Table of Contents

1 Efforts to Improve X3D Audio

1.1 Audio in X3D ISO

Spatial sound has a conceptual role in the Web3D environments, due to highly realism scenes that can provide. Since Web Audio API is the most popular sound engine, we propose to get the necessary steps required to make X3D fully compatible with this library.

Particularly, two types of nodes are included in the X3D, the first is about the sound description and the other one for the sound source. Specifically, the first node is the **X3DSoundNode**, which is an abstract node for all sound nodes. It is minimalist with only one attribute, *metadata*, which expresses important information for the significance, appearance and the proposed role of the model. The second node is the **X3DSoundSourceNode**, which is the abstract node for each node that is used to emit sound and it has a number of common fields with the *TimeSensor*, for example the *loop*, the *startTime*, the *stopTime*, the *pauseTime* and *resumeTime* (see Figure 1).

X3DSoundNode

```
X3DSoundNode : X3DChildNode {
  SFNode [in,out] metadata NULL [X3DMetadataObject]
}
```

X3DSoundSourceNode

```
X3DSoundSourceNode : X3DTimeDependentNode {
  SFString [in,out] description      ""
  SFBool   [in,out] loop             FALSE
  SFNode   [in,out] metadata         NULL [X3DMetadataObject]
  SFTime   [in,out] pauseTime        0    (-∞,∞)
  SFFloat  [in,out] pitch            1.0  (0,∞)
  SFTime   [in,out] resumeTime       0    (-∞,∞)
  SFTime   [in,out] startTime        0    (-∞,∞)
  SFTime   [in,out] stopTime         0    (-∞,∞)
  SFTime   [out]    duration_changed
  SFTime   [out]    elapsedTime
  SFBool   [out]    isActive
  SFBool   [out]    isPaused
}
```

Figure 1: X3DSoundNode - X3DSoundSourceNode in X3D

The third node is the **Sound**, which is derived from the X3DSoundNode. It is designed for the description of the X3D scene sounds. Specifically, it determines both the location and the behaviour of the sound. Additionally, the geometry describes that the sound can be directed and be emitted in an elliptical pattern. Two ellipsoids constitute the pattern, which specifies the borders for level of loudness of the sound. Also, ellipsoids can be reshaped in order to provide more or less directional focus from the location of the sound. Consequently, the sound node is intended to recognize the source and is related to the direction, the location, the priority and general, the spatial features of the sound source (see Figure 2).

Sound

```
Sound : X3DSoundNode {
  SFVec3f [in,out] direction  0 0 1 (-∞,∞)
  SFFloat [in,out] intensity  1     [0,1]
  SFVec3f [in,out] location   0 0 0 (-∞,∞)
  SFFloat [in,out] maxBack    10    [0,∞)
  SFFloat [in,out] maxFront   10    [0,∞)
  SFNode  [in,out] metadata   NULL  [X3DMetadataObject]
  SFFloat [in,out] minBack    1     [0,∞)
  SFFloat [in,out] minFront   1     [0,∞)
  SFFloat [in,out] priority   0     [0,1]
  SFNode  [in,out] source     NULL  [X3DSoundSourceNode]
  SFBool  []       spatialize TRUE
}
```

AudioClip

```
AudioClip : X3DSoundSourceNode, X3DUrlObject {
  SFString [in,out] description      ""
  SFBool   [in,out] loop             FALSE
  SFNode   [in,out] metadata         NULL [X3DMetadataObject]
  SFTime   [in,out] pauseTime        0    (-∞,∞)
  SFFloat  [in,out] pitch            1.0  (0,∞)
  SFTime   [in,out] resumeTime       0    (-∞,∞)
  SFTime   [in,out] startTime        0    (-∞,∞)
  SFTime   [in,out] stopTime         0    (-∞,∞)
  MFString [in,out] url              []   [URI]
  SFTime   [out]    duration_changed
  SFTime   [out]    elapsedTime
  SFBool   [out]    isActive
  SFBool   [out]    isPaused
}
```

Figure 2: Sound-AudioClip in X3D

The fourth node is the **AudioClip**, which is derived from the X3DSoundSourceNode. It specifies audio data that can be referenced by Sound nodes. Basically, it loads an external audio file with a view to handle playing, stopping and starting (see Figure 2). As regard the attributes, AudioClip has a number of fields in common with TimeSensor, because it is an X3DSoundSourceNode and implements the X3DTimeDependentNode abstract type. Basically, the fields of the sound nodes and their interrelation are presented in Figure 3 by an interpretive hierarchic diagram.

