training required as the users know how to connect to a network and navigate to a website.

To keep the setup time for the musicians as low as possible, a single web page provides the entire content. Except for an initial drop-down box to select the score for the instrument at hand, there is no further control element on the page.
The content is provided and controlled by the server, the client-side technique used to display the scores is JavaScript and the communication is done via WebSockets.

2.3 Server side

The main design goals for the server-side were portability, ease-of-use, low latency and low cost. The opera was performed in multiple venues in Belgium and New York without a dedicated staff member for the server or network setup, hence the requirement for a system that once configured can be setup in a short amount of time. Also, the server and network gear consisted of a re-purposed laptop, a second-hand internet router and some cables – parts that can be quickly taken onto a flight. The sole purpose of the router was to provide a wifi access point and a private network, there is no internet access required. A mixed operation of wired and wireless operation is possible. The preferred method is a wired network, but recent improvements of wireless network technology make it a viable option.

Figure 1. Performance in New York without a human conductor.

The laptop runs the Linux operating system, the *nginx* [9] webserver and the main application written in C++/*Qt5* [10] . If equipped with a second screen or a projector, the video output can be shown in full-screen mode on the second screen while the first screen shows the window with control elements.

2.4 Multichannel audio

In contrast to *Das Kreuz der Verlobten*, bars and rhythm should also be possible. The visual information was not sufficient. So, in addition to the stereo sound (Gabriel's voice, radio and effects) for the audience, we needed at least one additional channel for the click. Preliminary experiments with the ensemble confirmed the assumption that the tempo is often unstable with purely visual information. The possibility of working with multi-channel sound for

[8] http://www.huygens-fokker.org/bpsite/

[9] https://nginx.org/

[10] https://www.qt.io/

future productions should also remain.
The first rehearsals with the software still contained some bugs, which were fixed step by step. For example, in the beginning, there was still very strong stutter. Reducing the update rate was the solution to this issue. Besides, it turned out that the files of the graphics in *png* [11] format were very large and caused significant load on the wifi network. In the *jpeg* [12] format, however, the pictures painted by Marc Kirschvink in 1080p with small file sizes could achieve an acceptable quality despite lossy compression.

3. REALISATION

The main concept is based on the video shown on a large screen and the scores of all musicians will be synchronized with the video. If the video is paused or positioned to a different time code, the displayed scores will scroll accordingly. The graphical score consists of a long series of images, which, when placed side by side would make up a single long score for the entire musical piece. These images scroll by at a constant speed and the currently shown position is synced to the video time code.
The video playback is controlled by an application with graphical user interface that runs on the server, see Fig. 2.

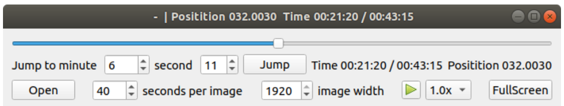

Figure 2. Video controls.

In addition to the normal controls a video player offers, there are fields labeled *image width* and *seconds per image*. Those fields specify the width of a single image of the score and the amount of time it will take to scroll by.
The total amount of images needed depends on the length of the video track, the width of the images and the number of seconds per image. Next to the time code labels, the *Position* label shows the current image and pixel associated with the current time code.
For convenience the *Jump* button was added, which repeatedly allows a quick jump back to a given time code, a useful tool for rehearsals.
For the musicians this is hidden, they only see a full screen browser and after the initial selection of the instrument a score, see Fig. 3.
Here the score consists of two parts, the top is the graphical notation and the bottom quarter is used for supplemental information, either scores or text.

4. IMPLEMENTATION DETAILS

The entire solution is implemented as a classic client server architecture. All communication is done via http and WebSockets. In contrast to classic AJAX which uses *XMLHttpRequests* [13], Websockets allow bidirectional communication between the client and the server. In our case, we make extensive use of the server to client direction.
Other research by *Augusto Ciuffoletti* [14] has shown that latencies below one second are possible over the internet. Our implementation proves that in a local, dedicated wireless network the attained latency is low enough to synchronize scores for musicians.

