A Dynamic Audio Experience creation platform in Web Audio

Chinmay Pendharkar
Sonoport Asia Pte. Ltd.
Blk 71 Ayer Rajah Crescent
Ayer Rajah Industrial Estate,
Singapore
chinmay.pendharkar@sonoport.com

Peter Bäck
Sonoport Asia Pte. Ltd.
Blk 71 Ayer Rajah Crescent
Ayer Rajah Industrial Estate,
Singapore
peter.back@sonoport.com

Lonce Wyse
Communication and New Media Department
National University of Singapore
11 Computing Drive,
Singapore
lonce.wyse@nus.edu.sg

ABSTRACT

This paper describes innovative aspects of the Sonoport Studio, a dynamic audio experience creation platform which will be presented in a demonstration at the 1st Web Audio Conference in Paris.

Two key components of the platform are 1) Sound Model templates that represent classes of dynamic sonic behaviors, and 2) Interaction Model templates that represent classes of interactive behavior that are commonly mapped to interactive sound elements.

These components, together with a database of sound files and other authoring tools, comprise an interactive audio experience creation platform which is aimed at a typical creative web designer/developer who wants to take advantage of the new capabilities offered by the Web Audio platform, but may not have a deep synthesis or audio processing background.

Categories and Subject Descriptors

H.5.5 [**Information Systems**]: Information Interfaces and Presentation (HCI)—*sound and music computing*

General Terms

Design

Keywords

Interactive Audio, W3C Web Audio API, Audio Synthesis

1. INTRODUCTION

The modern Web platform has drastically changed the capacity of the Web to deliver rich, interactive, experiences to millions of users. With technologies such as, Web Sockets, Web RTC, Web GL and Web Audio, modern browsers can deliver audio-visual experiences which can be interacted with using traditional mouse and keyboards as well as various new sensors such as DeviceOrientation [3], Geolocation [2], Voice and Camera.

With such a wide range of possibilities for audio content creation on one hand, and interactive user behaviors on the other, it can be challenging to create an application experience that feel intuitive and natural.

As we ported Sonoport's Dynamic Sound Engine to the Web platform [1], we wanted to address not only the web developers' need to access a rich sonic palette, but also to support the web developers' creation of interactive web audio experiences including interaction design.

In the set of demonstrations we will go through some of the novel ideas and approaches we designed into this platform to help developers easily connect user interactions to synthesized audio.

2. WEB PLATFORM AND COMPLEXITY

The openness and acessibility of the Web platform has attracted many new developers from other platforms. As the Web platform evolved and modernized it incorporated many new technologies that had previously only been available in native code contexts. These technologies, such as Web Audio and WebGL brought along their own semantics and knowledge base into the Web platform. To be able to effectively use these new technologies, a typical web developer has to learn the semantics, and techniques of these technologies with respect to the Web platform.

For example, let's consider a scenario of adding a 'swoosh' sound when someone interacts with a specific HTML `div` tag. Detecting when a user moves the pointer over a specific HTML Element is straightforward using `DOMEvents`. However adding the interactivity with audio is significantly more complex. It would involve creating an `AudioContext`, downloading and decoding the audio file, creating `AudioBuffersSourceNodes`, starting and stopping on mouseover events. And to make it sound natural, one would have to add a `GainNode` and add some basic ADSR Envelope functionality to ease in and out the volume of the sound. All this just for a simple mouse-over sound effect.

Furthermore, if the audio itself needs to be processed differently based on different interaction paradigms, for example changing the pitch of a sound based on where a user clicks, it would require the developers to have some understanding of audio digital signal processing.

To look at how this can be made easier for typical web developers we analyzed the various dynamic aspects of the system needed to create such a rich dynamic audio experience. To address the two critical areas of dynamics we found (interactive sound and interactive behavior), we created frameworks that help manage the complexity of design. Our demonstrations will highlight these frameworks and how they can be used to create a complete dynamic audio experience authoring platform.

3. SOUND MODEL TEMPLATES

We are used to interacting with physical things in the world around us. These things make sounds when we interact with them. Our expectations about objects, interaction, and sound are transferred to the virtual world of objects and interactions that we experience in computer and web-based media.

