A system for soundscape generation, composition and streaming

Mattia Schirosa, Jordi Janer, Stefan Kersten, Gerard Roma
Universitat Pompeu Fabra
Music Technology Group
Roc Boronat 138, Barcelona
firstname.lastname@upf.edu

ABSTRACT

Soundscape design is beginning to receive considerable attention in virtual environments and interactive media developments. Current trends (e.g. online communities and games, web and mobile technologies and augmented-reality tourism platforms, 2D and 3D virtual cartography and urban design) might require new paradigms of soundscape design and interaction.

We propose a generative system that aims at simplifying the authoring process, but offering at the same time a realistic and interactive soundscape. A sample-based synthesis algorithm is driven by graph models. Sound samples can be retrieved from a user-contributed audio repository. The synthesis engine runs on a server that gets position update messages and the soundscape is delivered to the client application as a web stream. The system provides standard format for soundscape composition. Documentation and resources are available at [1] .

1. INTRODUCTION

Audio is a crucial element for building immersive virtual environments. In this context, a principal role of audio is the creation of a sound ambiance or soundscape, in fact, during the last decade, several technologies emerged to provide a more realistic and interactive audio content. We can include, among others, real-time synthesis techniques (e.g. physical models [2], sound ambience textural synthesis [3], [4]) or spatial audio reproduction systems (e.g. 5.1, Ambisonics).

Audio generation systems generally feature different characteristics depending on the type of media production (e.g. animation films, virtual worlds, arcade games, etc.). They can differ in terms of interaction possibilities, latency requirements, or the achieved sonic realism.

At the same time, we observe that current developments such as online communities, web and mobile technologies, might bring applications that make use of a new paradigm of soundscape generation. Precisely in this new context, we believe that the authoring process should be simplified, also encompassing user-generated content.

Our system is positioned in this new scenario. It takes advantage of user-contributed sound assets and it provides a technology that allows the system to be part of web-based tools for the design and interaction of the soundscape.

The system is formed by three components: composition format, generation module and streaming web service. Figure 1 shows the whole system overview.

The composition format is a series of rules and parameters that describes the sound space, the sound concepts that live within that space, their generation probabilities and several types of sequencing behaviour between events. This information is encoded in a KML score file: KML is an XML schema aimed at describing and displaying geographic data [5]. The composition is also aimed at producing the database: an XML document that stores the sound concept events annotation mapped to segmented sound samples. Composition format is addressed to designer, composers and users. Composition or design are human processes, it is a challenge to think to a complete automatic design.

The generation module parses the score and the database annotations, it generates the graph structures, performs the sequencing and the multi-listener spatialisation synthesis, provide an OSC (Open Sound Control) interface for real-time soundscape performance interaction and listener position control.

The actual generation runs on the streaming server, the listeners position and orientation are communicated through HTTP request messages, and the soundscape is accessed as a MP3 stream. The developed web API includes session management, allowing personalized streams for simultaneous listeners in the same soundscape.

2. BACKGROUND

When we speak about "Soundscape" we mean an interactive and explorable audio environment that completely differs from linear sound design works (e.g. cinema, animation, tv). We elaborate the soundscape definition presented in [6] and [7] : *"Soundscape" is a complex temporal-spatial structure of sound objects that composes the perception of an environment in a listener throughout its hearing, moving and discovering process.*

The soundscape is composed by a set of sound zones, which are populated by a set of sound concept classes which are described by a sequencing structure and realised in a set of sound events.

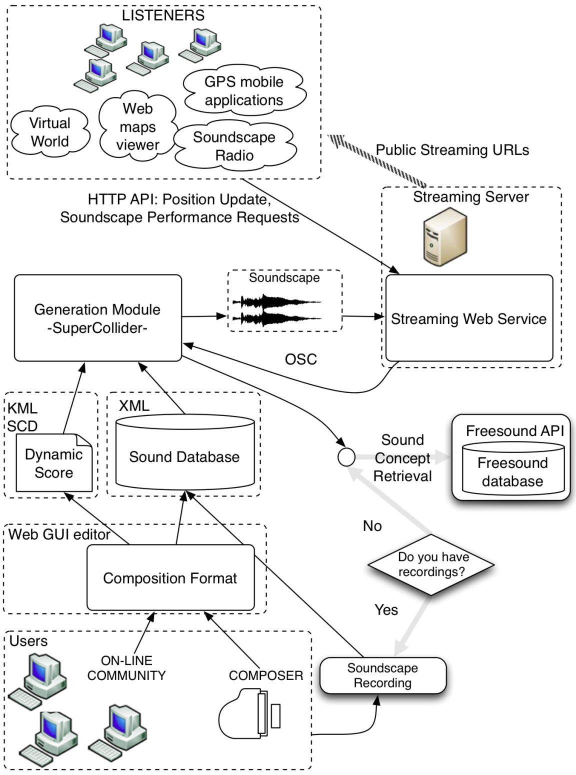

Figure 1. System overview

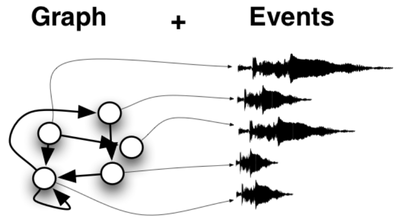

Figure 2. The sound concept

Sound zone is specific sub-part of the soundscape that presents characteristic sound ambiances and sound sources that allow to distinguish it from the other sub-parts. For instance, in one of our experiment, the sonification of the Second Life re-creation of the Canary Islands Las Palmas city, we identify 4 zones which are managed independently: the main city square, the city beach, the most important museum, and the biggest park just outside the museum. Zones have their own description scale that can differ from zone to zone. They have parameters that describe the area of space they refer to (geometry, closed ambience, scale etc.). Each Zone is composed by a set of sound concepts.

