HAL open science

Sound Objects for SVG

Audrey Colbrant, Yohan Lasorsa, Jacques Lemordant, David Liodenot, Mathieu Razafimahazo

To cite this version:

Audrey Colbrant, Yohan Lasorsa, Jacques Lemordant, David Liodenot, Mathieu Razafimahazo. Sound Objects for SVG. SVG Open 2010, Telecom Paris, Aug 2010, Paris, France. inria-00540356

HAL Id: inria-00540356

https://inria.hal.science/inria-00540356v1

Submitted on 26 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sound Objects for SVG

Audrey Colbrant [WAM]
Yohan Lasorsa [WAM]
Jacques Lemordant [WAM]
David Liodenot [WAM]
Mathieu Razafimahazo [WAM]

Abstract

A sound object can be defined as a time structure of audio chunks whose duration is on the time scale of 100 ms to several seconds. Sound objects have heterogeneous and time-varying properties. They are the basic elements of any format for Interactive Audio (IA). We have designed an XML language [A2ML] for Interactive Audio which offers, concerning the sequencing of sounds, a level of capabilities similar to that of iXMF, the interactive audio file format defined by the Interactive Audio Special Interest Group [IASIG]. A2ML uses SMIL [SMIL] timing attributes to control the synchronization of sound objects and supports 3D sound rendering, DSP and positional parameters's animation, by embedding the SMIL animation module. Like in a traditional mixing console, mix groups can be used to regroup multiple sound objects and apply mix parameters to all of them at the same time. An API allows external control and dynamic instantiation of sound objects.

As with graphics, a declarative language for interactive audio is much more powerful than a node-graph based approach implemented using an imperative language. The structured declarative model offers easier reuse, transformability, accessibility, interoperability and authoring. An XML declarative language for audio like A2ML could help to reach the goal of the IXMF workgroup, i.e. build a system by which composers and sound designers create an interactive soundtrack and audition it by simulating target application control input while working in the authoring environment.

In this paper, we will show how an XML language for interactive audio can be used with SVG. After an introduction to the history of sound objects, we will use the example of a computational character with an simple orient behaviour to demonstrate the complementarity of SVG and A2ML. The best way to use these two languages is to synchronize them with a third one, a tag-value dispatching language. We will then present a complex application for which the use of both SVG and A2ML is natural, i.e. a navigation system for visually impaired people based on OpenStreetMap.

Table of Contents

Time-Structured Objects 2
- Sound Objects 2
- Video Objects 3
- Earcons 3
- Soundscapes 4
- Format for Sound Objects 4
- Format for Mix Groups 5

SVG-A2ML Objects 6
- A computational character 6

SVG-A2ML Navigation 9
- OpenStreetMap 10
- Map-Aided Positioning 10
- Radar-Based Rendering 10
- Soundscape 11

Acknowledgements 13
Bibliography 13

Time-Structured Objects

The main attempt to describe in a declarative format time structured objects was the SMIL language [SMIL]. The composition model in SMIL is hierarchical, i.e. nodes cannot become active unless all of their ancestors are active. This kind of composition is adaptive by nature to varying bandwidth conditions and selection of optional content. The price to pay for this adaptability, is a lack of real-time interactivity at the level of the composition, a primary feature which is needed in interactive audio applications such that games or indoor-outdoor navigation applications.

One solution is to limit the hierarchical composition model to only two levels of time containers, sequential containers with optional exclusive containers inside. As we shall see, this simple composition model was in fact used earlier by audio and music compositors, even if expressed on a less formal basis.

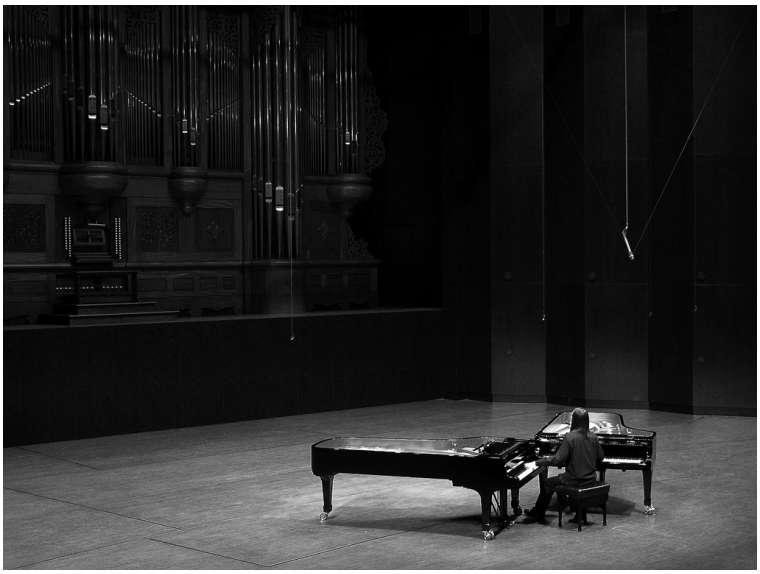
Time structuration is of interest not only for audio but also for other kinds of media like video and graphics. A similar composition model, known under the name MTV'model [MTV] can be applied to video and a graphic composition model could be used for audio visualization.

