Towards the Next Generation of Web-based Experiments: A Case Study Assessing Basic Audio Quality Following the ITU-R Recommendation BS.1534 (MUSHRA)

Michael Schoeffler
International Audio Laboratories Erlangen*
Am Wolfsmantel 33
Erlangen, Germany
michael.schoeffler@audiolabs-erlangen.de

Fabian-Robert Stöter
International Audio Laboratories Erlangen*
Am Wolfsmantel 33
Erlangen, Germany
fabian-robert.stoeter@audiolabs-erlangen.de

Bernd Edler
International Audio Laboratories Erlangen*
Am Wolfsmantel 33
Erlangen, Germany
bernd.edler@audiolabs-erlangen.de

Jürgen Herre
International Audio Laboratories Erlangen*
Am Wolfsmantel 33
Erlangen, Germany
juergen.herre@audiolabs-erlangen.de

ABSTRACT

Listening tests are widely used to assess the quality of audio systems. The majority of such listening tests is conducted in controlled environments with selected participants and professional audio equipment. In the last few years, conducting listening tests over the Internet, as so called web-based experiments, has become popular. A recent study has shown that web-based experiments lead to comparable results as laboratory experiments.

Until now, it was only possible to implement a limited number of listening test types as web-based experiments because web standards were missing some crucial features, e. g. sample manipulation of audio streams. With the upcoming of the Web Audio API, a much wider range of listening test types can be implemented as new audio processing features have been introduced. This paper demonstrates which new possibilities are enabled by the Web Audio API. To this end, the ITU-R Recommendation BS.1534 (MUSHRA) is taken as an example.

Keywords

MUSHRA, ITU-R Recommendation BS.1534, web-based experiments

1. INTRODUCTION

In the field of audio, assessments are conducted to find out what benefit new audio systems (systems under test) have compared to the state of the art. Audio assessments can be categorized into objective and subjective evaluation methods. Objective evaluation methods assess the system under test based on a defined algorithm. Prominent examples of such objective methods are PEAQ [4] for assessing audio quality and POLQA [10] for assessing speech quality. Subjective evaluation methods are assessments which are conducted by humans. If humans take part in a structured subjective evaluation, one speaks of listening tests. Well-known recommendations for designing such listening tests are, for example, ITU-R BS.1534 [11] for assessing intermediate quality audio coding systems and ITU-R BS.1116 [7] for high quality audio coding systems.

Since time and other resources are spent on recruiting participants, carrying out the listening test and analyzing the results, listening tests are more resource-consuming than objective methods which assess systems under tests typically in a few seconds or minutes. Another advantage of objective evaluation methods is that the results are reproducible for each assessment if the conditions are not changed. Results of listening tests are only reproducible to a certain degree depending on the statistical significance of the results. However, listening tests have one major benefit compared to objective evaluation methods: their results fully reflect the perception of humans who are the target of most of the new audio developments. How well objective evaluation methods reflect the perception of humans depends on the specific attribute that is assessed. For example, when assessing audio quality, results obtained by PEAQ correlate well with results of an ITU-R BS.1116 listening test conducted under certain conditions. However, current objective evaluation methods are not accurate enough when assessing more holistic attributes as, e. g. the overall listening experience where listeners are asked to take everything into account that influences their enjoyment [19]. Especially the overall listening experience is an attribute which is rated very differently by listeners [20]. Thus, objective evaluation methods must consider not only the sensation and perception of humans but also yet unknown cognitive processes where only a limited prediction algorithm exists so far [21]. In such a case or when an objective evaluation method is simply not available, listening tests are the only choice for assessments. A noteworthy fact is that many objective evaluation methods are based on the results of listening tests. For example, PEAQ utilizes a neural network which was fitted with listening test results. This emphasizes the importance of listening tests in audio research and engineering.