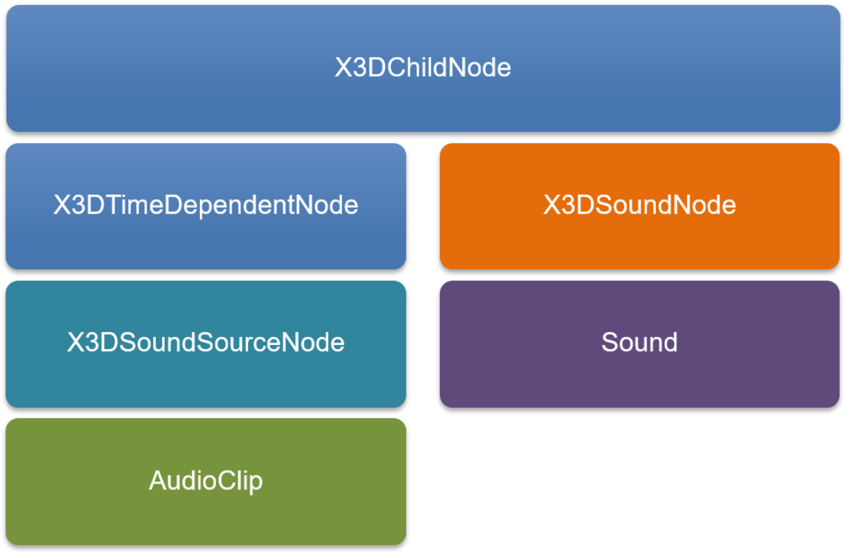


Figure 3: Inheritance diagram of Audio nodes in X3D

1.2 Background Work Using X3D Sound Components

A method for the introduction of spatial sound component in the X3DOM framework, based on X3D specification and Web Audio API has already implemented. The proposed method incorporates the introduction of enhanced sound nodes for X3DOM which are derived by the implementation of the X3D standard components, enriched with accessional features of Web Audio API. Particularly, with this approach the complex sound design and implementation is becoming transparent to the programmer and moreover the applications are independent of the sound libraries are employed.

The development consists of the registration of Web Audio API and custom components into X3DOM framework. All these nodes are used by HTML DOM, which is essentially an X3D scene, directly from the X3DOM and no through JavaScript structure. On the other hand, JavaScript code, which is indicated as JavaScript Controller, accords all the indispensable instruction of nodes design and role. It interacts with the HTML file, for the purpose of parsing the 3D scene and being updated on any potential change in the scene.

Below is the list of the implementation of the major X3D sound nodes (X3DSoundSourceNode: AudioClip and X3DSoundNode: Sound) using Web Audio API functionality.

Table 1: X3DSoundSourceNode: AudioClip

Node	X3D	Our Approach	
X3DSoundSourceNode: AudioClip	SFString **description**	SFString **description**	WebAudioAPI:**AudioSource**
	SFBool **loop**	SFBool **loop**	WebAudioAPI:**AudioSource**
	SFNode **metadata**	SFNode **metadata**	WebAudioAPI:**AudioSource**
	SFTime **pauseTime**	SFTime **pauseTime**	WebAudioAPI:**AudioSource**
	SFFloat **pitch**	SFFloat **pitch**	WebAudioAPI:**AudioSource**
	SFTime **resumeTime**	SFTime **resumeTime**	WebAudioAPI:**AudioSource**
	SFTime **startTime**	SFTime **startTime**	WebAudioAPI:**AudioSource**
	SFTime **stopTime**	SFTime **stopTime**	WebAudioAPI:**AudioSource**
	MFString **url**	MFString **url**	WebAudioAPI:**AudioSource**
	SFTime **duration_changed**		
	SFTime **elapsedTime**		
	SFBool **isActive**		
	SFBool **isPaused**		

Table 2: X3DSoundNode: Sound

Node	X3D	Our Approach	
X3DSoundNode: Sound	SFVec3f **location**	SFVec3f **position**	Web Audio API:**PannerNode**
	SFVec3f **direction**	SFRotation **orientation**	Web Audio API:**PannerNode**
	SFFloat **maxBack** SFFloat **maxFront** SFFloat **minBack** SFFloat **minFront**	SFFloat **coneInnerAngle**	Web Audio API:**PannerNode**
		SFFloat **coneOuterAngle**	Web Audio API:**PannerNode**
		SFFloat **coneOuterGain**	Web Audio API:**PannerNode**
		SFString **distanceModel**	Web Audio API:**PannerNode**
		SFFloat **maxDistance**	Web Audio API:**PannerNode**
		SFString **panningModel**	Web Audio API:**PannerNode**
		SFFloat **refDistance**	Web Audio API:**PannerNode**
		SFFloat **rolloffFactor**	Web Audio API:**PannerNode**
	SFFloat **intensity**	SFInt32 **gain**	Web Audio API: **GainNode**
	SFFloat **priority**	SFFloat **priority**	Web Audio API: **DynamicsCompressorNode**
	SFNode **source**	SFNode **source**	WebAudioAPI:**AudioSource**
	SFNode **metadata**		
	SFBool **spatialize**		
		SFVec3f **velocity (new)**	Web Audio API:**PannerNode**

In order to verify the validity of our method, several experiments/examples were carried out, in which new registered nodes were used (http://medialab.teicrete.gr/minipages/x3domAudio/index.html). These tests have been investigated with the use of Google Chrome 41.0, Firefox Mozilla 36.0 and Opera 28.0.

