Speech Therapy Software on an Open Web Platform

Selim S. Awad, Ph.D
The University of Michigan-Dearborn
Department of Electrical and Computer Engineering
Dearborn, Michigan 48128
Email: sawad@umich.edu

Christopher Piechocki, B.S.E
The University of Michigan-Dearborn
Department of Electrical and Computer Engineering
Dearborn, Michigan 48128
Email: cpiechoc@umich.edu

***Abstract*—Over the past several decades there have been a handful of speech therapy software programs made available to people with speech problems, specifically for stuttering related speech issues. The goal of this paper is to introduce the ability to offer these patients a speech therapy program accessible from an ordinary web browser and thereby gaining the many benefits of using an open web platform. With the open web platform, a rich internet application can be developed with real time audio processing and media capture via the *Web Audio API* and *Media Capture and Streams API*. By combining these powerful APIs with an online database, patients can have a fully managed speech therapy application that can be accessed from the comfort of their homes. This new type of speech therapy application will allow the doctor to monitor the patient's advancements outside of the clinical setting.**

***Keywords*—Speech therapy, Open Web Platform, Web Audio API, Media Capture and Streams.**

I. Introduction

In 1967, research in stuttering led to the finding that persons who stutter could be instructed to prolong their speech in order to reduce their stuttering frequency. These studies would eventually lead to the preferred treatment techniques for reducing stuttering. These "fluency shaping" procedures include reducing speech rate, increasing the duration of phonation, and imitating prolonged speech samples [10]. More recently, software applications have been developed to enhance these speech therapy methods. One particular example is the StuSoft/Prosidy programs developed at the University of Michigan-Dearborn and used by the Speech & Language Department at William Beaumont Hospital of Royal Oak Michigan. This software provides a patient with the ability to have real time audiovisual feedback of the shape of his/her speech. The program also provides a scoring measure of the quality of the patient's speech. This digital feedback allows the patient to be aware of the limitations of his/her speech and thereby make corrections. The drawback of the StuSoft/Prosidy speech therapy software is in how it was designed to be deployed. It was originally developed as a PC application for the operating systems available at the time (with the most recent major update in 1998). The aim of this paper is to demonstrate the feasibility of developing speech therapy software utilizing the open web platform. There are a multitude of benefits to using an open web platform that are not easily achieved with a standard PC application:

- No software installation
- Access from multiple devices (including mobile devices)
- Tracking progress with an online database
- Remote monitoring of patient's progress

A. Speech Therapy with Audio-Visual Feedback

The web application software that is being demonstrated here is aiming to provide an audiovisual feedback system to aid in the established fluency shaping procedures. The crux of how beneficial the software will be is whether or not audio-visual feedback systems can influence a patient in controlling characteristics of their speech. During a study of the effects of modifying phonation intervals on stuttering, it was found that providing audiovisual feedback to the patient was an adequate tool for enabling the patient to control characteristics of their speech [10]. These findings lend themselves to the hypothesis that audiovisual feedback may also be beneficial during the fluency shaping procedures. In addition, it was noted in that same study that there are other advantages to computerized speech feedback systems. One problem with fluency shaping is that it requires clinicians to be trained in producing specific speech characteristics. Such training yields characteristics that may vary widely which may cause the results to be less consistent [10]. The proposed web application would provide a consistent audio speech utterance that the patient would try to mimic by being in the form of a recording. In addition, the web application provides software algorithms that apply consistent criteria for evaluating the quality of the patient's mimicked speech. In this manner, the proposed web application would at least mitigate some of the limitations discussed with the fluency shaping procedures as they exist today.

B. Software Frameworks Considered

Many frameworks were considered for delivering a modern speech therapy software program. The main considerations were Java for its vast PC support, a mobile OS (Android or iOS) application for its ease of access and finally an open web platform (HTML5 and JavaScript) for its broad accessibility.

On Oracle's website www.java.com, they tout their Java platform as being accessible on 89% of US desktops and 3 billion mobile phones running Java. However, devices such as Apple's iPhone do not support Java and are unlikely to in the future as the end user license agreement forbids executable code from running on the device [5]. While the PC support for Java is impressive, it was removed from consideration due to its limited support on mobile devices and the unlikelihood of this support changing in the future.

978-1-4799-5241-0/14/$3100

Another consideration for the development of a modern speech therapy software application was to choose one of the popular smart-phone or tablet operating systems, specifically iOS or Android. However, choosing one of these platforms would have automatically excluded PC options from patients who do not plan to purchase a smart phone. In addition, development on these platforms would require continued support for new devices in a market that changes rapidly.

In investigating the best platform for delivering a modern speech therapy application, an application using the open web platform was considered for its broad accessibility of being available on both PC and mobile web browsers alike [6]. The crux of whether this platform would work or not is its support for microphone access, for without said support, the platform is not capable of delivering speech therapy applications. However, the open web platform is growing quickly and has recently added support for microphone access through the browser API: *Media Capture and Stream*. Thus, a speech therapy application can be delivered on the open web platform by coupling HTML5 with its graphical canvas tag, JavaScript, and browser APIs. In addition, a web-based application can be connected to a server-side database to allow for tracking of the patient's progress.