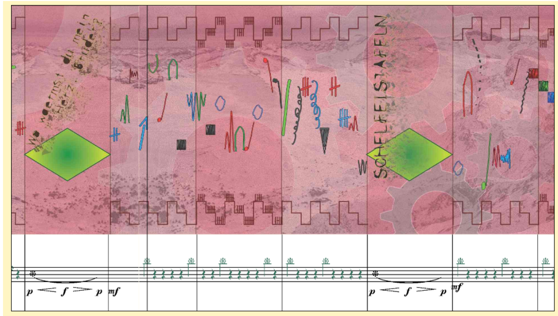

Figure 3. Musicians' score.

4.1 Server application

The autoconductor application itself is an application written in *C++* [15] using the Qt5 framework. Qt5 was chosen because of the good multimedia library, GUI toolkit and *WebSocket* [16] support. Written in plain C++ the system load caused by the application is low and it yields excellent response times required for low latency use cases. Main component is the video player with multichannel audio support, the video output is rendered in a separate window for dual screen setups: one screen for video, one for the control panel. In a separate thread there is a WebSocket server running, it keeps track of all connected clients and sends the current position of the video in regular intervals. The data transmitted over the WebSocket are formatted in *JSON* [17] and carries only the current image number and the current pixel position in the image. Its up to the client to display the correct images in the correct location, see Fig. 4.

4.2 Client-side JavaScript

The client side is a single web page, the initial view shows a drop-down box to select the instrument or track. Once selected, the drop-down box is hidden and a new HTML canvas element will use the entire view port to render the score.
All following action and communication is handled by client side JavaScript.
Immediately after the initial instrument or track selection, the JavaScript code will open a WebSocket to the server and start preloading all the JPEG images for the entire graphical

[11] https://en.wikipedia.org/wiki/Portable_Network_Graphics
[12] https://en.wikipedia.org/wiki/JPEG_File_Interchange_Format
[13] https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp
[14] https://arxiv.org/abs/1901.00724
[15] https://isocpp.org/
[16] https://developer.mozilla.org/de/docs/WebSockets
[17] https://www.json.org

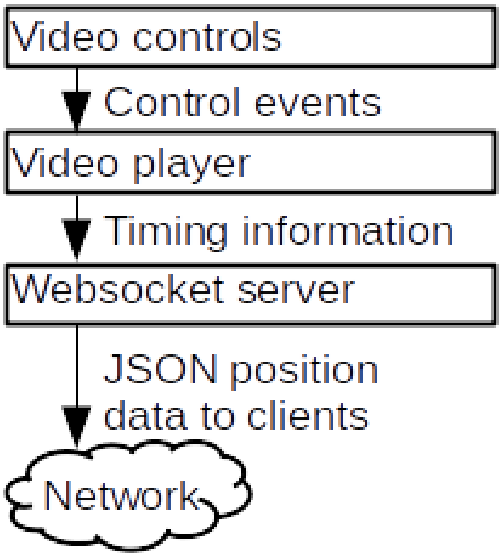

Figure 4. Server side.

notation and all PNG images for the precise notation, see Fig. 5.

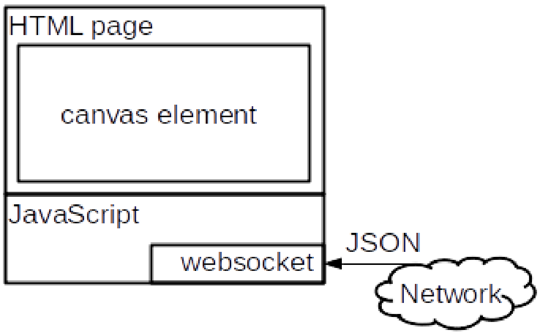

Figure 5. Client side.