Bringing listening tests to the web and conducting them as so-called web-based experiments (also called web experi-

*The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Fraunhofer-Institut für Integrierte Schaltungen IIS.

ments or Internet experiments) has become popular. A web-based experiment is an experiment developed by using web technologies (HTML, JavaScript, etc.) and runs within a browser. Typically, a web-based experiment is conducted over the Internet. Web-based experiments have many advantages compared to laboratory experiments (but also some disadvantages). For example, web-based experiments are especially useful when the test of a hypothesis requires more participants than locally available. In such a case, a web-based (and "crowdsourced") experiment might be a solution for getting more participants. In addition, web-based experiments simplify the recruitment process, especially, if participants from different cultures speaking different languages are required. Already in 1996, Welch and Krantz started to conduct web-based experiments [26]. Since then, it was unknown whether the validity of web-based auditory experiments is sufficient or not, since many environment variables cannot be controlled. However, Schoeffler et al. compared laboratory- and web-based results (62 and 1168 subjects) of an auditory experiment and found no significant differences [22, 25]. Their results indicate that, if the experiment is well designed and the uncontrolled variables are negligible for the research question, the results of web-based experiments are reliable. Nowadays, web-based experiments have become an established alternative to traditional laboratory experiments.

The purpose of this paper is to demonstrate how the new Web Audio API will improve web-based experiments and enable new opportunities for designing them. Therefore, the widely used subjective evaluation method MUSHRA [11] is used as an example. This paper illustrates how a recommendation-compliant MUSHRA framework has been implemented and why this was not possible before.

2. RELATED WORK

A web-based MUSHRA framework named BeaqleJS has been published before by Kraft and Zölzer [13]. However, their MUSHRA framework is purely based on HTML5 and JavaScript and does not utilize the Web Audio API. Hence, it is impossible to implement MUSHRA fully recommendation-compliant without the Web Audio API since functionality for manipulating samples of an audio stream is needed. Besides the audio processing, also the Graphical User Interface (GUI) layout is not fully recommendation-compliant. For example, BeaqleJS shows horizontal rating scales but ITU-R BS.1534 recommends vertical rating scales. Our proposed implementation targets researchers and engineers who want to conduct fully compliant MUSHRA tests.

The Web Audio API is already used by researchers for purposes which are not related to auditory experiments. For example, Choi and Berger implemented WAAX which is a comprehensive audio library based on Web Audio API [3]. Moreover, Borins built a compiler that transforms FAUST (Functional Audio Stream) into JavaScript code that uses Web Audio API [2]. FAUST is a functional programming language specifically designed for real-time signal processing and synthesis [15]. Furthermore, Herrero utilized the Web Audio API to implement a browser-based interactive music player based on the MPEG-A Interactive Music Application Format (IM AF) [6]. Another example is a browser-based real-time binaural synthesis processor by Collados which uses the Web Audio API for audio processing [12].

3. LIMITATIONS OF CURRENT WEB STANDARDS

Web-based listening tests are developed by using web technologies. There exists a wide range of various web technologies ranging from proprietary to non-proprietary technologies and ranging from a small number to a huge number of supporting devices and browsers. This paper focuses on technologies based on open web standards in a sense of formal and non-proprietary specifications, in particular HTML recommendations by the World Wide Web Consortium and the ECMA Script Language Specifications (often referred as Java Script) by Ecma International. Proprietary technologies, like Adobe Flash or ActiveX, are not considered in this paper since open specifications are usually supported by a wider range of devices and browsers.

For playing audio inside the browser, one of the first possibilities was using the `object` element which was standardized in HTML 4.01 [16]. The `object` element is used to embed multimedia objects (like audio, video, Java applets, ActiveX, etc.) into web pages. The appearance of the `object` element depends on the browser and it usually has only basic functionalities (play, pause, stop, etc.). Moreover, the audio was not played back natively by the browser but by a third party plugin (e. g. Adobe Flash or Apple Quicktime). If the required plugin for playing back audio is not installed, the `object` element does not work. Almost the same functionality as the `object` element is offered by the `embed` element which is specified in HTML5 [1], although it has been supported by major browsers for a long time. The limitations of both elements, having only basic functionality, resulted in introducing the `audio` element which is also specified in HTML5 [1].

The `audio` element offers a wide range of attributes (e. g. `preload`, `mute` and `volume`) or events (e.g., `loadeddata` and `volumechange`) for controlling and monitoring the audio play back. Although the `audio` element offers much more audio functionality than the `object` and `embed` element, it still misses some features which are required for many types of listening tests. The most important limitation is that the `audio` element does not support to arbitrarily manipulate the samples of its audio stream. Any listening test which requires dynamic modifications of the stimulus, e. g., applying crossfadings, cannot be implemented. Another significant limitation of the `audio` element is that it uses the predefined (operating system) configuration of the audio interface. Some listening tests require to instantly switch between mono, stereo, quadraphonic and surround systems which might not be possible by using one static configuration in a listening test. Further examples are localization tests, where the same stimulus is played back by different loudspeakers (e. g., as in [23]). When implementing such tests, it is very convenient to dynamically play back a stimulus by an arbitrary loudspeaker. Moreover, for web-based experiments, it can be very helpful to collect information about the participant's software and audio interface used. Regarding the software, helpful information could be browser type, browser version, operating system and screen resolution. Regarding the audio interface, helpful information could be sample rate and the number of input and output channels. Such information about the audio interface cannot be collected by using the `audio` element.