Some of the common ways sounds change, or more specifically how we expect sounds to change with interaction can be embodied in parameterized algorithms which generate those sounds in real time. These algorithms, called Sound Models [5], can be purely synthetic or make use of audio files [4] to create parameterizable sonic textures and events.

Algorithmic similarities in various Sound Models can be leveraged to create templates that can produce multiple Sound Models. These Sound Model templates produce a Sound Model based on the audio files and the initial parameter values that are used.

One example of a Sonoport Sound Model template is *Trigger*. *Trigger* employs a polyphonic queue to generate a Sound Model of something being triggered by a single user interaction. It could be a Sound Model of a gun in a video game, a drum hit or just a Sound Model creating a sound effect of clicking on a button. However, such a Sound Model is commonly expected to play again if it is triggered repeatedly. More importantly if the sound caused by the previous interaction hasn't finished playing yet, the next user interaction can still start playing a new sound. This behaviour is encoded into a *Trigger* Sound Model template along with parameters to change certain properties such as the pitch of the sound.

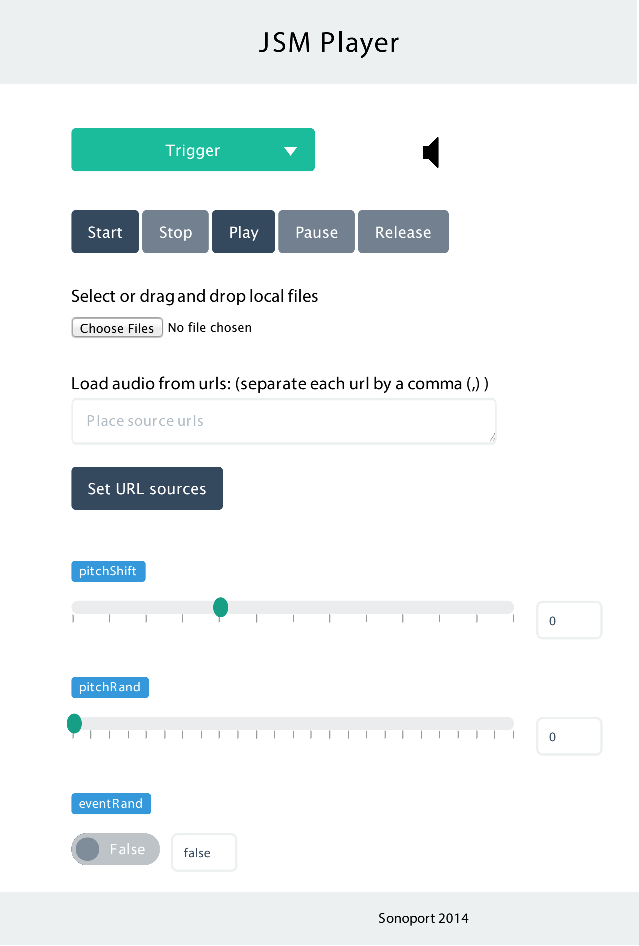
Sound Model templates encode many of the common ways we expect sounds to behave and take care of all the complexities of scheduling, polyphony, subtle variation, and enveloping for the user. Sound Models created by these templates are also designed to be extensible via composition with other Web Audio AudioNodes. This allows them to be chained with other AudioNodes for manually adding specific effects or processing beyond that created by the Sound Model itself.

We have created a set of Sound Model templates which can be used to then create rich, interactive and dynamic sound experiences. We will demonstrate the working of these Sound Model templates using the Sonoport *JSM Player*, a Web interface built to develop and test Sound Models (Figure 1) 1.

4. INTERACTION MODEL TEMPLATES

While Sound Models embody the dynamic audio processes, Interaction Models embody the user interaction patterns. We considered the various types of interactions we had to create for the various demonstrations and interactive media projects we had done in the past and looked at common patterns which could be encoded into some form of templates, we called these Interaction Model templates.

The usual mouse and keyboard combination of user interfaces gives us a surprisingly varied and rich set of interactions. Interaction Models using these interfaces include everything from simple point and click, to hovering over an object on screen, to dragging on screen objects to moving the pointer location with respect to other objects. Some of these interactions may not be explicitly supported natively by the Web platform and have to be gleaned from a combination of multiple user interface events. Abstracting this complexity out makes adding interactions to audio a lot simpler for web developers.