The WebSocket connection to the server is opened and incoming JSON messages are parsed. The incoming data is used to display the correct images on the HTML canvas.
The initial approach was to load the images on demand but this caused spikes in network load and caused jitter on the display timing: whenever a new image was due, all connected clients issued two new requests to the webserver to load the two images needed. On average, the network load was low, but the congestion by periodic spikes caused temporary missing images on the client side. Also, some of the sync packets were lost, causing jitter in displaying the images, see Fig. 6.
Preloading the images consumes more memory on the client side and also causes peak in the network load at the initial load, but there is no additional network traffic during the performance. Even a single client reloading the entire graphical notation during the performance causes some negligible load and has no noticeable impact on the other clients. Images are preloaded in order, so if the current position is not at the very beginning, a few seconds could be missed.

Figure 6. First test with a server and 10 client devices.

Another major advantage of preloading all the images is that jumps during rehearsals are instantly: Scrolling in either direction on the central server causes the clients to show the correct position in real time.

5. POSSIBLE ENHANCEMENTS

The software was purpose build for the project *The Glacier - Opera 2.0*, but with some improvements this might be a generic tool.
Here is a short list of possible enhancements that might make the software usable for a wider audience.

- Variable speed throughout the score. Currently the score scrolls by at a fixed speed, it would be nice to have a feature that allows to set the scroll speed for each passage.
- Conductors score, one page with all parts. There is no way to display all scores on one screen at the moment. This is mainly a layout issue to fit everything on one screen in a legible manner.
- User friendly way to import and sort score images. Finding the correct images relies on filenames – each image has a dedicated filename. It would be nice to have a tool that splits the scores from a PDF and renames them automatically.
- Plugin for *MaxMSP* [18]. *MaxMSP* is quite popular, providing an interface to it would be a great addition.
- A midi-output synchronized with the score. In this way, electronic instruments, lights via *DMXIS* [19], effects or events could be controlled remotely in *Ableton* [20] or similar digital audio workstations.
- Live measurement and indication for the current latency. Some basic latency measurements have shown that a wired network is indeed preferrable over a wireless network. A display of the quantitative latency and the variance between clients might give interesting insights for future improvements and research.

[18] https://cycling74.com/
[19] https://www.enttec.co.uk/en/product/controls/dmx-lighting-control-software/dmxis/
[20] https://www.ableton.com

6. CONCLUSION

After initial difficulties on the first rehearsals, the software ran stable from the final rehearsal starting a week before the first performance. 15 devices running different operating systems logged in via wifi. Sometimes there were stutters. But this is due to the wifi network, this could be remedied with a better router. A lot of different devices from notebooks to tablets to smartphones were used, so that the Webapp system proved itself in any case. Autoconductor is going form the basis for many new compositions and performances. The software is constantly being developed further and the current version for Linux can be found at this link:`https://www.christianklinkenberg.com/research/autoconductor/`

7. REFERENCES

[1] M. Jago, *Adobe Premiere Pro CC Classroom in a Book*. San Fransisco: Adobe Press, 2017.

[2] L. V. C. Hope, *Visualising the Score: Screening Scores in Real-time Performance*. Interactive Media E-Journal, 2011.

[3] R. L. C. Hope, *The Decibel ScorePlayer - A Digital Tool for Reading Graphic Notation*. ECU Publications, 2015.

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

[5] B. M. J. Bell, "Smartvox-a web-based distributed media player as notation tool for choral practices," in *Proc. Int. Conf. on New Tools for Music Notation and Representation - TENOR 2017*, La Corua, Spain, 2017.

[6] K. Stone, *Music Notation in the Twentieth Century: A Practical Guidebook*. New York City: WW Norton & Co., 1980.

[7] E. Karkoschka, *Das Schriftbild der Neuen Musik*. Celle: Moeck, 1991.

[8] N.-L. Müller, *The Bohlen-Pierce Clarinet: Theoretical Aspects and Contemporary Applications*. Books on Demand, 2020.

[9] S. S. M. Stahnke, *1001 Mikrotne*. Von Bockel Verlag, 2015.

[10] S. F. D. Kastan, *On Color*. New Haven: Yale University Press, 1991.

[11] J. Wolfe, *Sound Musicianship – Understanding the Crafts of Music*. Cambridge Scholars Publishing, 2012.