Sound Objects

The implication of repetition, looping, indetermination in audio have been explored by many groups. One well-known group is "Group de Recherche Musicale" (GRM) of the French school [GRM]. The terms "objets sonores" and "musique concrète" were first coined in 1951 by Pierre Schaeffer [GRM], repetition and looping being studied together with sound abstraction. At the same time, there was a lot of work around indeterminate music by the composers of the New York School of the 1950, John Cage and Morton Feldman, for example.

If the French school, with Xenakis among other composers, was then interested by the use of mathematical models, stochastic processes and algorithmic transformations in music composition, the American school pioneered the style of minimalist music with La Monte Young for drone music, Philip Glass for repetitive structures and Steve Reich for loops with phasing patterns. We have written below Steve Reich's Piano Phase [REICH] in our A2ML language for interactive audio. Cues is the name for sound objects used by audio games and electronic music composers.

Figure 1.



Piano Phase by Peter Aidu

```
<!--A2ML document-->
<a2ml xmlns="http://wam.inrialpes.fr/a2ml">
   <cues>
      <cue id="piano1" begin="0" loopcount="-1">
         <chunk>
            <sound src="piano_loop.flac"/>
         </chunk>
      </cue>

      <cue id="piano2" begin="0" loopcount="-1">
         <chunk>
           <sound src="piano_loop.flac">
              <!--Phasing:playback speed to 95%-->
              <rate value="95000"/>
           </sound>
         </chunk>
      </cue>
    </cues>
</a2ml>
```

Interesting work is still going on with New York-based composer Kenneth Kirschner, his interest being raised in indeterminate music by the shuffle mode of the iPod and by the possibility of using flash as a real-time composition system, with a mix of piano pieces, field recordings and electronic music. We can declare this kind of music in A2ML and run it on an iPod with our A2ML sound engine.

Sound objects have been logically adopted by audio game composers and designers to create interactive soundtracks. Audio for games is the main domain where they are used together with proprietary software to help composition. An unanswered question is whether this kind of audio is received differently in games with a visual context than in music systems. [GRM]

Video Objects

Generating video from video clips is now extensively used on some TV channels, MTV for example. The MTV style sidesteps traditional narrative. The shaping device is the music and narrative is less important than a feeling state.This makes the jump cut more important than the match cut. Premonitions of a ‘You Tube Narrative Model’ [YOUTUBE] can be considered in relation to Dancyger’s MTV Model: the feature film as an assemblage of ‘set-pieces’ which appropriate both the structure (2-4 minutes) and aesthetic (high production values/rapid montage) of the music video. The concept of video objects can be easily grasped by reusing the ideas about the structuration and synchronization of sound objects and by changing the media: going from audio to video. It could be interesting to design a format similar to A2ML for time structured video objects with declarative transitions and effects. This language could be applied to MTV and YOU tube models.

Earcons

In the field of sonic interaction, sound objects are called earcons. Earcons are used to give information to the user, in a navigation system for example. This use of audio is completely opposite to the kind of perception, the ‘acousmatic listening’ coming from the Schaeffer's concept of sound objects. This is a rather interesting fact and a consequence of advances in audio technology. This kind of audio-only way of conveying information about the world is called Auditory Display. The auditory display can be spread around the user in 3D and the information given as recorded or virtual speech, non-speech sounds, earcons, auditory icons, or a combination of all of these. Earcons are structured sequences of sounds that can be used in different combinations to create complex audio messages, whereas auditory icons are everyday sounds used to convey information to the user. A2ML sound objects can be considered as a powerful generalization of earcons.