Most of the limitations of current web standards are compensated with the introduction of the Web Audio API which enables extended access to the host audio interface, including sample manipulation of audio streams and control of the channel configuration.

4. IMPLEMENTING ITU-R BS.1534

4.1 MUSHRA Methodology

In 2001, the ITU formally described in Recommendation BS.1534-0 the first version of MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) which is a test method for assessing intermediate audio quality [8]. A second revision of MUSHRA was just recently published in ITU-R Recommendation BS.1534-2 which introduced changes in the test design as well as in the analysis of the results [11].

In a MUSHRA test, assessors are presented with an open reference stimulus and a number of test stimuli (conditions). The conditions contain the hidden reference stimulus, at least two anchor stimuli (low quality anchor and mid quality anchor) and stimuli which were processed by the systems under test. MUSHRA allows to have a maximum of eleven conditions (eight systems under test, two anchors and the hidden reference) to be presented in one trial. Since the conditions are shown in random order, the assessor does not know which condition is a system under test, the hidden reference or an anchor. All conditions are rated relatively to the open reference stimulus. By adding the hidden reference stimulus to the conditions, an anchor for the highest rating is implicitly set. Without adding the hidden reference, the highest rating might also be given to a system under test which is the best system among all conditions but still causes some artifacts. With this design, it is more likely that only conditions are rated with the highest score which cannot be distinguished from the open reference. Another purpose of the hidden reference is to find out whether an assessor would rate the hidden reference stimulus with a very high or the highest rating. If an assessor rates the hidden reference too low in too many trials, his or her ratings are excluded from the results by the post-screening process. BS.1534-2 recommends to add two anchor stimuli which are low-passed filtered versions of the reference stimulus. The low quality anchor has a cut-off frequency of 3.5 kHz and the mid quality anchor has a cut-off frequency of 7 kHz.

When assessing the conditions, the assessors are allowed to switch instantaneously between conditions and the open reference while listening. Besides switching between conditions, looping is a common strategy for assessing the conditions. Looping means that only an excerpt of the stimuli is marked and listened to repeatedly while assessing. For example, by marking a one-second critical part of a 10-second long stimulus, it is much easier to focus on the artifacts of this part caused by the system under test. The conditions are rated according to a continuous quality scale (CQS) ranging from 0 to 100. The scale is divided in five equal intervals, where each interval is labeled with an adjective (0-20: Bad, 20-40: Poor, 40-60: Fair, 60-80: Good, 80-100: Excellent).

In most cases, the assessors are asked to rate the Basic Audio Quality (BAQ) of each condition. In ITU-R Recommendation BS.1534 Basic Audio Quality is defined as:

> "This single, global attribute is used to judge any and all detected differences between the reference and the object."

The MUSHRA methodology has been investigated by various researchers. E. g., Zielinski et al. found potential biases in MUSHRA listening tests. For revealing these biases, they designed two experiments according to the first revision of MUSHRA (described in ITU-R Recommendation BS.1534-1 [9]). Another example is a study by Schinkel-Bielefeld et al. where they investigated the differences between experienced and inexperienced listeners who were participating in a MUSHRA listening test [18]. Furthermore, Schinkel-Bielefeld et al. conducted another listening test to find out whether the familiarization of items has an influence on ratings [17]. Regarding the results analysis, Sporer et al. and Nagel et al. addressed some statistical aspects [24, 14].

Although MUSHRA has been originally designed to evaluate the quality of audio coding systems, it is widely used for evaluating other types of audio systems. For example, in applications of source separation Emiya et al. used a variation of the MUSHRA test to assess the quality of systems aiming at extracting sound sources from a mixture [5].

