SONIFICATION FOR EVERYONE EVERYWHERE – EVALUATING THE WEBAUDIOXML SONIFICATION TOOLKIT FOR BROWSERS

Hans Lindetorp

KMH Royal College of Music
Stockholm, Sweden
hans.lindetorp@kmh.se

Kjetil Falkenberg

KTH Royal Institute of Technology
Stockholm, Sweden
kjetil@kth.se

ABSTRACT

Creating an effective sonification is a challenging task that requires skills and knowledge on an expertise level in several disciplines. This study contributes with WebAudioXML Sonification Toolkit (WAST) that aims at reaching new groups who have not yet considered themselves to be part of the ICAD community. We have designed, built, and evaluated the toolkit by analysing ten student projects using it and conclude that WAST did meet our expectations and that it lead to students taking a deep approach to learning and successfully contributed to reaching the learning outcomes. The result indicates that WAST is both easy-to-use, highly accessible, extensively flexible and offers possibilities to share the sonification in any device's web browser simply through a web link, and without installations. We also suggest that a sonification toolkit would become an even more creative environment with virtual instruments and mixing features typically found in Digital Audio Workstations.

1. INTRODUCTION

Since the foundation of the International Community for Auditory Display (ICAD), the numerous publications have formed a substantial body of knowledge of sonification reaching readers and practitioners from far outside the immediate research field; however, there are still lots of areas to explore that could incorporate sonification as a pedagogical and didactic method in education crossdisciplinary and crosscultural. In the Sonification Handbook [1], Hermann, Hunt and Neuhoff envision a future where "sonification will be a standard method for data display and analysis" and sonification techniques are established and standardized. For this to happen, several obstacles need to be overcome. *Parameter mapping sonification* is one of the most established techniques for sonifying data and to produce a meaningful result the developer needs skills for data preparation, sound synthesis, mapping parameters and finally listening and tuning the settings [2]. The process also requires an understanding of auditory perception [3], sound design, and musical composition. There have been attempts to formalize ways of describing the relation between the original data and the auditory domain [4] but since this relationship is far from being a simple linear link [5], further research is needed.

Another challenge that has not yet been addressed extensively is how sonification applications can be distributed. If the result of a sonification was distributed at all, it has most often been delivered as a prerendered audio file rather than in an open, interactive format that can be explored, evaluated and manipulated. This in turn makes the workflow between customer and manufacturers of product sound design less iterative, potentially leading to less creative solutions in sonification applications.

All these challenges have great implications on education in the auditory display field. Learning activities typically involves exploring, interacting, collaborating, presenting, distributing and evaluating, and every contribution towards making those steps more accessible, transparent and standardized would greatly benefit the field. This could potentially lead to more engaged students, and it could encourage a broader, more interdisciplinary research community.

With this study we want to contribute to the community with a software toolkit called *WebAudioXML Sonification Toolkit* (WAST) that aims at reaching new groups of students, researchers and sound designers who have not considered themselves to be part of the ICAD community before. We have designed, built and evaluated the toolkit within a pedagogical context to better understand how to make sonification applications easy to both use, evaluate and distribute. We also want to challenge the tension between flexibility and accessibility and therefore provide a toolkit that embraces both perspectives.

2. BACKGROUND

In the following, we present review of similar works and a description of WAST. To provide the context around which WAST is built, we also briefly introduce the Web Audio API and WebAudioXML.

2.1. Accessible sonification toolkits

Toolkits and frameworks for facilitating non-experts to create and make sonifications available have been presented since the first ICAD conference in 1994. Then, the authors of Listen Specification Language (LSL) wrote [6]

> "We have designed a language that simplifies the task of specifying which occurrences during program execution are to be auralized and how. [. . .] we want LSL to be usable by programmers regardless of their preference for a given other programming language [. . .]"

In 1996, another system called Listen was introduced with the statement [7]

> "In spite of great deal of work on sonification, the need for a general-purpose data sonification toolkit, that can conveniently allow experimentation with scientific data, remains."

Later, in 1997, the same team presented a development of Listen with a more musical approach called MUSE and wrote that their main objectives were [8]

> "...to create a general purpose sonification tool, which can create musical and non-fatiguing sounds and still allow meaningful exploration of scientific data [and] provide an interactive and flexible environment for the user to map data to [different] sound parameters."