Soundscapes

A soundscape is more than an auditory display. It can be defined as set of sound objects, interactions between these objects, and interactions with space when there is an architectural envelope. Interactions between sound objects are an essential feature of soundscapes. Internal and external synchronization through SMIL events in A2ML allows to build complex soundscapes with interactions. We have build such a soundscape called ‘Copenhagen Channels Underwater Soundscape’, with a preliminary version of A2ML. [ICMC07]

Format for Sound Objects

Initially, a sound object as defined by Pierre Schaeffer is a generalization of the concept of a musical note, i.e., any sound from any source which in duration is on the time scale of 100 ms to several seconds. In A2ML, this concept was extended and raised to its full power with a time structuring of sounds with randomization, attributes for internal and external synchronisation and DSP parameters animation. This declarative approach to sound objects allows for:

- Better organization (sound classification)
- Easy non-linear audio cues creation / randomization
- Better memory usage by the use of small audio chunks (common parts of audio phrases can be shared)
- Separate mixing of cues to deal with priority constraints easily
- Reusability

As the accent is on the interactivity, we don't want a full hierarchical time composition model. We want to allow:

- one-shot branching sounds (selection among multiple alternate versions)
- continuous branching sounds (selection against multiple alternate next segments)
- parametric controls mapped to audio controls parameters like gain, pan/position, pitch, ...)

A2ML sound objects allow in fact more than that. In the SMIL terminology, sound objects are sequential containers for chunks and optionaly these chunks can be exclusive containers for sounds. Randomization is provide by attributes at the level of these containers providing indeterminacy, a required feature.

Sound objects have synchronization attributes like SMIL containers, chunks have attributes to specify audio transitions and sounds have attributes to control audio parameters like volume, pan, mute. These attributes can be animated like in SVG through the embedded SMIL animnation module. To allow for dynamically ordering or selecting which media chunks get played, sometimes influenced by the application state, sometimes to reduce repetition, sound objects contain only references to chunks. It's an essential step towards audio granulation which represents the future of audio for games.

We have try in a few words to explain the main concepts behind the time structuration of A2ML's sound objects. We have, like in SVG, support for instantiation, animation, transition, synchronization and styling with selectors. Styling in SVG correspond to submixing in audio and will described in the next paragraph. Consequently, a language like A2ML for interactive audio, is easily mastered by people familiar with SVG. A RELAX NG schema for A2ML can be found at [A2ML] and more explanations in [AES127].

The following example is an A2ML fragment used to describe the sonification of a building. This kind of documents is used in our indoor navigation system and played on iPhones. Recall that, in [IASIG], sound objects are called cues and we have followed this terminology in A2ML. This A2ML fragment contains cues models to be instantiated by events.

```
<!--A2ML document-->
<!-- background music -->
<cue id="ambiance"  loopCount="-1"  begin="ambiance">
   <chunk pick="random" fadeOutType="crossfade" fadeOutDur="0.5s">
      <sound src="piano1.wav"/>
      <sound src="piano2.wav"/>
      <sound src="ocean.wav"/>
   </chunk>
   <chunk pick="exclusiveRandom" fadeOutType="crossfade" fadeOutDur="0.5s">
      <sound src="electronic1.wav"/>
      <sound src="electronic2.wav"/>
      <trigger chance="10"/>
   </chunk>
</cue>

<!-- Environmental cue -->
<cue id="floor_surface" loopCount="1" begin="floor.change">
   <chunk pick="fixed">
      <sound src="carpet.wav" setActive="floor.carpet"/>
      <sound src="marble.wav" setActive="floor.marble"/>
   </chunk>
</cue>

<!-- Environmental cue -->
<cue id="hallway" loopCount="1" begin="hallway">
    <chunk>
        <sound src="hallway.wav"/>
    </chunk>
</cue>
```

Format for Mix Groups

Mixing consists in combining multiple sources with effects into one output. Submix is a common practise which corresponds to the creation of smaller groups before generating one output. In electro-acoustic music, these groups are called sections, the rhythm section, the horn section, the string section, ... In A2ML, mix groups can be used to regroup multiple cues and apply mix parameters on all of them at the same time. In addition to mixing multiple cues, they can also be used to add DSP effects and locate the audio in a virtual 3D environment. The main difference with traditional mix groups is that a cue can be a member of multiple sections, and the effects of all of them will apply, making sections very versatile. The sound manager's response to a given event may be simple, such as playing or halting a sound object, or it may be complex, such as dynamically manipulating various DSP parameters over time. The sound manager offers a lower level API through which all instance parameters can be manipulated such as positions of the sound objects and the auditor.