4.2 Technical Requirements for Implementing MUSHRA

The experiment GUI, for presenting the stimuli and the reporting method, as well as the audio processing are critical components for auditory experiments since the presentation and minor defects in the audio processing might have a significant influence on the participants' responses. For example, it has been shown that the appearance of the GUI has a significant influence on the average time that is needed for reporting the location of a stimulus in localization tests [23]. Regarding the defects introduced by audio processing, if, e. g., high-quality audio coding systems are assessed, ratings of the assessors might be strongly influenced by artifacts caused by the defects rather than by the artifacts caused by the audio coding systems. Nevertheless, a proper audio processing (e. g., without clipping and frame drops) is a prerequisite of any type of auditory experiment and not only of MUSHRA.

The idea of MUSHRA is to have a methodology which can be used for listening tests carried out at different places and still leading to very comparable results. Therefore, the ITU-R Recommendation BS.1534 describes the presentation and the audio processing in detail. A framework for conducting MUSHRA tests must have the possibility to render all control elements needed for the test, including buttons for playing back the stimuli and a scale for rating the items.

With respect to the audio processing, a framework must be able to load uncompressed PCM (Pulse-code modulation) or losslessly encoded files that store the stimuli. If a framework only allowed to load files of an audio coding format with lossy compression, the lossy compression would introduce additional artifacts into the stimuli. In the context of web-based experiments, it depends on the browser whether uncompressed files can be loaded. However, even if the browser does not natively support to load uncompressed files, one can write a fall-back function for loading uncompressed files and storing the audio samples into a buffer. With the help of the Web Audio API, it is possible to play back the samples stored in such a buffer. ITU-R Recommendation BS.1534 recommends to apply a 5 ms fade-in and 5 ms

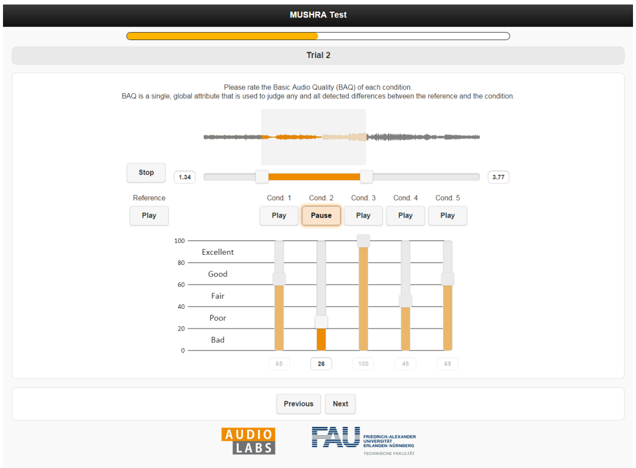

Figure 2: Screenshot of the MUSHRA implementation.

fade-out with a raised-cosine-envelope whenever the assessor switches between test conditions or a loop has ended and starts again from the beginning. Figure 1 depicts a timing diagram which shows a typical scenario where the assessor switches between items and sets loops. For applying such a fade-in and fade-out at any point in time, it is required to manipulate the samples of an audio stream. Since the Web Audio API allows to manipulate the audio stream, a fade in and fade out, as recommended by the ITU-R Recommendation BS.1534, can be implemented.

4.3 Web-based MUSHRA Implementation

4.3.1 *Software Design*

A framework was developed which implements the second revision of the ITU-R Recommendation BS.1534 and is based on the web standards HTML5, JavaScript and Web Audio API. The libraries jQuery and jQuery Mobile were used for manipulating the Document Object Model (DOM) and for implementing the Graphical User Interface.

The framework is configured by a single file which enables to set up an individualized MUHSRA test. In more detail, the config file allows to add different types of pages to the listening test, e. g., a page which shows instructions, another page to adjust the loudness, and pages to rate the stimuli according to MUSHRA. The content as well as the order of the listening test pages are fully customizable. Moreover, the config file contains a list of stimuli (reference and test conditions) for each trial. If desired, the low and mid anchor are automatically generated based on the reference stimulus.

The values of the config file are used to render MUHSRA pages as shown in Figure 2. For each MUSHRA page, it can be configured whether to allow looping of the stimuli or not. In addition, an advanced loop element can be activated which shows the wave form of the stimuli that is playing back.

The ratings are exported as comma-separated values (CSV) and can either be send by mail to the experimenter or stored in a file on the server.