1.3 Proposal to Improve X3D Sound Component – Introduction of Web Audio API components

The X3D comprises a fundamental part in the 3D web development. At the same time, the spatial sound should be an integral part of an immersive 3D application, due to the fact that it can improve the sense of realism and the immersion both in complex acoustic environments and dynamic virtual scenes. Based on that, we propose the enrichment of X3D with spatial sound features, using the structure and the functionality of Web Audio API.

Particularly, the Web Audio API involves handling audio operations inside an audio context (AudioContext node) and has been designed to allow modular routing. Particularly, the approach of Web Audio API is based on the concept of audio context, which presents the direction of audio stream flows, between sound nodes (AudioNode).

A simple, typical workflow for web audio API[1] is presented in Figure 4.

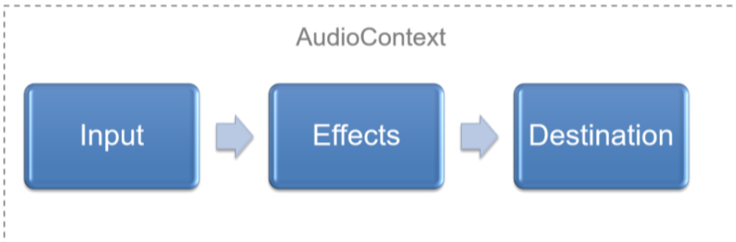
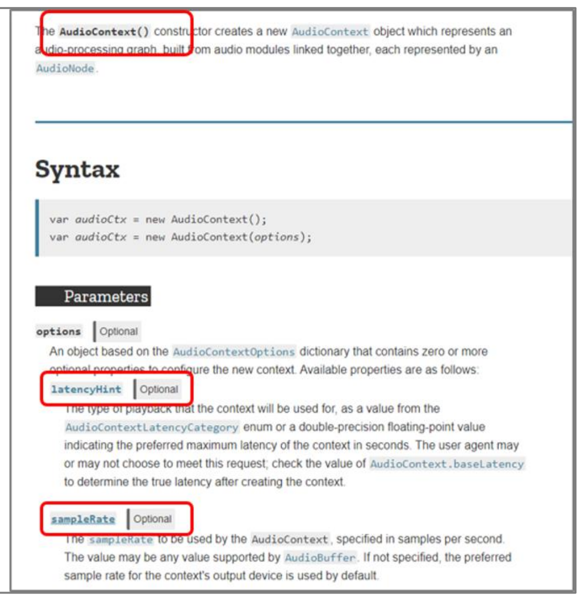


Figure 4: A simple example of modular routing[1]

Taking to account all the above together, the proposed method is the registration of a new node, under the X3D Sound node, with the name AudioContext. This node can be used as parent of the AudioNode, which can have all the supported interfaces/nodes of Web Audio API (Figure 5 and Figure 6).

AudioContext*: Sound {*
SFNode [in,out] metadata NULL [X3DMetadataObject]
SFDouble [in, out] latencyHint 0.0 (-∞, ∞) //optional or
SFString [in,out] latencyHint " "
SFFloat [in,out] sampleRate 0 (-∞, ∞) //optional

}



The AudioContext() constructor creates a new AudioContext object which represents an audio-processing graph, built from audio modules linked together, each represented by an AudioNode.

Syntax

```
var audioCtx = new AudioContext();
var audioCtx = new AudioContext(options);
```

Parameters

options Optional
An object based on the AudioContextOptions dictionary that contains zero or more optional properties to configure the new context. Available properties are as follows:

latencyHint Optional
The type of playback that the context will be used for, as a value from the AudioContextLatencyCategory enum or a double-precision floating-point value indicating the preferred maximum latency of the context in seconds. The user agent may or may not choose to meet this request; check the value of AudioContext.baseLatency to determine the true latency after creating the context.

sampleRate Optional
The sampleRate to be used by the AudioContext, specified in samples per second. The value may be any value supported by AudioBuffer. If not specified, the preferred sample rate for the context's output device is used by default.