The following example is A2ML fragment used to describe the rendering of sound objects used in the sonification of a building. The reverb effect is of studio production type and not resulting from physical space simulation. SMIL animation of DSP parameters is used in the animate element.

```
<!--A2ML document-->
<sections>
  <!-- Mix group for guidance sound objects.
      Reverb will be used to inform of room size changes. -->
 <section id="audioguide" cues="waypoint door stairway elevator">
```

```
    <dsp name="reverb">
      <parameter name="preset" value="default"/>
      <animate id="pre_change" attribute="preset" values="ch_reverb"/>
    </dsp>
    <volume level="70"/>
 </section>

  <!--  3D Rendering activation  -->
  <section id="objects3D" cues="wp door stairway elevator floor">
     <mix3D>
        <distance attenuation="2 />
     </mix3D>
   </section>

  <!-- Submix group for the environment -->
   <section id="details" cues="atrium_door_number">
     <mix3D>
         <distance attenuation="5 />
     </mix3D>
     <volume level="100"/>
   </section>
```

SVG-A2ML Objects

SVG Animation and A2ML Animation are structured declarative language that have an execution model. By using SVG with A2ML, we are able to build objects with audio and graphic behaviour in a declarative way. This could be considered one day as a basic media functionality. In this section, we will show how to combine these two temporal languages by showing a way to associate these execution models in both direction, the graphics driving the audio or the audio driving the graphic. A2ML having the capability to raise external events, the situation is symmetric and it will be enough to explain how we proceed when we want graphics to drive the audio.

We have to find a way to transfer events in the SVG world into A2ML events. We could have looked at a compound document (CDF) solution but if we want to keep the interaction between the graphic designer, the audio designer and the application programmer at its minimum, it's not the way to go, i.e. we don’t want sound objects to be positioned in the SVG world by inclusion or by reference.

Requests for instantiation of sound objects are done through TVDL, a Tag-Value Dispatching Language, reminiscent from NVDL (Namespace-based Validation Dispatching Language) [NVDL] and used by OpenstreetMap SVG renderers, the used of pairs tag-value being a basic feature of OpenStreetMap. As TVDL allows to build groups or layers of sound objects, selective rendering of sound layers is possible depending on the user's preferences. This is a primary requirement in mobile applications, navigation application for example. TVDL-A2ML can be thought as an audio style sheet for SVG. However, it's more than that because the sound objects can react to the context, and this goes further than a simple audio file activation.

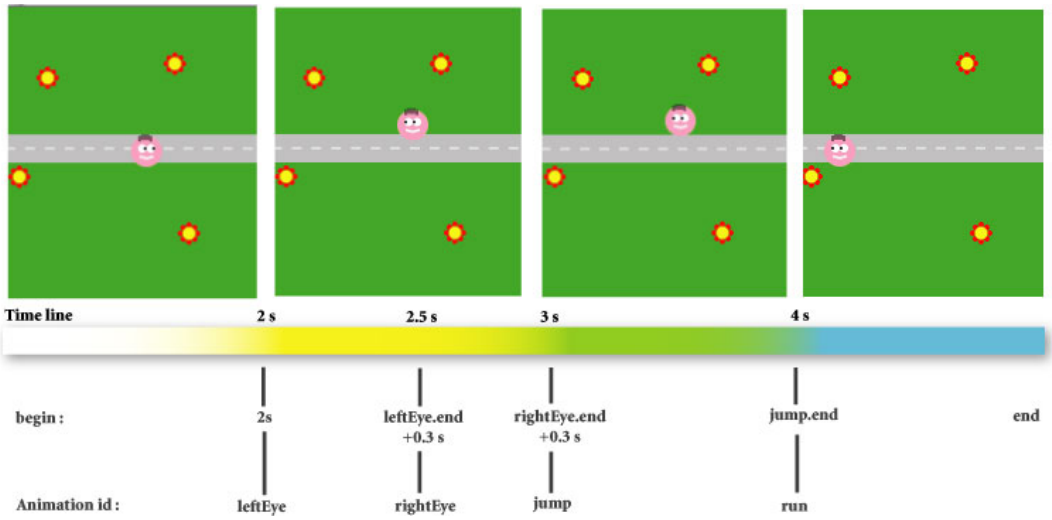
TVDL is used to send requests to a sound manager with A2ML reading capability. The sound manager’s response to a given cue instantiation may be simple, such as playing or halting a 3D sound source, or it may be complex, such as dynamically manipulating various DSP parameters over time. The sound manager is also offering an API through which all instance parameters can be manipulated such as positions of sound sources and auditor.