More recently, in the previous ICAD of 2019, Philips and Cabrera [9] presented both a new toolkit and a review of similar works. According to them, sonification toolkits can be placed along a spectrum spanning from flexibility to accessibility, where flexible toolkits are commonly written using general purpose languages, and toolkits with increased accessibility are written as dedicated applications. In-between these are what they call domain-specific languages, with Supercollider [10] and Pure data [11] being the most popular.

Also in their review of similar works, it was found that few sonification toolkits have seen continued development [9]. Only one of the five included studies had operable code to test, and this particular project [12] was much more narrow in scope. We could confirm this in a survey of past ICAD contributions; this indicates that there appears to be obstacles to presenting and distributing interactive sonifications to a wider public than the academic community. In our survey, we queried the Georgia Tech Library (https://smartech.gatech.edu/handle/1853/49750) with the keywords "interactive sonification", and "toolkit" or "pedagogy" to find a body of works that resembled our study presented here. (From such unrefined search, some of the found publications were less relevant, while others were presumably not pulled from the database.) The search resulted in a list of 52 publications spanning from 1994–2019. Of these, 35 studies had no links to either audiovisual examples or code, and only seven papers had working links to code or applications. This indicates that, until now, software and examples are typically not made available and accessible for general use in works presenting interactive sonifications and related tools.

From the literature search above and from reviews by other scholars [13, 14, 9] it appears that the Web Audio API is not often used in sonification toolkits. Web Audio could be considered as a domain-specific language (based on JavaScript) with a defined target, namely to be implemented in web pages for providing audio contents in browsers – and thus a very strong candidate for being universally accessible. Still, there are only eleven papers in the ICAD proceedings mentioning "web audio", and none describing accessible toolkits.

2.2. Web Audio API

Web Audio API provides a system for controlling audio in a web page that can host advanced audio applications without any need for compilations or installations. Web Audio API is a "W3C Candidate Recommendation" [15] available in all major browsers on desktop as well as mobile devices and lets the developer control deep aspects of audio signal processing using ordinary JavaScript. The introduction of Web Audio API has resulted in academic activities,[1] many online demo examples,[2] commercial tools built for learning,[3] and production of music.[4] While there apparently exist great examples, and although the technology and standard is in active development, Web Audio API is, as seen above, relatively unexplored as a platform for sonification purposes. We argue that the technology is particularly well suited for pedagogical and research purposes due to its open source, cross platform, online accessible structure and that it does not require any installations for the end user.

2.3. WebAudioXML

While Web Audio API is a very powerful system, it has also proven to cause students with little or no prior programming skills to stumble on technical challenges rather than enabling them to express their creativity [16]. As a solution to this problem, WebAudioXML [17] was developed and released as an open source framework [18]. It consists of an XML syntax specification and a parser that offers features for configuring a hierarchical structured audio graph with any number of connected audio nodes. Another core feature is the ability to connect external variables to audio parameters and to map incoming values in a flexible way to meaningful values in the target parameter. In addition to the native audio node types available in Web Audio API, it also provides objects for routing audio signals, buffering audio files and specifying envelopes. For JavaScript developers, there is also an API that makes all audio nodes accessible through the standard Web Audio API syntax.

WebAudioXML is an XML standard like MusicXML [19] but differs a lot since it is not used to describe a musical structure but rather an audio configuration. It adds high level functions and objects to simplify the development of audio applications in a similar way as tone.js [20], but does not contain any built-in synthesizers or effects and does not require any additional JavaScript code. Compared with frameworks like Web Audio Modules (WAM) [21] and Web Audio Plug-in (WAP) [22], WebAudioXML differs radically as it operates on a higher level and potentially act as a container for such frameworks by using the AudioWorklet node [23].

3. WEBAUDIOXML SONIFICATION TOOLKIT

WAST is an open source, online toolkit for exploring sonification models using parameter mapping. There is a demo available[5] that can be accessed using any standard web browser. The source code including a template configuration can be freely downloaded,[6] and the dataset, the audio generators, and the parameter mappings can be modified which allows the developer to distribute a unique configuration on any web site. The design goals for the toolkit were to offer an easy accessible and extensively configurable web-based application for exploring sonification of statistical data. One feature was to have a preset configuration for demo purposes which works out-of-the-box without any installation or configuration. For more advanced users, any aspect of the input data, the audio generation and the parameter mapping should be fully configurable and shareable using standard file formats. The toolkit con-