Figure 5: Registration of AudioContext node (Web Audio API) to X3D (link)

[1] https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

```
AudioNode: AudioContext {
  SFNode  [in,out]  metadata  NULL  [X3DMetadataObject]
  SFInt32 [in,out]  numberOfInputs  0
  SFInt32 [in,out]  numberOfOutputs  0
  SFString [in,out]  channelCount 0
  SFString [in,out]  channelCountMode max
  SFString [in,out]  channelInterpretation speakers
}
```

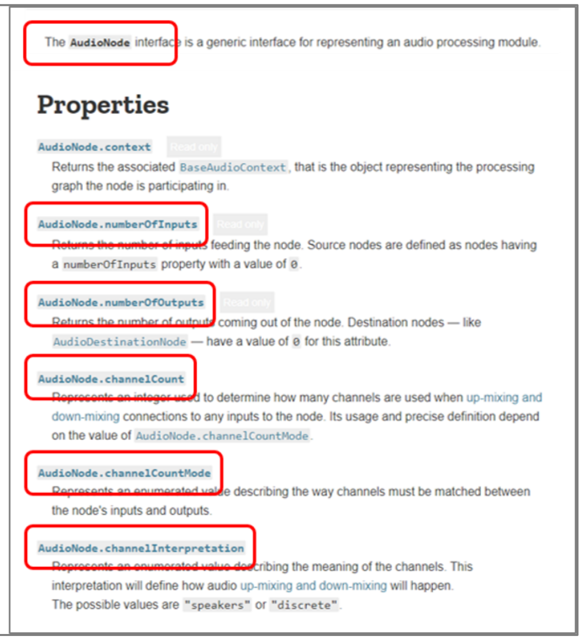


Figure 6: Registration of AudioNode node (Web Audio API) to X3D (link)

All the other interfaces of the Web Audio API (link) could be derived from the AudioNode (Figure 7).

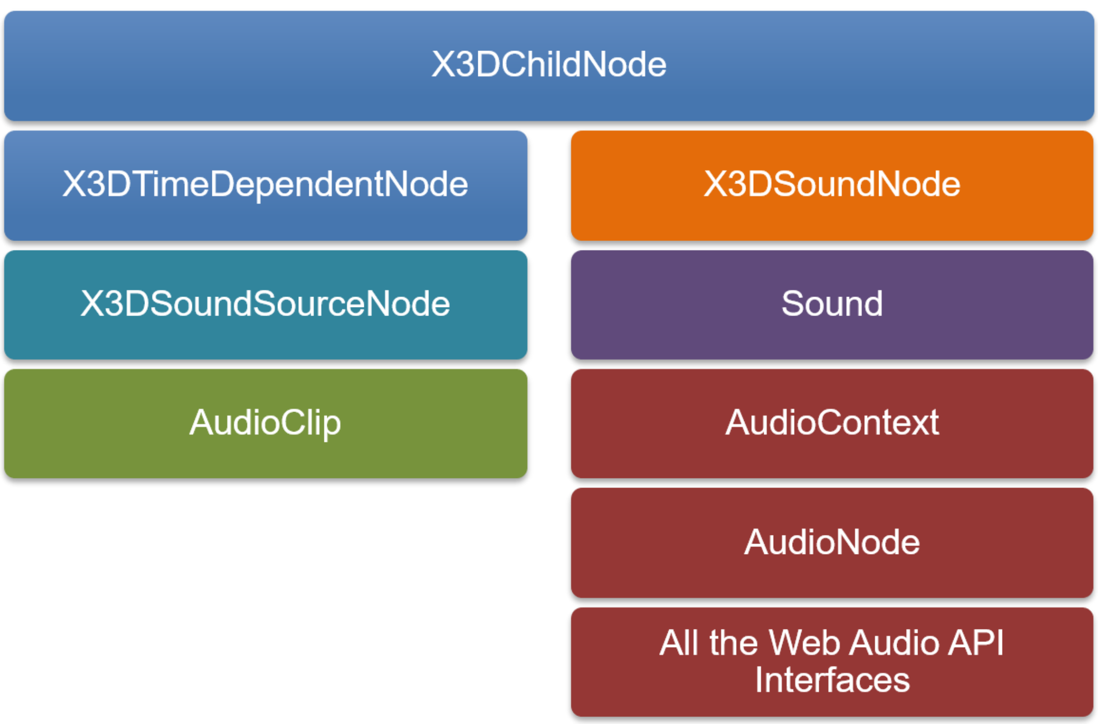


Figure 7: Proposal: Inheritance diagram of enhanced Audio nodes in X3D

1.4 Revised Work Using X3D Sound Components based on the Above Proposal to Improve X3D Sound Component

This section presents the revisions in the implementation of examples/experiments that were presented in Section 1.2. Based on the structure, which is highlighted in Figure 7, the implementation includes:

- the registration of the X3D core Sound Nodes to match with the proposal that has already described.