Select or drag and drop local files

No file chosen

Load audio from urls: (separate each url by a comma (,))

Place source urls

pitchShift: 0

pitchRand: 0

eventRand: false

Sonoport 2014

Figure 1: A Web based interface to Sound Models

4.1 Interaction Levels

The first step was to categorize the various types of Interaction Models in terms of complexity of the interactions involved. Since finally these were going to be templates that web developers would use in their own projects, having sorted by complexity would make it intuitive. We created levels of interactions as defined in Table 1.

For each of these *Interaction Levels*, we created Interaction Model templates that sensed user interaction events based on the corresponding Interaction Model and affected a Sound Model based on these events. A critical part of creating these templates was to associate Interaction Models with Sound Models which were complementary to each other. For example, a *Tap* or *Click* Interaction Models is compatible with the Trigger Sound Model template described in section 3. In many scenarios we found multiple Sound Model tem-

Table 1: Levels of Interaction Models

Level	Description	Example
1	very basic interactions with an onscreen object	tap, click, hover over
2	events being triggered for each time the mouse was moved	mouse-move, mouse-position,
3	onscreen object and their relation with the mouse or pointer	drag an object, move about an object
4	physical device orientation and location, external inputs like voice	device-rotate, device-move

plates being compatible with a specific Interaction Model. The user of an Interaction Model template is given a choice among all the Sound Model templates that were deemed compatible for that specific Interaction Model template.

The Interaction Model templates also setup default audio sources or textures and parameters for each compatible Sound Models templates so loading a new Interaction Model template allows a web developers to immediately experience the type of interactive audio that can be created using that specific template. The sounds and parameters can then be changed to meet the specific sonic requirements of the web developer.

These templates are created using Web technologies such as HTML5 Canvas, and have a standardized metadata API for packaging and loading. This makes them portable and extensible, and allows easy creation of Interaction Model templates by third parties.

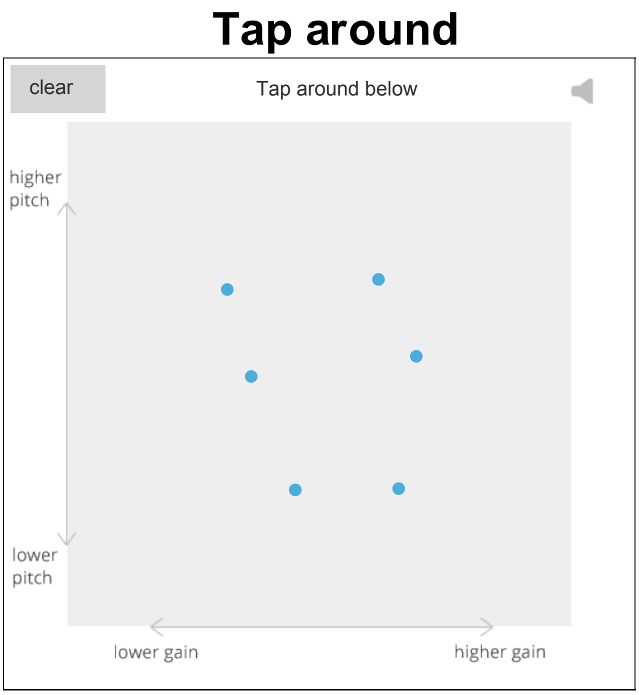


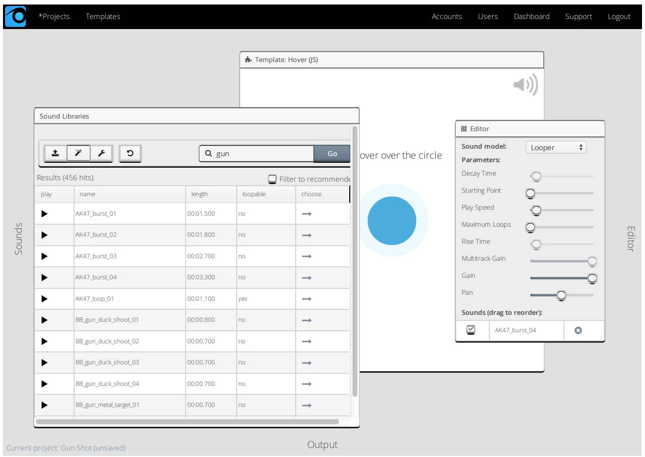
Figure 2: A Web based interface to Interaction Model Templates

We will demonstrate a set of *Level 1* and *Level 2* Interaction Model templates that we have created for mouse and touch based interactions. Figure 2. shows the Web Interface used for testing and tweaking these templates that we will be using to demonstrate this framework.