[1] e.g., the Web Audio Conference, https://webaudioconf.com

[2] e.g., https://github.com/mdn/webaudio-examples

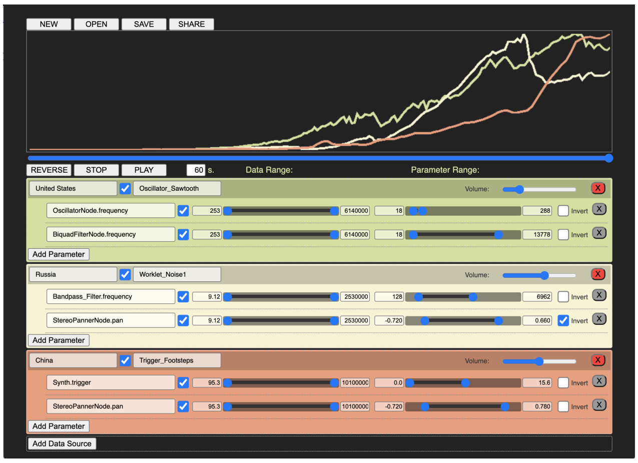
[3] e.g. Music First, https://www.musicfirst.com

[4] e.g. Soundtrap, https://www.soundtrap.com

[5] https://hanslindetorp.github.io/SonificationToolkit

[6] https://github.com/hanslindetorp/SonificationToolkit

sists of a data source module, a visual display, a parameter mapping interface (see figure 1), audio generators and audition, and also features for saving and sharing.



60 s.

United States ☑ Oscillator_Sawtooth
OscillatorNode.frequency ☑ 253 6140000 18 288 □ Invert
BiquadFilterNode.frequency ☑ 253 6140000 18 13778 □ Invert

Russia ☑ Worklet_Noise1
Bandpass_Filter.frequency ☑ 9.12 2530000 128 6962 □ Invert
StereoPannerNode.pan ☑ 9.12 2530000 -0.720 0.660 ☑ Invert

China ☑ Trigger_Footsteps
Synth.trigger ☑ 95.3 10100000 0.0 15.6 □ Invert
StereoPannerNode.pan ☑ 95.3 10100000 -0.720 0.780 □ Invert

Figure 1: The WebAudioXML Sonification Toolkit User Interface with a visual display for the data sources at the top and a list of mapping configurations below.

3.1. Data module

The data module reads a dataset from an external comma separated (CSV) file. The demo installation contains data from https://www.gapminder.org/data/ representing the CO2 emission (in metric tonnes per person) for burning fossil fuels for various countries during the period 1800-2018. The file can be replaced with other available or compiled datasets containing any number of variables that can be selected from within the toolkit. The data needs to be formatted so that the rows contain variables and the columns contain values along a timeline. The first row is reserved for column headers and the first column is reserved for the variable names which are presented in the user interface.

3.2. Visual Display

The visual display generates a curve for each selected variable in the dataset and is primarily designed to support the sonification process; thus, there are no tick marks or units indicated on the axes. The values are distributed on the x-axis and the range on the y-axis is stretched to use the full plot area for each variable, as in Fig. 1. This aims at giving focus to the individual changes for each variable to highlight potential mapping possibilities. There is also a mode available where all variables are displayed using the same range on the y-axis.

3.3. Audio Generators

An audio generator works similar to a virtual instrument in a Digital Audio Workstation (DAW). There could be any number of audio generators in a configuration, and the default template file demonstrates five different possibilities. There is one subtractive synth using a sawtooth oscillator connected through a biquad filter, one synth with a square wave connected through a biquad filter modulated by a low frequency oscillator, a noise generator filtered through a band pass filter, a sample loop player, and finally, a generator triggering short audio files with a controllable time interval. All available parameters are selectable through pop-up menus and are preprogrammed with appropriate minimum and maximum values. All generators are configured using the WebAudioXML syntax and can be built using any number of audio nodes to mix, chain, filter and add effects like distortion, dynamic compression, envelopes, multi-tap delay and convolution reverb. Through the WebAudioXML syntax, it is also possible to simulate various synthesizer models including additive, subtractive and frequency modulated syntheses. Trivial tasks like changing samples can be done by just replacing the audio files. All configurations are stored in an “audio.xml” file.