A computational character

We consider a simple computational character: a point on a line. The body of the character is a point. We will refer to the character, i.e. the combination of its behavior based on SVG-A2ML animation and its body as Point. We will discuss the development of a simple audio-graphic orient behavior for

Point. Point will turn its left eye, then right eye, jump four times and then move to the left on his line as shown in figure 1.

Figure 2.



SVG Animation

We give now fragments of SVG document that represent Point with its orient behaviour:

```
<!-- SVG document-->
<svg version="1.1" width="180" height="208"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">

 <script type="text/ecmascript">
   function A2MLAnim(evt) {
     var id_attribute=evt.target.getAttribute('id');
    ...
    }
 </script>
 <defs>
  <g id="PointProto" >
    <circle id="head" cx="0" cy="0" r="11" fill="rgb(255,160,192)"/>
    <g id="eyes">
      <circle id="whiteEyeLeft" cx="-3" cy="-2" r="3" fill="white"/>
      <circle id="whiteEyeRight" cx="3" cy="-2" r="3" fill="white"/>

       <circle id="leftEye" cx="-1" cy="-2" r="1" fill="black">
            <animate id="LeftEye"
                begin="2s"  onbegin="A2MLAnim(evt)"
                attributeName="cx" from="-1" to="-5"
                dur="0.2s" fill="freeze"/>
       </circle>

       <circle id="rightEye" cx="5" cy="-2" r="1" fill="black">
             <animate id="rightEye"
                 begin="leftEye.end+0.3s" onbegin="A2MLAnim(evt)"
                 attributeName="cx" from="5"  to="1"
                 dur="0.2s"  fill="freeze"/>
       </circle>
      </g>
```

```
        ...
    </g>
 </defs>
 ...
 <use xlink:href="#PointProto" x="100" y="102" id="Point">
    <animate id="jump"
       begin="rightEye.end+0.3s" onbegin="A2MLAnim(evt)"
       attributeName="y"
       values="102;80;102;80;102;80;102;80;102"
       dur="1s" fill="freeze"/>
    <animate id="run"
       begin="jump.end" onbegin="A2MLAnim(evt)"
       attributeName="x" from="100" to="-15"
       dur="1s" fill="freeze"/>
 </use>
 ...
</svg>
```

The four animations follow each other as specified by the SMIL attributes in the SVG document. At its start, each animation triggers the activation of the A2MLAnim javascript function associated with the onbegin attribute. This function retrieves contextual data such as the name of the element, the id of the animation, the name of the parent element and forwards it to the A2ML event manager which triggers the A2ML event specified in the TVDL document shown below.

```
<!-- TVDL document-->
<rules name="standard">
    <rule e="circle" k="animate" id="leftEye|rightEye">
        <cue name="eye.move">
    </rule>

    <rule e="g" k="animate" id="*">
        <layer name="point">
            <rule k="animate" id="jump">
                <cue name="point.jump">
            </rule>
            <rule k="animate" id="run">
                <cue name="point.run">
            </rule>
        </layer>
    </rule>
</rules>
```

This document describe the association rules of SVG and A2ML objects. Thanks to the id retrieved, a graphic animation is synchronized with an audio animation. This document allows us to modify these associations easily without having to take into account the description of graphic and audio animations. It is possible to do audio rendering of more Point's actions by adding another TVDL document or by grouping them inside a layer element. The value associated with the name attribute of the cue element in the TVDL document is the name of the event which will be sent to the Sound Manager, called IXE for Interactive eXtensible Engine. The IXE will the start the cue designated by the event. The A2ML document is shown below:

```
<!-- A2ML document-->
<a2ml xmlns="http://wam.inrialpes.fr/a2ml">
<cues>
```

```
  <cue id="eye" loopCount="1" begin="eye.move">
      <chunk>
         <sound src="eye_move.flac"/>
      </chunk>
      <pan>
         <animate id="pointEye" attribute="pan"
            begin="eye.move" dur="0.2s" from="100" to="-100"/>
      </pan>
  </cue>

  <cue id="jump" loopCount="4" begin="point.jump">
      <chunk>
         <sound src="point_jump.flac"/>
      </chunk>
      <volume>
         <animate id="pointJump" attribute="level"
           begin="point.jump" dur="0.25s" from="100" to="0"/>
      </volume>
  </cue>

  <cue id="run" loopCount="1" begin="point.run">
      <chunk>
         <sound src="point_run.flac"/>
      </chunk>
      <pan>
         <animate id="pointRun" attribute="pan"
           begin="point.run" dur="1s" from="100" to="-100"/>
      </pan>
  </cue>
</cues>
</a2ml>
```