Sound Node (X3D)

```
x3dom.registerNodeType("Sound", "X3DSoundNode", defineClass(x3dom.nodeTypes.X3DSoundNode,
function(d) {
  x3dom.nodeTypes.X3DSoundNode.superClass.call(this, d),
  this.addField_SFNode("metadata", x3dom.nodeTypes.X3DMetadataObject),
  this.addField_SFVec3f(d, "direction", 0, 0, 1),
  this.addField_SFFloat(d, "intensity", 1),
  this.addField_SFVec3f(d, "location", 0, 0, 0),
  this.addField_SFFloat(d, "maxBack", 10),
  this.addField_SFFloat(d, "maxFront", 10),
  this.addField_SFFloat(d, "minBack", 1),
  this.addField_SFFloat(d, "minFront", 1),
  this.addField_SFFloat(d, "priority", 0),
  this.addField_SFNode("source", x3dom.nodeTypes.X3DSoundSourceNode),
  this.addField_SFBool(d, "spatialize", !0)

  //New attributes
  //In order to use it in html under the Sound node of X3D
  this.addField_SFNode("transform", x3dom.nodeTypes.Transform),
  this.addField_SFNode("panner", x3dom.nodeTypes.PannerNode),
  this.addField_SFNode("filter", x3dom.nodeTypes.BiquadFilterNode),
  this.addField_SFNode("delay", x3dom.nodeTypes.DelayNode)
}
```

X3DSoundSourceNode(X3D)

```
x3dom.registerNodeType("X3DSoundSourceNode", "X3DTimeDependentNode",
defineClass(x3dom.nodeTypes.X3DSoundNode, function(d) {
  x3dom.nodeTypes.X3DSoundSourceNode.superClass.call(this, d),
  this.addField_SFString(d, "description", ""),
  this.addField_SFBool(d, "loop", !1),
  this.addField_SFNode("metadata", x3dom.nodeTypes.X3DMetadataObject),
  this.addField_SFTime(d, "pauseTime", 0),
  this.addField_SFFloat(d, "pitch", 1),
  this.addField_SFTime(d, "resumeTime", 0),
  this.addField_SFTime(d, "startTime", 0),
  this.addField_SFTime(d, "stopTime", 0)

  //New attributes
  this.addField_MFString(d, "url", [])//From AudioClip X3D Node
  this._audio = document.createElement("audio"), "Microsoft Internet Explorer" != navigator.appName &&
document.body.appendChild(this._audio)
}
```

- the registration of the Web Audio API core Sound Nodes to match with the proposal that has already described

AudioContext: For managing and playing all sounds. Like a graph with all nodes

```
x3dom.registerNodeType("AudioContext", "Sound", defineClass(x3dom.nodeTypes.X3DSoundNode,
function(d) {
  x3dom.nodeTypes.X3DSoundNode.superClass.call(this, d),
  this.addField_SFFloat(d, "currentTime", 0),
  this.addField_SFNode("destination", x3dom.nodeTypes.AudioDestinationNode),
  this.addField_SFNode("listener", x3dom.nodeTypes.AudioListener),
  this.addField_SFFloat(d, "sampleRate", 0),
  this.addField_SFNode("mozAudioChannelType", x3dom.nodeTypes.AudioListener)
}
```

AudioNode: For every Node in the graph, if numberOfInputs then it is a source node

```
x3dom.registerNodeType("AudioNode", "AudioContext", defineClass(x3dom.nodeTypes.X3DSoundNode,
function(d) {
   x3dom.nodeTypes.X3DSoundNode.superClass.call(this, d),
   this.addField_SFNode("context", x3dom.nodeTypes.AudioContext),
   this.addField_SFInt32(d, "numberOfInputs", 0),
   this.addField_SFInt32(d, "numberOfOutputs", 0),
   this.addField_SFInt32(d, "channelCount", 0),
   this.addField_SFString(d, "channelCountMode", "max"),
   this.addField_SFString(d, "channelInterpretation", "speakers")
}
```

PannerNode: Describe its position with right-hand Cartesian coordinates, its movement using a velocity vector and its directionality using a directionality cone

```
x3dom.registerNodeType("PannerNode", "AudioNode", defineClass(x3dom.nodeTypes.X3DSoundNode,
function(d) {
   x3dom.nodeTypes.X3DSoundSourceNode.superClass.call(this, d),
   this.addField_SFVec3f(d, "position", 0, 0, 0),
   this.addField_SFRotation(d, "orientation", 0, 0, 1, 0),
   this.addField_SFVec3f(d, "velocity", 0, 0, 0),
   this.addField_SFFloat(d, "coneInnerAngle", 360),
   this.addField_SFFloat(d, "coneOuterAngle", 360),
   this.addField_SFFloat(d, "coneOuterGain", 0),
   this.addField_SFString(d, "distanceModel", "inverse"),
   this.addField_SFFloat(d, "maxDistance", 1e4),
   this.addField_SFString(d, "panningModel", "HRTF"),
   this.addField_SFFloat(d, "refDistance", 1),
   this.addField_SFFloat(d, "rolloffFactor", 1)
}
```