3.4. Parameter Mapping Interface

The core feature for interacting with the toolkit is the graphical user interface for parameter mapping. It is built using menus for connecting data variables to audio parameters of an audio generator. Any variable can be connected to any number of parameters in one or several audio generators. Each mapping can be specified to use the full range or only a specified part of the range of the variable. In a similar way, the audio parameter can be set to follow the variable between a specified minimum and maximum value. The interpolation will use en exponential curve for parameters such as frequency and gain, while parameters such as stereo panning will use a linear curve. Finally, each mapping can optionally be set to use inverted polarity.

3.5. Audition

The sonification can be auditioned using playback controls similar to a DAW. It can be played forward or backwards with a total playback time specified by the user. A slider beneath the visual display acts as a position pointer and allows the user to interactively explore the relation between the data and the sound either over a section or at a specific point in time. The different audio generators appear on separate tracks and can have independent volume settings similar to the mixer in a DAW. It is also possible to try out different combinations of generators and mappings by turning them on or off independently.

3.6. Saving and Sharing

When an instance of WAST is published to a web site, the configuration of the data and the audio generators are fixed and the explorative aspects of the toolkit are limited and focused to the parameter mapping. Developers can thus preload the toolkit with a dataset and a collection of audio generators for other users to explore by selecting data variables and mapping them to audio parameters. These mappings can be shared, either by saving and opening data from a file or by posting a URL containing all the settings. This feature makes it possible to share a link to a unique sonification configuration that can be explored interactively in any browser.

4. METHOD

At The Royal College of Music (KMH) and the Royal Institute of Technology in Stockholm (KTH) our research projects are often tightly connected with the pedagogical activities of ongoing courses [24]. Many of our courses are aimed at building bridges between art and technology as we want knowledge and practices to be shared between the disciplines. In the present study, we draw from our earlier experiences with students having little or no prior programming skills, and we recruited a student group from an interactive media program at a third university, Södertörn University in Stockholm. There, they study interactive media but not sound or music specifically, and not more than basic web programming.

4.1. Participants

We collected data in conjunction with a bachelor course called “Advanced Visualization Methods” at the media technology department at Södertörn University. In the course, they had four separate week-long sections including theory, methods and hands-on workshops for each of the fields visualization, sonification, haptics and multimodality. Ten students (6F, 4M, age=24–33yrs) with individual sonification projects were included in the study. Several of the students had no musical background, and none had extensive musical background. They did not have any programming courses in their curriculum.

The students were asked to create a sonification using a dataset of their choice (examples were given), a sound source that could either be any of the preconfigured audio generators in a provided template file, or any collection of samples that they could record or download as they found best. They were finally asked to connect the variables in the dataset to one or several parameters in the audio generators. During the allotted project work time, spanning three weeks including holidays, the authors made themselves easily available for individual supervision using video conferencing.

We used two sources for collecting and analyzing data: statistical analysis of the project files, and text analysis of the students reflections in their project report. The students gave their consent for us to use both the code in their projects and the written reflections for this study.

4.2. Statistical Analysis of Project Files

All files from the student projects were analysed by means of counting the number of added and removed audio files, and counting the number of changes done to the WebAudioXML configuration files. We particularly focused on the added and removed audio elements compared to the template file. We also counted the parameter mapping configurations to better understand what parameters and what value ranges for those parameters that the students used. To simplify the analysis of the parameter mapping configurations, we developed a dedicated statistical module in the toolkit that can export all parameter mapping data in a format suitable for statistical analysis in common spreadsheet applications. The mappings were classified and sorted according to the parameter type and the amount of change that the students had made compared to the default values.

4.3. Summary of Student Reflections

All students had to write a project report of 3–4 pages with aims, strategies, result and reflections as part of their examination. In the assignment text, we explicitly asked them to discuss benefits and shortcomings of the toolkit and to present and discuss a vision for a future sonification project. This included describing the data, sonification, target audience, the message they aim to communicate, the purpose, and possible other aspects. We collected and sorted the response into the four categories “benefits and shortcomings”, “affordances”, “feature suggestions” and “future visions” to find general patterns, and to identify and label quotes that expressed typical or interesting points of view.

5. RESULT

All participants managed to complete their tasks with WAST and presented the projects with sounds and parameter mappings based on aesthetic considerations. None of the students required individual supervision during their project work. Furthermore, they could reflect on the process and pointed out thoughtful insights in their project reports. The results are presented in two parts where we first apply statistics to describe the project files, and second, summarize the most important outcome from the students reports.