Each cue has its own animation set to an audio parameter such as volume or pan control. The animation duration can be declared explicitly or be the same as the SVG animation. The described solution presents a synchronization mechanism with SVG animations that triggers A2ML animations. Symmetrically it is also possible to launch an SVG animation from an A2ML animation. In this way, we obtain a synchronization mechanism of SVG and A2ML animations.

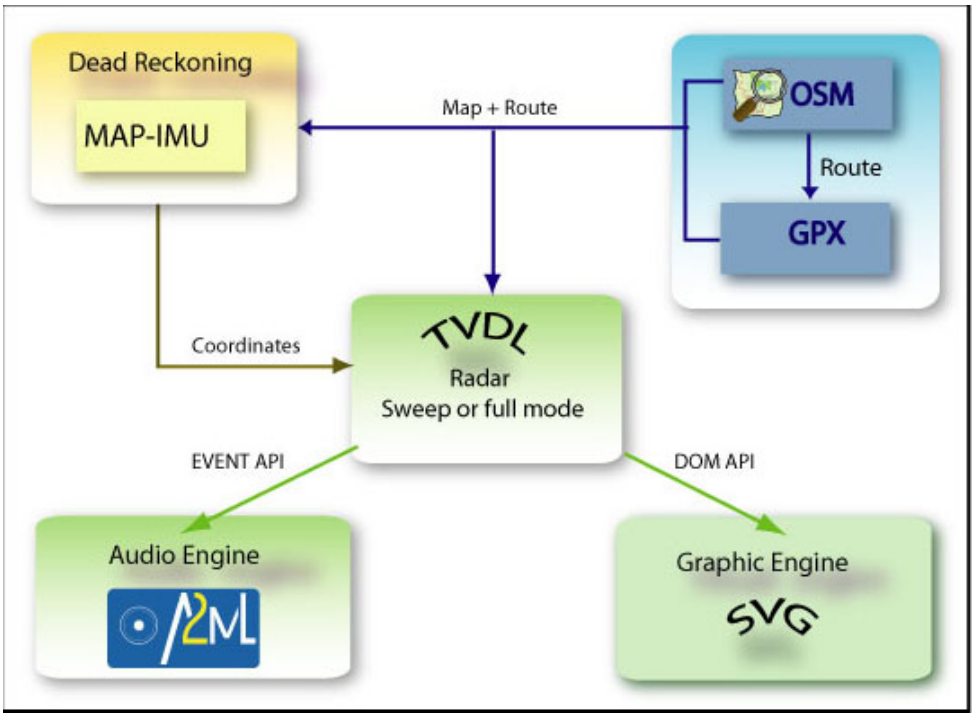
SVG-A2ML Navigation

We will use a guidance application for visually impaired people, we are developing at INRIA (Grenoble, France) under the Autonomy project (supported by FEDER), to illustrate the way SVG and A2ML can be used to develop interactive Augumented Reality Audio (ARA) applications. An indoor-outdoor semi-automatic navigation system for visually impaired people can be built using:

- A2ML to build a local soundscape using a radar metaphor.
- SVG to build a distant operator console (semi-automatic navigation with the help of an operator in a call-center).
- SVG to simulate navigation while auditioning the soundscape during its authoring.

Once again, the glue between these two similar languages is done through rules expressed in a tag-value dispatching language TCDL. These rules are activated using a radar metaphor (see figure 4). The architecture of this navigation system is shown on figure 3.

Figure 3.



Navigation System Architecture

OpenStreetMap

We choose OSM as a format for geographic data because with its very simple schema and tag-value extension mechanism, it can be easily used to build a node-graph representation of buildings with non-geographic features represented by the pairs Tag-Value. Having the same format for both indoor and outdoor data, brings a great simplicity in designing a global (indoor+outdoor) navigation application. The tag-value approach used by OSM to describe geographical information is also particularly interesting with an event-based audio system like A2ML.