ITU-R Recommendation BS.1534-2 has many constraints which must be considered to be recommendation-compliant (e. g., max. 11 conditions). These constraints are checked

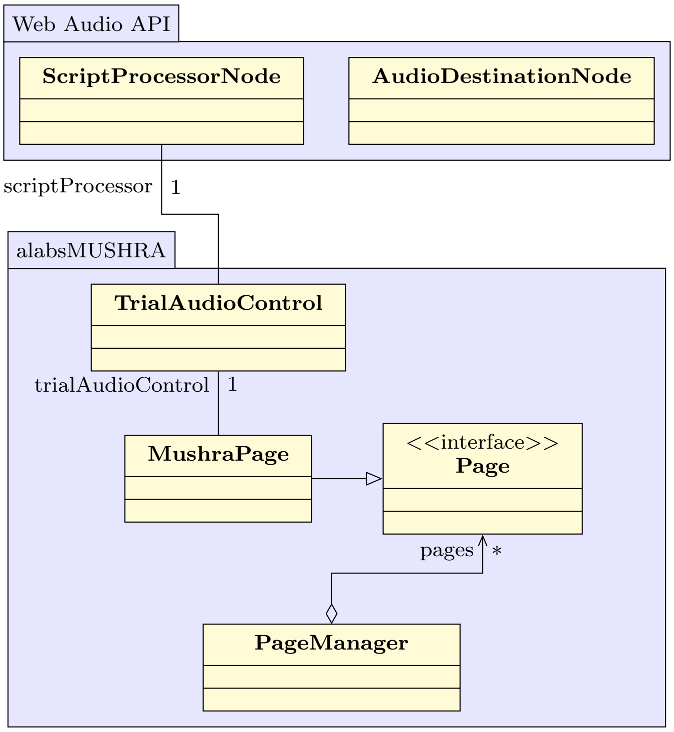

Figure 3: The software architecture of the implementation depicted as class diagram.

by our implementation while loading and processing the listening test configuration. If a constraint is violated, an error message is shown.

4.3.2 *Software Architecture*

For presenting a MUSHRA GUI to the assessor, three different classes (`PageManager`, `MushraPage`, `Trial`) are mainly involved. Depending on the entries in the config file, a number of pages is added to the `PageManager`. The `PageManager` controls the sequence of the pages and stores page-specific values (e. g., MUSHRA ratings) for further processing. One of these pages might be a `MushraPage` which is the class for representing the actual MUSHRA GUI. The required audio processing is fully separated from the `MushraPage` class and contained by the `TrialAudioControl` class. The `TrialAudioControl` class has all methods needed for a MUSHRA test, like starting and stopping the play back of conditions, switching to another condition and starting fading. The `TrialAudioControl` class utilizes the Web Audio API, in particular one `ScriptProcessorNode`[1] connected to the `AudioDestinationNode`. A `ScriptProcessorNode` allows the generation, processing or analyzing of an audio stream using JavaScript. The `AudioDestinationNode` represents the speakers of the audio interface. Figure 3 depicts the software architecture.

4.3.3 *Software Implementation*

More information about the implementation of the soft-

[1]At time of writing this paper, the `ScriptProcessorNode` was marked as deprecated and is going to be replaced by `Audio Workers` in the near future.