5.1. Statistical Analysis of Project Files

The projects varied very much in size and complexity: from the smallest project with two source variables and one parameter mapping per variable, to the biggest project with fifteen added audio files and six parameter mappings connected to three different variables. We measured the total number of added audio files and audio elements compared with the template for the different projects and discovered that the smallest project had five changes and the most complex project had 46, see Fig. 2. There are four smaller projects with less than ten changes, three medium size projects with approx. 20 changes, and finally three bigger projects with 40 or more changes.

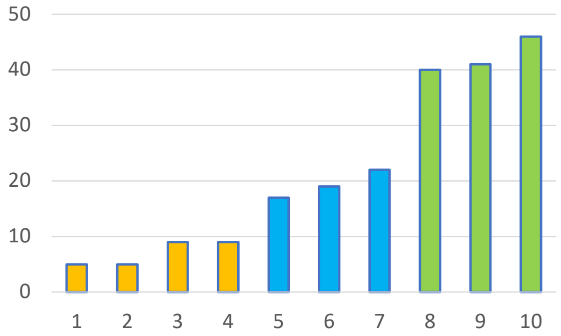


Figure 2: Number of changes made to the template file, per project. Project are arbitrarily grouped by sizes “small”, “medium” and “large” in the chart, indicated by yellow, blue and green colours, respectively.

The template configuration that the students were given contained 35 audio elements (including Mixer, Chain, Synth, Voice, AudioBufferSource, Oscillator, BiquadFilter, Gain, StereoPanner and Convolver elements) but were without parameter mappings. The following data represents the total amount of change the students did in order to accomplish their projects. There were 45 new audio files, 39 AudioBufferSourceNodes, 25 GainNodes, 21 StereoPannerNodes, 22 BiquadFilters and 25 Chain elements added to the projects in total. Those were the most frequently altered objects in the projects. Other types of elements were only occasionally added or deleted and are not considered further here.

The analysis of the parameter mappings indicates that the projects specified 22 variable sources in total and mapped them to

40 target audio parameters. Almost all mappings used the maximum input range of the variables which causes the whole variable range to affect the target audio parameter. None of the mappings used the invert polarity function and seven out of ten projects uses either the default value range or a fixed value for the target audio parameter. The remaining three projects had the target range explicitly set to chosen values for the filter frequency, playback rate and stereo pan parameters, indicating that they were fine-tuned, possibly to serve a particular purpose.

The most used parameters were amplitude gain, filter frequency, playback rate, stereo pan and sample trigger frequency. To understand to what degree a certain parameter type was included in the mapping configuration, we multiply the number of instances of the parameter with the relative usage of its full range. When we compare this value for the five parameter types, it indicates that the usage of gain represents 58% of all the mappings in the projects, frequency 22%, playback rate 13%, and trigger and pan only 7% together, see Fig. 3. The settings for trigger and pan were mostly used to specify a fixed value for playback or mixing purposes rather than mapping to a variable, which also cause low proportions for these parameters.

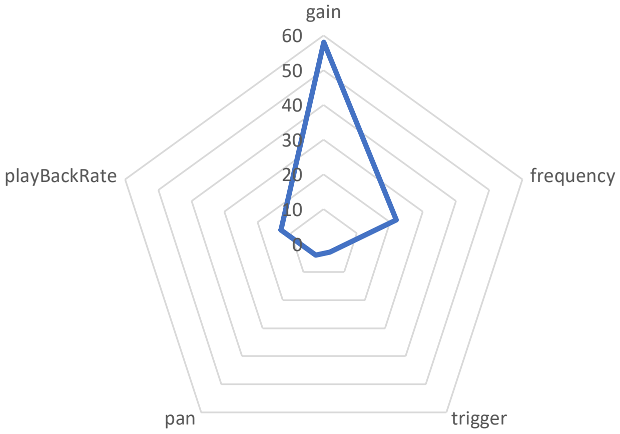


Figure 3: The plot shows the proportions of how the different parameter types were used, measured by combining number of instances and applied range.

5.2. Summary of Student Reflections

In the 3–4 pages long project reports we asked the students to discuss benefits and shortcomings of the toolkit and to present and discuss a vision for a future sonification project including describing the data, sonification, target audience, the message they aim to communicate, the purpose, and possible other aspects. Below we present the findings from reading the reports and reflective texts based on these issues.