GPS eXchange Format (GPX) is a light-weight XML data format for the interchange of GPS data (POI, routes, and tracks) between applications and web services on the internet. We use it for the output of our indoor routing algorithm. Its namespace-based extension element allows us to enhanced route data with contextual tag-value data, useful for both map-aided positioning and audio rendering.

Map-Aided Positioning

An Inertial Measurement Unit (IMU) (accelerometers, magnetometers, gyroscopes) is now embedded in smartphones. From it, a positioning system called dead-reckoning can be built. Our MAP-IMU module estimates the user position with the IMU and corrects it thanks to smart Map-Aided algorithms. In outdoor, the MAP-IMU module can be assisted by a satellite based differential GPS. With the MAP-IMU module, position can be determined without any special infrastructure, except perhaps for big halls like we can found in train stations or for building with a special topology like rooms connected without corridors.

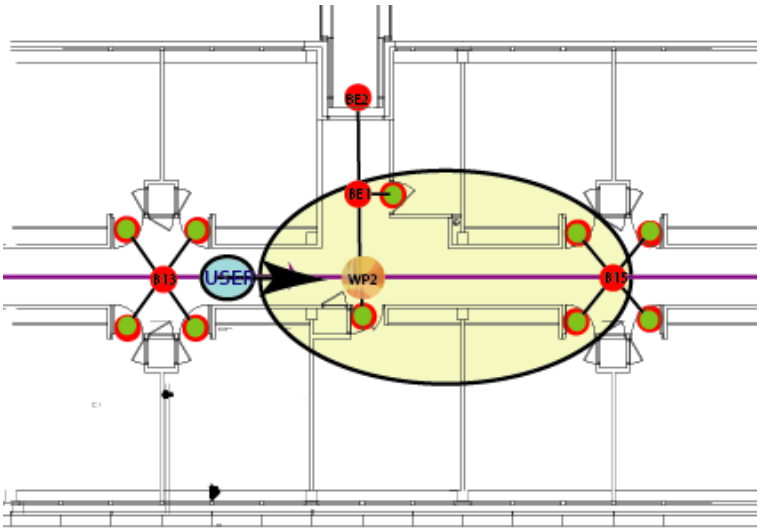
Radar-Based Rendering

The radar device is an interaction technique, which allows a precise scanning of the scene and high quality information feedback. The radar can be thought of as a flashlight, which is used to illuminate the scene. Everything that gets lit by the radar answers in a predefined manner, by either a verbal

description or an agreed sound signal. This allows us to do a local rendering of the soundscape describing the environment. The radar can be employed in two modes:

- sonar mode with automatic spatial sweep.
- full mode which consists in playing the sound of detected objects concurrently.

Figure 4.



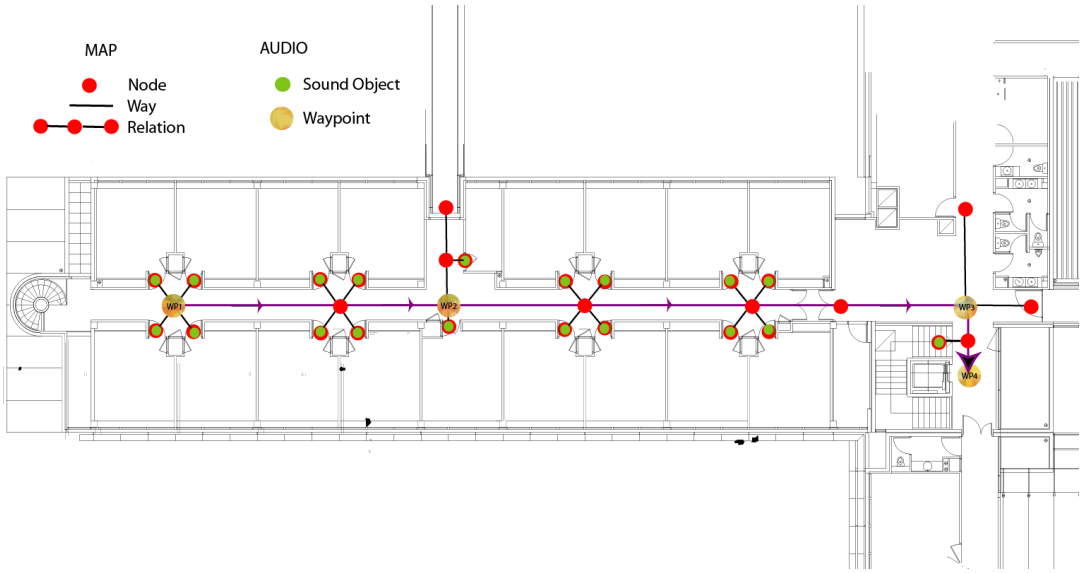
SVG view used for audition of our Radar-based Rendering