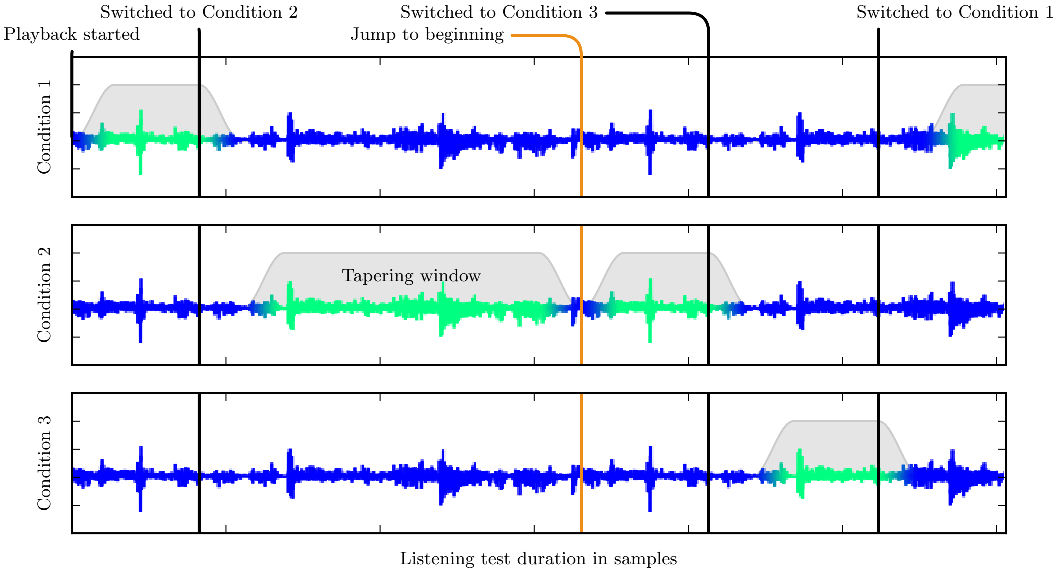

Figure 1: Timing diagram excerpt of a typical MUSHRA session showing the audio processing while switching between conditions and using loops. Green colored waveform indicates the user's current playback audio stream.

ware can be found at `http://www.audiolabs-erlangen.de/resources/webMUSHRA`.

5. OUTLOOK

We expect to see a new generation of web-based auditory experiments once the Web Audio API is standardized as W3C Recommendation and is widely supported. A lot of test methodologies, like MUSHRA, require advanced audio processing functionality which has been a limitation of previous web standards. However, by using the Web Audio API, researchers have the possibility to bring more auditory experiments to the web and conduct them as web-based experiments. Moreover, methodologies which are based on the "Method of Adjustment", where participants dynamically control the independent variable that influences the auditory stimulus, could not be easily implemented without the Web Audio API. As the influence of the independent variable on the audio stream could not be processed dynamically, for each level of adjustment, a static audio file had to be previously generated. By utilizing the Web Audio API, the required dynamic audio processing can be easily implemented. Furthermore, almost any type of auditory experiments which uses input signals (e. g., voice of a participant) can be realized since capturing and processing of an input audio stream is supported by the Web Audio API. E. g., the participant's microphone signal can be captured, to monitor the loudness of the environment. For some types of listening tests it is crucial that the background noise level is below a certain threshold. Moreover, using a web-based infrastructure for listening tests might reduce the organizational effort for both participants and experimenters, since no fixed appointments for the sessions are needed anymore. Besides the Web Audio API, also other new web standards help to improve web-based auditory experiments. E. g., the WebSocket API and the WebRTC API enable real-time interaction during a listening sessions. By using these APIs, the experimenter could monitor an ongoing listening session or directly talk to a participant which brings web-based experiments even closer to laboratory experiments.

To date, Web Audio API still has room for improvements in reliability and interoperability. For many auditory experiments it is crucial that no samples are dropped from the output buffer (resulting in possible clicks and cracks). Therefore, advanced error handling would be beneficial to detect unsuccessful trials. Furthermore, experiments would benefit from extended access to the audio interface to read information, such as all supported channel configurations and samplerates.

6. CONCLUSION

The new opportunities of web-based auditory experiments, triggered by the spreading of the Web Audio API, have been demonstrated. By using the Web Audio API, we built a fully-compliant implementation of the ITU-R Recommendation BS.1534 (MUSHRA) as web framework. Our implementation targets researchers, engineers and developers who want to conduct recommendation-compliant MUSHRA tests.

7. REFERENCES

[1] R. Berjon, S. Faulkner, T. Leithead, S. Pfeiffer, E. O'Connor, and E. D. Navara. HTML5. Candidate Recommendation, W3C, July 2014.

[2] M. Borins. From Faust to Web Audio: Compiling Faust to JavaScript using Emscripten. In *Linux Audio Conference*, Karlsruhe, Germany, 2014.

[3] H. Choi and J. Berger. WAAX: Web Audio API Extension. In *Proceedings of New Interfaces for Musical Expression (NIME)*, pages 499–502, Daejeon, Korea, 2013.

[4] C. Colomes, C. Schmidmer, T. Thiede, and W. C. Treurniet. Perceptual Quality Assessment for Digital Audio: PEAQ – The New ITU Standard for Objective Measurement of the Perceived Audio Quality. In *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*, pages 337–351, Florence, Italy, 1999.

[5] V. Emiya, E. Vincent, N. Harlander, V. Hohmann, et al. The PEASS Toolkit-Perceptual Evaluation methods for Audio Source Separation. In *9th Int. Conf. on Latent Variable Analysis and Signal Separation*, 2010.