BiquadFilterNode: Different kinds of filters, tone control devices or graphic equalizers

```
x3dom.registerNodeType("BiquadFilterNode", "AudioNode",
defineClass(x3dom.nodeTypes.X3DSoundNode, function(d) {
   x3dom.nodeTypes.X3DSoundNode.superClass.call(this, d),
   this.addField_SFInt32(d, "frequency", 0),
   this.addField_SFFloat(d, "detune", 0),
   this.addField_SFFloat(d, "Q", 0),
   this.addField_SFFloat(d, "gain", 0),
   this.addField_SFString(d, "type", "lowpass")
}
```

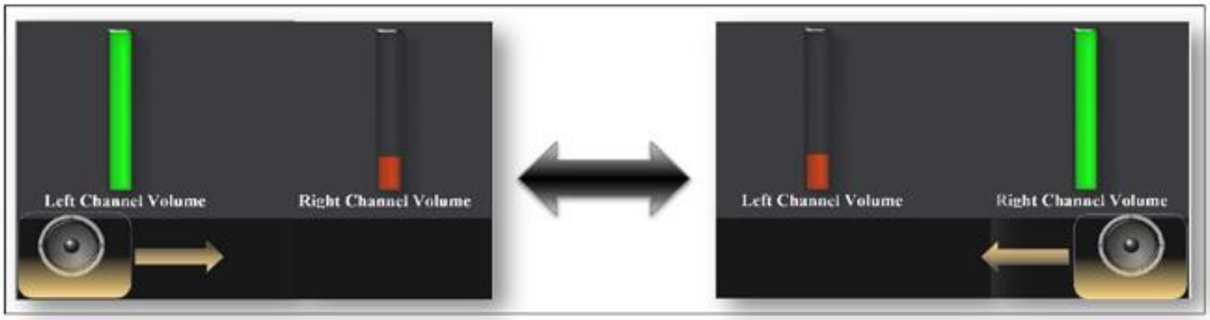
DelayNode: A delay between the arrival of an input data and its propagation to the output

```
x3dom.registerNodeType("DelayNode", "AudioNode", defineClass(x3dom.nodeTypes.X3DSoundNode,
function(d) {
   x3dom.nodeTypes.X3DSoundNode.superClass.call(this, d),
   this.addField_SFInt32(d, "delayTime", 0)
}
```

- the development of JavaScript Controller (wrapper) which is responsible to detect/recognize for Sound element in html file. Also, there is code for the creation of both a 3D sound object and an Array with the new registered nodes. Finally, the 3D sound object is updated and exchanges data with html DOM.

Hence, some of the examples/experiments (and the results of them) can be described by the following figures/Code:

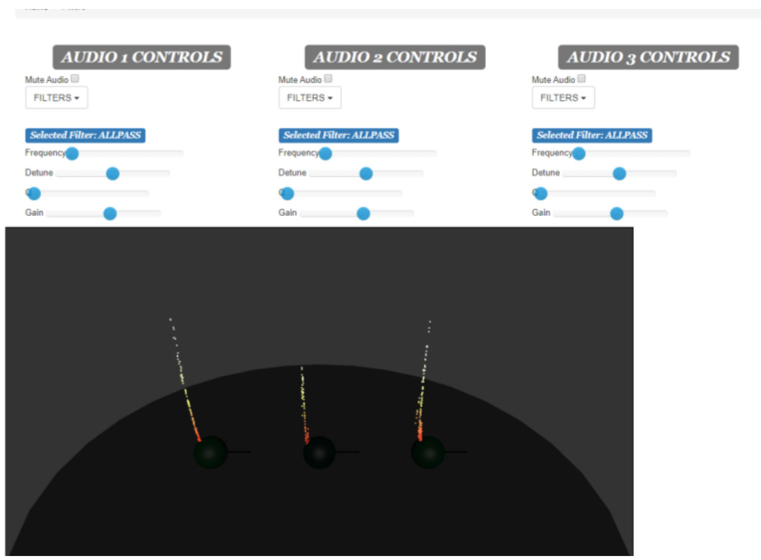
<u>Spatial sound with a sound source which moves in the scene (right left)</u>




```
<Sound>
    <Transform USE='Audio3'/>
    <PannerNode position='0 0 0' orientation='1 0 0' velocity='0 0 0' coneInnerAngle='360'
    coneOuterAngle='360' coneOuterGain='0' distanceModel='inverse' maxDistance='10000'
    panningModel='HRTF' refDistance='1' rolloffFactor='1'></PannerNode>
    <X3DSoundSourceNode loop='true' url='"sound/violin.mp3" "sound/violin.ogg"' pitch='1.0' pauseTime='-1' resumeTime='-1' startTime='0' stopTime="-1"/>
</Sound>
```