Almost all students found the toolkit to be useful and to support creative work, and four students explicitly described the user interface as "straightforward" or "clear" and that a sonification could be created "in a fairly easy manner". One student claimed it to be "an excellent resource for sonification". The shortcomings were mainly focusing on three areas; unfixed bugs, the visual graph and the different applications needed for preparing data and audio configuration. The critique regarding the visual graph focused mainly on the lack of units, tick marks and labels on the axes, and that it was hard to understand the relationships between the different variables when they did not used the same scaling on the y-axis. While most of the students were critical of the practical effect of plotting all data in one graph, one pointed out that the limited visual information caused the creative process to focus more on the sound than the visuals. Several students found it a bit frustrating having to use a spreadsheet application for editing the dataset, and that they had to use an external XML-editor to configure the audio generators. They would have preferred built-in features for all different configurations in one place.

When summarizing the affordances of the toolkit the students expressed how it "facilitates" and helps to "focus on the creative process [to] explore and change parameters and variables" and lets you "create a sonification of data without having to create an own software" or "without the need of learning advanced JavaScript and web development". Three students also pointed out the flexibility and the possibilities for "knowledgeable programmers to make their own version" of the application.

Some of the students contributed with suggestions for further development of the toolkit. They asked for better documentation with "instructions and clear guidelines", a feature for previewing the implemented sounds and more advanced data interpretation that e.g. could "detect meeting/impact points" for different variables allowing them to be mapped to sounds as well.

Most future ideas expressed by the students focused on aesthetic, audience and communicative aspects of sonification. The envisioned future sonifications ranged from product sound design for human–computer interactions to large installations in museums. Several students were intrigued by the emotions induced or communicated from experiencing different sonification examples and envisioned future projects involving environmental and sociological data expressed through sound to contribute to a better world. Students' anticipation of using sonification as a method for communicating data varied from considered unlikely due to lack of knowledge, to enthusiastic. The enthusiasm was particularly associated with a wish to further explore the use of multiple variables and "create something like an orchestra full of sounds".

6. DISCUSSION

WAST has in many ways matched our expectations and met the perceived needs for an efficient tool for students novice to the sonification field. The fact that all of the participants managed to create and distribute an online, interactive application is, arguably, already a success. Even if the students only have touched briefly upon the huge field of auditory display, their project results and reflections clearly indicates that they have gained a lot of insights and experience by working with the toolkit. We also notice that the students could easily work together, and ask and give help by demonstrating their sonifications, which is an indication that the students are taking an understanding (deep) approach to learning [25]. They could also collaborate synchronous as the code is immediately available, which further promotes an active learning strategy that has been mentioned by other scholars [26].

We noticed that the participants used recorded audio files rather than synthesized sounds. This leads us to conclude that they find it easier to start the sonification process from an existing sound rather than building up a virtual instrument from scratch. Further development with outset from this observation could lead to a built-in feature for recording and looping sounds which potentially could release even more creativity and expressiveness in the sonifications.

Most parameter mappings were using the default range for the involved parameters, which makes us speculate that it was found either difficult or unnecessary to fine-tune the mappings. While most credited the user interface as being “straight-forward” and “easy-to-use” we have reasons to believe that it was due to lack of artistic visions or experience rather than technical challenges. A new feature for the toolkit that could potentially address this issue is a section for previewing different sounds and parameters in a similar way as trying out a sound on a synthesizer.

The most frequently used parameters were gain and frequency; while several students described their creative process as being “explorative” it seems like they found the impact of these parameters easiest to understand. Even if the students succeeded in making appreciable sonifications using this limited set of parameters, it leaves a great unexplored potential for manipulating the sounds more artistically by combining several parameters in complex configurations. This would be too complicated for the target student group if no intermediate features are added to the toolkit. As a consequence of this insight, we have performed a design study leading up to a new mapping syntax for WebAudioXML [27]. It can emulate the behaviour of “super-knobs” on commercial keyboards where one main parameter can control a predefined set of lower level audio parameters.

Among the critique and suggestions presented in the student reflections we note some that are especially intriguing: the discussion regarding an integrated interface, the visual guide, and a more advanced interpretation and analysis of the data. The idea to work with all aspects of the sonification process including data preparation, audio configuration, parameter mapping in a single application is brilliant and would lead to a Sonification Workstation with similar affordances as DAW:s have had for the music production community. In such an envisioned product it will be interesting to further explore the possibilities and shortcomings of integrating a more advanced visual tool with the audio section.