After weighing the advantages and disadvantages of each platform, the open web platform was chosen as it could meet all the needs to deliver a modern speech therapy software application.

II. Speech Therapy on an Open Web Platform

The developed speech therapy web application utilizes two key open web platform audio APIs: the *Web Audio API* and *Media Capture and Streams API*. These two APIs provide the means to access the user's microphone and then process it in real time.

A. Web Audio API Overview

The *Web Audio API* is a JavaScript API available in many web browsers that allows for "...the mixing, processing, and filtering tasks that are found in modern desktop audio production applications" [1]. As described in the specification of the API, the concept for audio processing is to provide a nodal structure of objects within an "Audio Context" object. These objects are JavaScript objects that are created, initialized and connected within the JavaScript code. For the developed speech therapy web application, the node structure can be seen in Fig. 1.

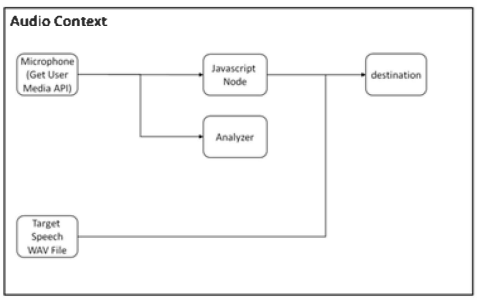

Fig. 1. Audio Context of Speech Therapy Web Application

There are two inputs into this Audio Context: the user's microphone and a target speech wav file. The user's microphone input will be discussed later as this node involves a second API. The target speech wav file, contains the *goal* speech utterance that which the user will try to mimic. The *Web Audio API* allows for the access of this wav file and its PCM data. It is the PCM data that must be processed in order to give real-time feedback to the user. In addition, the *Web Audio API* allows for the wav file to be played on button click events initiated by the user [8].

The "JavaScript" node seen in Fig. 1 is a special type of node created using the JavaScript method *createScriptProcessor()*. This node provides the ability for the developer to process the audio data via JavaScript. The method takes as a parameter a buffer size. When the number of available audio samples equals this buffer size, an event is triggered and a JavaScript function is executed for the audio processing to occur. This is the means by which real-time processing of the microphone data is possible [7][9].

B. Media Capture and Stream API Overview

Returning to the microphone node seen in the Audio Context diagram, this node makes use of the *Media Capture and Streams API*. This API utilizes a browser method called *getUserMedia()* which allows the developer to access a user's multimedia devices such as web cameras and microphones [2]. For this speech therapy web application, the multimedia device of interest is the user's microphone. It is this device that the user will speak into when prompted by the application. While the *getUserMedia()* method provides access to the device, the *Web Audio API* provides the access to the PCM data in real time via the "JavaScript" node referenced in the AudioContext diagram[7][9].

One of the limitations of using these APIs as of this writing, is that the sampling rate will be fixed based on the sound card of the user's device. The software can be developed to handle multiple sampling rates. However, for this particular application, the default sampling rate will most likely be much higher than desired because the application is only concerned with the frequency range of speech rather than what most sound cards are designed to handle in order to playback music. This means that additional processing is needed and wasted to handle the fixed sampling rate offered by the APIs. Some investigations were made into the feasibility of decimating within the JavaScript node; however, the savings on the backend calculations were not worth the expense of the front end filtering.

III. Speech Therapy Web Application Software

A. Short Time Average Magnitude Processing

The audiovisual feedback given to the user is in the form of a plot of the short time average magnitude of his/her speech utterance [3].

$$Mn = \sum_{m=0}^{6s} |x(m)|w(n-m) \quad (1)$$

where $w(n-m)$ rectangular window

The length of the window was designed to be 20ms with 10ms shifts (50% overlap) based on the criteria used by the Stusoft/Prosidy software package [3]. However, the JavaScript node from the *Web Audio API* provides data in frames with length of a power of 2 (for this application, the frame length is 1024 samples). This combined with the fixed sampling rate (48000 Hz in one particular test environment) discussed earlier led to a technical compromise where the length of the window, $w(n-m)$, was chosen to be 21.3ms (1024 samples). This compromise simplified the code greatly as the frame length was divisible by the window size. To see the advantage of having the frame length being divisible by the window length, consider first that during real time processing, the short time average magnitude calculation will require samples from either the future frame (not possible) or from the previous frame [4]. This can be seen in Fig. 2 where the software makes use of the previous 512 samples that were provided in the previous callback. Therefore, the number of samples saved for evaluation in the next frame of data is always constant. If the frame length was not divisible by the window size, then the number of previous samples used in the next group of short time average magnitude calculations would change with each frame. In this undesirable case, the software complexity would have to increase in order to keep track of how many samples from the past have to be used.

Fig. 2. Audio Frames Processing

B. User Interface

The user interface for this web application was designed in HTML5 and CSS in order to conform with the concept of developing the speech therapy software on an open web platform. An early concept of this interface can be seen in Fig. 3. In its most simplest form, the user has button interfaces to hear the *goal* speech utterance and to record his/her own speech utterance.