5. AUTHORING PLATFORM

We integrated Sound Model templates, Interaction Model templates and other novel ideas such as control mappings into the web application called Sonoport Studio. The Sonoport Studio helps web developers quickly create an interactive audio experience using dynamic Sound Model templates and Interaction Model templates using a graphical user interface to change parameters, values, mappings and sounds.

While the Sound Model and Interaction Model templates are abstract representations of the dynamics of sound and interaction, the Sonoport Studio itself is designed to help improve a web developer's workflow for creating interactive audio experiences. Figure 3. shows current version of the Web interface of the Sonoport Studio.



Results (456 hits)

play	name	length	loopable	choose
▶	AK47_burst_01	00:01.500	no	→
▶	AK47_burst_02	00:01.800	no	→
▶	AK47_burst_03	00:02.700	no	→
▶	AK47_burst_04	00:03.300	no	→
▶	AK47_loop_01	00:01.100	yes	→
▶	BB_gun_duck_shoot_01	00:00.800	no	→
▶	BB_gun_duck_shoot_02	00:00.700	no	→
▶	BB_gun_duck_shoot_03	00:00.700	no	→
▶	BB_gun_duck_shoot_04	00:00.700	no	→
▶	BB_gun_metal_target_01	00:00.700	no	→

Sound model: Looper

Parameters:
Decay Time
Starting Point
Play Speed
Maximum Loops
Rise Time
Multitrack Gain
Gain
Pan

Sounds (drag to reorder):
☑ AK47_burst_04

Current project: Gun Shot (unsaved)

Figure 3: Sonoport Studio authoring platform

The Sonoport Studio ties in with audio repositories to allow search, preview and importing of audio files for Sound Models. These audio files can then be tested with various Interaction Models and Sound Models. Sonoport Studio also has Web interfaces to manipulate Sound Model parameters, mappings between Interactions Models and Sound Model parameters to customize the experience for the intended use case.

Once developers have created an experience that they are comfortable with the project can be exported either as raw code, or as files or projects for HTML5 visual design tools such as Google Web Designer or Adobe Edge.

While the Sonoport Studio is still in the throes of development, we will demonstrate a working version of the Sonoport Studio capable of using the underlying frameworks of Sound Model and Interaction Model templates to create a dynamic sound experience.

6. CONCLUSIONS

Web Audio holds the promise of allowing rich, interactive audio experiences on the Web. However, it's complexity and semantics make it difficult for web developers without specific domain knowledge to use it effectively. Using abstractions of dynamism in sound in the form of Sound Model templates and of dynamism in user interaction in the form of Interaction Models templates, we demonstrate a platform for authoring such rich, interactive audio experiences on the Web. Such a platform can make it easy to create an application where interactions generate audio that sounds natural and intuitive.

7. ACKNOWLEDGMENTS

This paper/demo is based on technology developed at Sonoport, www.sonoport.com.

8. REFERENCES

[1] C. Pendharkar, P. Bäck, and L. Wyse. Adventures in scheduling, buffers and parameters : Porting a dynamic audio engine to webaudio. In *Proceedings of the Web Audio Conference*, Paris, France, 2015.

[2] A. Popescu. Geolocation api specification. `http://dev.w3.org/geo/api/spec-source.html`, July 2014.

[3] R. Tibbett, T. Volodine, S. Block, and A. Popescu. Deviceorientation event specification. `http://w3c.github.io/deviceorientation/spec-source-orientation.html`, March 2014.

[4] L. Wyse and P. R. Kellock. Embedding interactive sounds in multimedia applications. *Multimedia Syst.*, 7(1):48–54, 1999.

[5] L. L. Wyse. Generative sound models. *Proceedings of the Multimedia Modeling 2005 Conference*, pages 370–377, 2005.