The data preparation is another interesting area. A lot of the work can be done in a spreadsheet editor but there are aspects that would benefit from being integrated in a sonification toolkit like the suggested feature to trigger sounds when two different variables are intersecting. If all these envisioned features would integrate into a workstation for sonification it would also point towards the potential for a standardized file format. This format would have to contain the dataset, the audio generators and the parameter mappings, and we argue that such a format would benefit from using standard open formats like XML or JSON.

A feature that was added to the toolkit late in the process was a statistical report feature. It exports the parameter mapping data into a format that is suitable for statistical analysis in other applications. This module was extremely valuable for the analysis of the project files and streamlined the process of collecting parameter mappings from all projects. An important insight from the study is that the value of a statistical report feature cannot be overstated when building tools for research and educational purposes. We recommend taking such a feature into the core design of the application to make evaluation as efficient and extensive as possible.

Finally, it was obvious that the students used some of the parameter mappings solely for mixing purposes. This was extra prominent in their use of the stereo panning parameter. They typically used it to position a sound in the auditory space rather than letting a variable dynamically pan the sound. This points out that there is an element of audio mixing involved in the process of creating a sonification and that a toolkit like WAST would benefit a lot from built-in functions for controlling e.g. volume, pan, EQ, compression and reverb, similar to a mixer in a DAW.

We argue, based on [9], that WAST is more accessible to non-experts than other sonification toolkits like Sonification Sandbox and xSonify due to its inherited web-based, cross-platform behaviour. It is also flexible as any general purpose languages like C++ while the core is fully programmable through a JavaScript API. In contrast to domain-specific languages like Supercollider and Pure Data, WebAudioXML is, strictly, not a language but a fully programmable toolkit that uses the XML standard. Finally, a toolkit based on the Web Audio API is more easy to distribute than other languages and dedicated applications.

Web Audio API was introduced publicly by W3C in 2011, but is still in development. Its potential for both presenting auditory displays in browsers and for allowing the general public to engage in interactive sonifications is great and to a large extent with unexplored possibilities, e.g. [23].

7. CONCLUSION

As presented in this study, we believe that the Web Audio API and both WebAudioXML and WebAudioXML Sonification Toolkit (WAST) are beneficial to web developers, students, sound designers and academics, and that the accessible nature of open source and web-based technology can contribute to reach new groups to become a part of the ICAD community. We also argue that an interactive, web-based tool like WAST can contribute to a more iterative and creative process between customers and manufacturers of product sound design. Not least, it can benefit the general public with broader access to information through sonifications. We also conclude that sonification have much in common with music production and that a toolkit would become an even more creative environment with virtual instruments and mixing features typically found in Digital Audio Workstations.

8. ACKNOWLEDGMENT

We want to thank the participating students for contributing with valuable reflections and letting us analyse and present their projects, and the teacher responsible for the course, Kai-Mikael Jää-Aro.

9. REFERENCES

[1] N. J. Hermann T., Hunt A., “Chapter 1: Introduction,” in *The Sonification Handbook*. Logos Verlag Berlin, 2011, p. 4.

[2] B. J. Grond F., “Chapter 15: Parameter mapping sonification,” in *The Sonification Handbook*. Logos Verlag Berlin, 2011, p. 366.

[3] M. A. Nees and B. N. Walker, “Listener, task, and auditory graph: Toward a conceptual model of auditory graph comprehension,” in *International Conference on Auditory Display, 2007*. Georgia Institute of Technology, 2007.

[4] J. Rohrhuber, “S–introducing sonification variables,” in *In Proceedings of the Supercollider Symposium*, 2010.

[5] J. G. Neuhoff, J. Wayand, and G. Kramer, “Pitch and loudness interact in auditory displays: Can the data get lost in the

map?" *Journal of Experimental Psychology: Applied*, vol. 8, no. 1, p. 17, 2002.

[6] A. P. Mathur, D. B. Boardman, and V. Khandelwal, "Lsl: A specification language for program auralization," in *Proceedings of the 2nd International Conference on Auditory Display (ICAD1994)*. Georgia Institute of Technology, 1994.

[7] C. M. Wilson and S. K. Lodha, "Listen: A data sonification toolkit," in *Proceedings of the International Conference on Auditory Display (ICAD1996)*. Georgia Institute of Technology, 1996.