Figure 8: Example 1

<u>Filters</u>



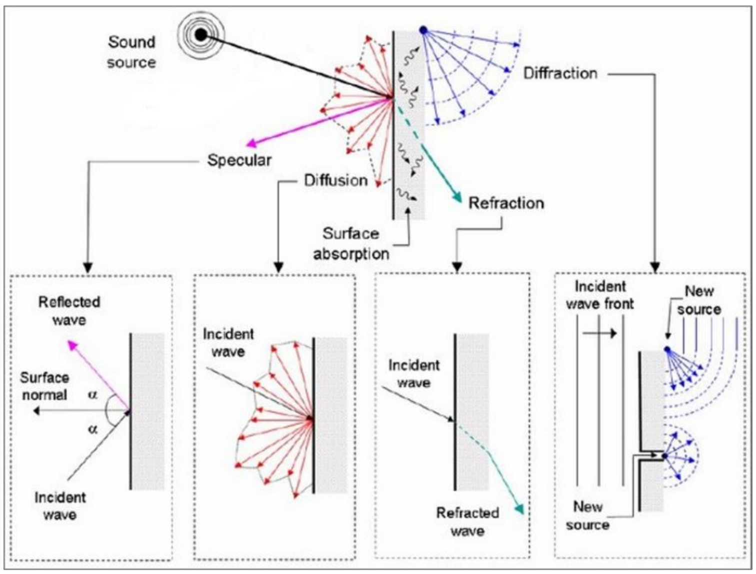

```
<Sound>
    <Transform USE='Audio2'/>
    <PannerNode position='0 0 0' orientation='1 0 0' velocity='0 0 0' coneInnerAngle='360' coneOuterAngle='360' coneOuterGain='0' distanceModel='inverse' maxDistance='10000' panningModel='HRTF' refDist
    <X3DSoundSourceNode loop='true' url='"sound/beat.mp3"' pitch='1.0' pauseTime='-1' resumeTime='-1' startTime='0' stopTime="-1"/>
    <BiquadFilterNode frequency='600' detune='15.0' Q='15.0' gain='1.0' type='allpass' />

</Sound>

<Sound>
    <Transform USE='Audio1'/>
    <PannerNode coneInnerAngle='360' coneOuterAngle='360' coneOuterGain='0' distanceModel='inverse' maxDistance='10000' panningModel='HRTF' refDistance='1' rolloffFactor='1'></PannerNode>
    <X3DSoundSourceNode loop='true' url='"sound/techno-beat.mp3"' pitch='1.0' pauseTime='-1' resumeTime='-1' startTime='0' stopTime="-1"/>
    <BiquadFilterNode frequency='600' detune='50.0' Q='30.0' gain='1.0' type='allpass' />
</Sound>



<Sound>
    <Transform USE='Audio3'/>
    <PannerNode position='0 0 0' orientation='1 0 0' velocity='0 0 0' coneInnerAngle='360' coneOuterAngle='360' coneOuterGain='0' distanceModel='inverse' maxDistance='10000' panningModel='HRTF' refDist
    <X3DSoundSourceNode loop='true' url='"sound/wobble-loop.mp3"' pitch='1.0' pauseTime='-1' resumeTime='-1' startTime='0' stopTime="-1"/>
    <BiquadFilterNode frequency='1000' detune='0.0' Q='0' gain='1.0' type='allpass' />
</Sound>
```

Figure 9: Example 2

2 Extended Efforts in X3D Audio

Many acoustic effects including surface reflection, physical phenomena such as interference and diffraction, the absorption, coefficient of materials should be taken into account, in order to increase the realism of the sound in a web 3D environment. For that reason, the next step, to enrich the X3D sound nodes, is to introduce new acoustical material descriptors. This approach is one of the first to provide sound material characteristics in the web 3D scene.

The following figure demonstrates the most important sound propagation phenomena, in order that the overview of acoustical material descriptors will be understandable.



- **absorption** coefficient
- **specular** coefficient
- **diffusion** coefficient
- **refraction** coefficient
- **diffraction** coefficient

Figure 10: Sound Propagation Phenomena[2]

2.1 Absorption coefficient

The sound **Absorption Coefficient (a)** of a surface is the ratio of the sound intensity absorbed or otherwise not reflected by a specific surface that of the initial sound intensity. This characteristic depends on nature and thickness of material. Particularly, the sound is absorbed when it encounters fibrous or porous materials, panels that have some flexibility, volumes of air which resonate, openings in the room boundaries (e.g. doorway). Moreover, absorption of sound by a particular material/panel depends on the frequency and angle of incidence of the sound wave.
Some of the typical sound absorbers could be the acoustic panels, membranes, drapes, foams, carpets and can be represent by a variable taking values between zero (zero absorption - total reflection) and one (perfect absorption - no reflection). Absorption coefficients are usually measured in reverberation chambers according to DIN EN ISO standard 354.

2.2 Specular coefficient

The sound **Specular Coefficient (s)** describes specular reflection, one of the physical phenomenon of sound, which occurs when a sound wave strikes a plane surface and a part of the sound energy is reflected back into the space but the angle of reflection is equal to the angle of incidence.