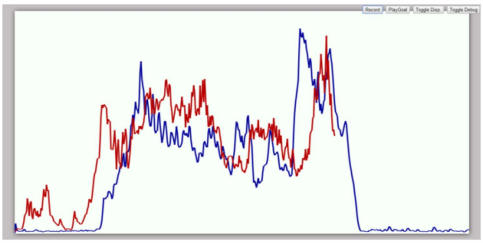

Fig. 3. Speech Therapy Application User Interface

As previously mentioned, the main visual cue given to the user is his/her short time average magnitude displayed in real time. In Fig. 3, this is seen by the red plotted signal. The signal plotted in blue is fixed to the display and represents the short time average magnitude of the *goal* speech utterance. While the user may not understand on a technical level the meaning behind the short time average magnitude, the user will quickly become aware of its relationship to his/her speech utterance as he/she practices within the tool. That is, the signal appears to increase when his/her voice increases and likewise with the rate of increase. This is the key to allowing the user to make alterations to his/her speech with each trial and thus improve upon his/her speech quality [3].

One of the advantages of using HTML5 and CSS is it's a very development friendly environment for simple graphics as needed with this speech therapy application. In the case of this application, the HTML5 canvas tag was used to draw on the screen. By carefully choosing the CSS attributes of the canvas tag, the JavaScript code need only to be concerned with fixed pixel dimensions as the browser will handle any scaling needed as the user adjusts the browser dimensions. The one caveat is that care must be taken to avoid pixilation. As an example, for the timing parameters given in this application (window length, sampling rate, speech duration, etc.), there will be 562 energy points to be plotted. However, the canvas length was chosen to be 1124 pixels wide because this width is closer to the width in pixels of modern displays. In this way, the energy points were plotted 2 pixels at a time by the JavaScript code. These considerations were made because it was noted that pixilation was more noticeable when allowing the browser to scale up a canvas with width of 512 pixels rather than scaling it down after plotting each point with 2 pixels on a canvas double the size.

IV. Conclusion

Developing a broadly accessible speech therapy software package on an open web platform was shown to be possible. Modern web browsers contain all of the APIs needed to make the speech therapy software a reality. The*Web Audio API* provides a developer a very easy to use API for processing audio in real time. The *Media Capture and Streams API* provides the developer with the needed access to the user's microphone. By combining these two APIs along with the graphical powers of HTML5 and JavaScript, a developer can make a web based rich application for speech therapy. By deploying the software on the web, the developers can couple said software with databases that will track the progress of a clinic's patients. In addition, the patients do not need to worry about installing software and ensuring that installations are compatible with their device's operating system. With continued development, clinical settings can make use of such software to track and manage a patient's progress even while the patient isn't physically in the clinic.

References

[1] Chris Rogers, *Web Audio API*, World Wide Web Consortium (W3C) specification, May 2014; https://dvcs.w3.org/hg/audio/raw-file/tip/webaudio/specification.html

[2] Daniel C. Burnett, et al., *Media Capture and Streams*, World Wide Web Consortium (W3C) specification, May 2014;http://dev.w3.org/2011/webrtc/editor/getusermedia.html

[3] Selim S. Awad, Louis Przebienda, and Richard Merson, "The Application of Digital Speech Processing to Stuttering Therapy," in *IEEE Instrumentation and Measurement Technology Conference*, Ottowa Canada, 1997, pp 1361-1367.

[4] Thad B. Welch, Cameron H.G. Wright, and Michael G. Morrow, *Real-Time Digital Signal Processing from Matlab®to C with the TMS320C6x DSK*, Boca Raton, Florida: CRC Press, 2006, pp 111-115

[5] Bloice, Marcus D, et al. Java's Alternatives and the Limitations of Java when Writing Cross-Platform Applications for Mobile Devices in the Medical Domain, in *Proceedings of the ITI 2009 31st Int. Conf. on Information Technology Interfaces* 2009, pp 47-54.

[6] Jacobs Ian, et al. (2012, Nov.). How the Open Web Platform is Transforming Industry. *Internet Computing, IEEE* 16(6), pp 82-86.

[7] Imbert Thibault, From microphone to .WAV with getUserMedia and Web Audio. *Typedarray.org*, [online] http://typedarray.org/from-microphone-to-wav-with-getusermedia-and-web-audio/ (Accessed: 2 Feb. 2014)

[8] Boris Smus, Getting Started with Web Audio API. *HTML5Rocks.com*, [online] http://www.html5rocks.com/en/tutorials/webaudio/intro/ (Accessed: 2 Feb. 2014)

[9] Eric Bidelman, Capturing Audio & Video in HTML5. *HTML5Rocks.com*, [online] http://www.html5rocks.com/en/tutorials/getusermedia/intro/ (Accessed: 2 Feb. 2014)

[10] M.L Gow, Modifying Phonation Interval Distributions During Solo and Chorus Reading: The Effect on Stuttering, Ph.D. dissertation, Dept. Speech and Hearing Sciences, Univ.of California, Santa Barbara, CA, 1998