[8] S. K. Lodha, J. Beahan, T. Heppe, A. Joseph, and B. Zane-Ulman, "Muse: A musical data sonification toolkit," in *Proceedings of the 4th International Conference on Auditory Display (ICAD1997)*. Georgia Institute of Technology, 1997.

[9] S. Phillips and A. Cabrera, "Sonification workstation," in *Proceedings of the 25th International Conference on Auditory Display (ICAD 2019)*. Department of Computer and Information Sciences, Northumbria University, 2019-06.

[10] J. McCartney, "Rethinking the computer music language: Supercollider," *Computer Music Journal*, vol. 26, no. 4, pp. 61–68, 2002.

[11] M. S. Puckette, "Pure data," in *Proceedings: International Computer Music Conference 1997, Thessaloniki, Hellas, 25-30 september 1997*. The International Computer Music Association, 1997, pp. 224–227.

[12] R. M. Candey, A. M. Schertenleib, and W. Diaz Merced, "Xsonify sonification tool for space physics," in *International Conference on Auditory Display, 2006*. Georgia Institute of Technology, 2006.

[13] J. W. Bruce and N. T. Palmer, "Sift: Sonification integrable flexible toolkit," in *Proceedings of the 11th International Conference on Auditory Display (ICAD2005)*. Georgia Institute of Technology, 2005.

[14] K. Beilharz and S. Ferguson, "An interface and framework design for interactive aesthetic sonification," in *Proceedings of the 15th International Conference on Auditory Display (ICAD2009)*. Georgia Institute of Technology, 2009.

[15] P. Adenot and R. Toy, *Web Audio API: W3C Candidate Recommendation, 18 September 2018*, 2018 (accessed February 18, 2020). [Online]. Available: https://www.w3.org/TR/2018/CR-webaudio-20180918/

[16] H. Lindetorp, "Immersive and interactive music for everyone," in *Proceedings of the Nordic Sound and Music Computing Conference 2019 (NSMC2019) and the Interactive Sonification Workshop 2019 (ISON2019) :*, 2019, pp. 16–20. [Online]. Available: http://smcsweden.se/proceedings/NordicSMC_ISon_2019_Proceedings.pdf

[17] H. Lindetorp and K. Falkenberg, "WebAudioXML: Proposing a new standard for structuring web audio," in *Sound and Music Computing Conference*. Zenodo, 2020, pp. 25–31, qC 20200722. [Online]. Available: https://zenodo.org/record/3898655#.X3HgbC0zLa4

[18] H. Lindetorp, *WebAudioXML*, 2020, accessed April 11, 2021. [Online]. Available: https://github.com/hanslindetorp/SonificationToolkit

[19] M. Good *et al.*, "MusicXML: An internet-friendly format for sheet music," in *Xml conference and expo*, 2001, pp. 03–04.

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

[21] J. Kleimola and O. Larkin, "Web audio modules," in *Proc. 12th Sound and Music Computing Conference*, 2015.

[22] M. Buffa, J. Lebrun, J. Kleimola, O. Larkin, and S. Letz, "Towards an open Web Audio plugin standard," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 759–766.

[23] H. Choi, "Audioworklet: the future of web audio," in *Proceedings of ICMC*, 2018.

[24] K. F. Hansen, R. Bresin, A. Holzapfel, S. Pauletto, T. Gulz, H. Lindetorp, O. Misgeld, and M. Sköld, "Student involvement in sound and music research: Current practices at KTH and KMH," in *Proceedings of the first Nordic SMC*. Stockholm: Zenodo, 2019, pp. 36–41.

[25] F. Marton and R. Säljö, "On qualitative differences in learning: I – outcome and process," *British Journal of Educational Psychology*, vol. 46, no. 1, pp. 4–11, 1976-02.

[26] A. Xambo Sedo, R. Støckert, A. R. Jensenius, and S. Saue, "Facilitating team-based programming learning with web audio," in *Proceedings of the International Web Audio Conference*. NTNU, 2019, pp. 2–7. [Online]. Available: https://hdl.handle.net/11250/2647714

[27] H. Lindetorp and K. Falkenberg, "Audio parameter mapping made explicit using webaudioxml," in *Proceedings of the Sound and Music Computing Conference 2021 (SMC2021)*, 2021.