[2] B. Kapralos, M. Jenkin, and E. Milios, “Sonel Mapping: A Probabilistic Acoustical Modeling Method,” Build. Acoust., vol. 15, no. 4, pp. 289–313, Dec. 2008

2.3 Diffuse coefficient

The sound **Diffusion Coefficient (d)** aims to measure the degree of scattering produced on reflection. Specifically, it is produced in the same way as the specular reflection, but in this case, the sound wavelength is comparable with the corrugation dimensions of an irregular reflection surface and the incident sound wave will be scattered in all directions. In other words, it is a measure of the surface's ability to uniformly scatter in all direction. The diffusion coefficient is measured in an anechoic chamber according to the ISO standard 17497-2.

2.4 Refraction coefficient

The sound **Refraction Coefficient (r)** of a medium determines the propagation speed of the wave. This, for a wave travelling from medium one into medium two, then the ratio of the refractive indices is equal to the inverse of the velocity ratios. Furthermore, the sound waves are refracted when parts of a wave front travel at different speeds. This happens in uneven winds or temperatures. Sound waves tend to bend away from warm ground, since it travels faster in warmer air.

2.5 Proposal to Improve X3D Sound Component – Introduction of Acoustical Material Descriptors

Our proposal is not just to customize an environment but instead to fully integrate Web Audio API into X3D language, by introducing new sound nodes and incorporate them in X3D. Specifically, the first approach to improve X3D sound components provides the introduction of the new acoustical material descriptors to **Material** node of X3D. Figure 11 presents the current X3D node of **Material** and the first proposed approach:

```
Material : X3DMaterialNode {
  SFFloat [in,out] ambientIntensity 0.2        [0,1]
  SFColor [in,out] diffuseColor     0.8 0.8 0.8 [0,1]
  SFColor [in,out] emissiveColor    0 0 0       [0,1]
  SFNode  [in,out] metadata         NULL        [X3DMetadataObject]
  SFFloat [in,out] shininess        0.2         [0,1]
  SFColor [in,out] specularColor    0 0 0       [0,1]
  SFFloat [in,out] transparency     0           [0,1]
}
```

```
Material : X3DMaterialNode {
  SFFloat [in,out] ambientIntensity 0.2        [0,1]
  SFColor [in,out] diffuseColor     0.8 0.8 0.8 [0,1]
  SFColor [in,out] emissiveColor    0 0 0       [0,1]
  SFNode  [in,out] metadata         NULL        [X3DMetadataObject]
  SFFloat [in,out] shininess        0.2         [0,1]
  SFColor [in,out] specularColor    0 0 0       [0,1]
  SFFloat [in,out] transparency     0           [0,1]
  SFFloat [in,out] absorption       0    [0,1]
  SFFloat [in,out] specular         0    [0,1]
  SFFloat [in,out] diffuse   0    [0,1]
  SFFloat [in,out] refraction    0    [0,1]
}
```

Figure 11: Strategies to Improve X3D Sound Component – 1st Structure

Finally, Figure 12 illustrates the second proposed approach, in which the new acoustical material descriptors is introduced to new **AcousticMaterial** node within **Shape** node of X3D.

```
Shape : X3DShapeNode {
  SFNode  [in,out] appearance NULL     [X3DAppearanceNode]
  SFNode  [in,out] geometry   NULL     [X3DGeometryNode]
  SFNode  [in,out] metadata   NULL     [X3DMetadataObject]
  SFVec3f []       bboxCenter 0 0 0    (-∞,∞)
  SFVec3f []       bboxSize   -1 -1 -1 [0,∞) or −1 −1 −1
}
```

```
Shape : X3DShapeNode {
  SFNode  [in,out] appearance NULL     [X3DAppearanceNode]
  SFNode  [in,out] geometry   NULL     [X3DGeometryNode]
  SFNode  [in,out] metadata   NULL     [X3DMetadataObject]
  SFVec3f []       bboxCenter 0 0 0    (-∞,∞)
  SFVec3f []       bboxSize   -1 -1 -1 [0,∞) or −1 −1 −1
  SFNode [in, out] AcousticMaterial NULL [X3DAcousticMaterial]
}
AcousticMaterial : X3DAcousticMaterial {
  SFNode  [in,out] metadata       NULL     [X3DMetadataObject]
  SFFloat [in,out] absorption     0    [0,1]
  SFFloat [in,out] specular       0    [0,1]
  SFFloat [in,out] diffuse   0    [0,1]
  SFFloat [in,out] refraction    0    [0,1]
  SFString [in,out] name //Name (and only the name) of the material
  SFString [in,out] description //Detailed description of the material
}
```

Figure 12: Strategies to Improve X3D Sound Component – 2nd Structure