[6] G. Herrero, P. Kudumakis, L. J. Tardón, I. Barbancho, and M. Sandler. An HTML5 Interactive (MPEG-A IM AF) Music Player. In *10th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Marseille, France, 2013.

[7] International Telecommunications Union. ITU-R BS.1116-1 (Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems), October 1997.

[8] International Telecommunications Union. ITU-R BS.1534-0 (Method for the subjective assessment of intermediate quality levels of coding systems), June 2001.

[9] International Telecommunications Union. ITU-R BS.1534-1 (Method for the subjective assessment of intermediate quality levels of coding systems), January 2003.

[10] International Telecommunications Union. P.863 (Perceptual Objective Listening Quality Assessment), 2011.

[11] International Telecommunications Union. ITU-R BS.1534 (Method for the subjective assessment of intermediate quality levels of coding systems), July 2014.

[12] A. Julià Collados. *Design of a Binaural Synthesis Processor in the Browser using Web Audio API.* Thesis, Universitat Politècnica de Catalunya, 2014.

[13] S. Kraft and U. Zölzer. BeaqleJS: HTML5 and JavaScript based Framework for the Subjective Evaluation of Audio Quality. In *Linux Audio Conference*, Karlsruhe, Germany, 2014.

[14] F. Nagel, T. Sporer, and P. Sedlmeier. Toward a Statistically Well-Grounded Evaluation of Listening Tests—Avoiding Pitfalls, Misuse, and Misconceptions. In *Audio Engineering Society Convention 128*, May 2010.

[15] Y. Orlarey, D. Fober, and S. Letz. Syntactical and Semantical Aspects of Faust. *Soft Computing*, 8(9):623–632, July 2004.

[16] D. Raggett, A. Le Hors, and I. Jacobs. HTML 4.01 Specification. Technical report, W3C, Dec. 1999.

[17] N. Schinkel-Bielefeld, N. Lotze, and F. Nagel. Does Understanding of Test Items Help or Hinder Subjective Assessment of Basic Audio Quality? In *Audio Engineering Society Convention 133*, Oct 2012.

[18] N. Schinkel-Bielefeld, N. Lotze, and F. Nagel. Audio quality evaluation by experienced and inexperienced listeners. In *Proceedings of Meetings on Acoustics Vol. 19*, Montreal, Canada, 2013. Acoustical Society of America.

[19] M. Schoeffler, A. Adami, and J. Herre. The Influence of Up- and Down-mixes on the Overall Listening Experience. In *Proc. of the AES 137th Convention*, Los Angeles, USA, 2014.

[20] M. Schoeffler and J. Herre. About the Different Types of Listeners for Rating the Overall Listening Experience. In *Proceedings of Sound and Music Computing Conference 2014*, Athens, Greece, 2014.

[21] M. Schoeffler and J. Herre. Towards a Listener Model for Predicting the Overall Listening Experience. In *Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound*, pages 11:1–11:8, Aalborg, Denmark, 2014. ACM.

[22] M. Schoeffler, F.-R. Stöter, H. Bayerlein, B. Edler, and J. Herre. An Experiment about Estimating the Number of Instruments in Polyphonic Music: A Comparison Between Internet and Laboratory Results. In *Proceedings of 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013. International Society for Music Information Retrieval.

[23] M. Schoeffler, S. Westphal, A. Adami, H. Bayerlein, and J. Herre. Comparison of a 2D- and 3D-Based Graphical User Interface for Localization Listening Tests. In *Proc. of the EAA Joint Symposium on Auralization and Ambisonics*, pages 107–112, Berlin, Germany, April 2014.

[24] T. Sporer, J. Liebetrau, and S. Schneider. Statistics of MUSHRA Revisited. In *Audio Engineering Society Convention 127*, October 2009.

[25] F.-R. Stöter, M. Schoeffler, B. Edler, and J. Herre. Human Ability of Counting the Number of Instruments in Polyphonic Music. In *Proceedings of Meetings on Acoustics Vol. 19*, Montreal, Canada, 2013. Acoustical Society of America.

[26] N. Welch and J. H. Krantz. The World-Wide Web as a Medium for Psychoacoustical Demonstrations and Experiments: Experience and Results. *Behavior Research Methods, Instruments, & Computers*, 28(2):192–196, June 1996.