Soundscape

We construct a soundscape that provides the user with a direct perceptual information about the spatial layout of the environment including the waypoints. We have three kinds of sound objects:

- Navigational beacons where the user walks directly toward the sound. A rapid beeping sound is spatialized so that the beeps appear to come from the direction of the next way point. A vocal announcement of the remaining distance to the next way point is done at some low frequency.
- Ambiant cues with environmental reverb providing clues as to the size and character of the enclosing space (room, hall,corridor, ...)
- Secondary sound objects indicating nearby items of possible interests such as doors, stairs and escalators.

Figure 5.



OSM for the Montbonnot INRIA Building Representation

For a navigation system, sonic interaction design matters. It takes a lot of time researching which sounds are more effective, such as a beep or a sound burst, impact or impulsive contact (preponderance of transients) sounds, hitting or breaking, bouncing or continuous contact (significant steady state part), rolling or sliding, all that with different kinds of material (wood, metal, glass, plastic). Very fast testing is therefore necessary together with mobile mixing, and this is supported by our audio rendering system which uses both a cue-based XML document and a Tag-Value Dispatching document. An example of a TVDL document for the INRIA building is shown below.

```
<!--TVDL document-->
 <rules name="INRIA Building">

  <rule e="node" k="anemity" v="*">
    <layer name="anemity1">
       <rule  k="anemity" v="toilets">
         <cue name="toilets" />
       </rule>
       <rule  k="anemity" v="bar">
         <cue name="bar"/>
       </rule>
    </layer>
  </rule>

   <rule e="relation" k="type" v="junction">
    <rule e="member" role="door">
      <cue name="door"/>
    </rule>
   </rule>

  <rule e="node|way" k="tactile_paving" v="yes">
   <cue name ="guide_tactile_paving"/>
  </rule>

  <rule e="node" k="floor_access" v="elevators">
   <cue name ="elevators"/>
```

```
    </rule>

  </rules>
```

Acknowledgements

This work was done by the WAM Project Team of INRIA (http://wam.inrialpes.fr) under the auspice of the Autonomy Project (Global Competitive Cluster Minalogic) and financed with the help of the European Fund of Regional Development (EFRD).

Bibliography

[WAM] *WAM Project-Team, INRIA [http://wam.inrialpes.fr].*

[GRM] *Sound Objects, based on Pierre Schaeffer's work [http://en.wikipedia.org/wiki/Pierre_Schaeffer].*

[REICH] *Piano Phase [http://fr.wikipedia.org/wiki/Steve_Reich].*

[MTV] Ken Dancyger. Focal Press. *Technique of Film and Video Editing, Third Edition: Theory and Practice* .

[YOUTUBE] *You Tube Narrative Model [http://flowtv.org/2007/06/feature-film-a-you-tube-narrative-model/].*

[IASIG] *Interactive XMF, Interactive Audio Special Interest Group [http://www.iasig.org./wg/ixwg].*

[SMIL] *Synchronized Multimedia Integration Language (SMIL 2.1), W3C [http://www.w3.org/TR/SMIL2/].*

[NVDL] *Namespace-based Validation Dispatching Language [http://www.nvdl.org].*

[OPENSL] *OpenSL ES [http://www.khronos.org/opensles].*

[ICMC07] Jacques Lemordant and Agnes Guerraz. International Computer Music Conference. *Mobile Immersive Music*.

[AES127] Yohan Lasorsa and Jacques Lemordant. 127th AES Convention. *An Interactive Audio System for Mobile*.

[AES128] Jacques Lemordant and Yohan Lasorsa. 128th AES Convention. *Augmented Reality Audio Editing*.

[A2ML] *A2ML specification [http://gforge.inria.fr/projects/iaudio].*