A **sound concept** describes the sonic behaviour of a "relevant" set of signals that are assumed to be perceived with the same meaning by the listener. Each concept is described by a set of sound events (simulated with sound samples), a sequencing structure stored in graph object (see figure 2) and parameters (see section 3) that affect the sequencing structure and the synthesis engine in real-time . The set of events collected in a concept class could represent several nuances of the concept. The user must decide where to put the threshold in concepts definition. For instance, describing the human sound activity of a soundscape, considering a coarse scale, we could choose just one concept: "people". Instead, using a higher detail, four concepts: "man", "woman", "children" and "elderly people" voices; finally, with a deeper detail, sixteen concepts: "laughts", "cheers", "screams", "quite voices", each one divided in male, female, young and elderly. The choice of sound concepts is fundamental for the system because it is the actual mapping performed by the user in order to control the temporal evolution of the soundscape through the interaction with the concept parameters. Certainly, this definition is related to the target soundscape and depends on the scene the user is interested to describe. This means that just one "people" concept is enough in a soundscape where humans are very rare, such as a forest. In this case, the user has no convenience in mapping all the specific voice sounds, instead the presence of human activity is a more interesting concept. On the contrary, in the soundscape of a pedestrian area of a european city, probably the user would be able to control the generation of each type of sound independently (smiles, cheers, screams, woman, children, etc.) as they could be related to specific soundscape status (e.g. during the morning children probability can increase, while during the night the concept could be almost deactivated).

Atmosphere is a particular type of concept, it is an overall layer of sound, which cannot be analytically decomposed into single sound objects, as no particular sound signal emerges. Atmosphere characterizes quiet states without relevant sound events. Each zone must have at least an active atmosphere because it is the background layer from which concept signals emerge.

Certainly the concept declaration is a subjective process, [6] it presents an interesting elaboration of the soundscape studies in order to have analysis procedures and listening exploration methodologies that assist the user in this task and in the definition of "relevant" signals.

Sound events are instances of a concept class. Each event links to a sound sample, thus it contains the information about sound sample URL, rate, duration, format, distance of recording, etc. Event recordings are completely focused on a specific source and try to avoid the presence of any background sound, thus it is preferred to select recordings performed with a highly directional microphone. Instead, atmospheric recordings represent a quite state of the soundscape with no relevant events, better performed with omnidirectional microphones. Events copy all the parameters from the parent class, but the user could specify their own specific parameters, for instance position.

The concept sequecing structure is represented by a **Graph** [8]: each vertex represent a sound sample. The edges represent a possible sequencing relation on a pair of vertices, the edge duration represents the pause between the triggering time of a vertex and the next one in the generative sequencing. Edge duration and vertex duration can be equal but do not necessarily have to coincide. If they are not equal the graph models triggering sequence were the next sound will be activated first or after the previous one stops. This behaviour allows to model pauses and cross-fades between vertices. Each edge has a specific probability to be chosen in the sequencing path. Actants are dynamic elements that navigate the graph and perform the triggering process according to the generative path based on probabilities. Each graph could have several actants.

3. COMPOSITION FORMAT

The composition format provides rules and parameters that assist the user in the soundscape composition through the sonic space description, the soundscape sequencing behaviour definition and the events database search. The user should sketch a metric map of the landscape, in order to have a reference system where to locate sounds. It is important to determine if the soundscape is composed by zones, this implies to name and define zone areas. Then the user chooses the sound concepts of each zone and, for each concept, he provides the annotation of its instances: audio samples that realise the events of the general concept class.

3.1 Sonic space

First, the user defines the sonic space, as shown in figure 3. The spatial parameters for zones are “geometry”, “scale” and “closed ambient”. Geometry describes the area where its features have effects, the synthesis engine supports only rectangular geometries. Scale is the ratio between the unit of the virtual space representation and meters. Each zone of the soundscape could have a different scale, this parameter affects the spatialisation engine. The user can assign a zone geometry to a closed ambient, in this case the spatialisation engine will mute all the other zones when the listener enters a closed ambient, while no closed ambient is audible when the listener is outside its geometry. Note that, if a zone has no spatial related feature, the geometry could also not be specified, in this case the zone is described by the set of its concepts positions, this allows to not rely on the limit of the rectangle geometry (for instance, the user can create a whole zone for all the streets of a city, in this case the zone is not a closed ambient but a container layer that allow to control all its concept parameters at the same time).

The second step focuses on concepts. Concepts have three position typologies: “point source”, “not point source” and “point source random generated in an area”. “Point” source is the most general type that represents a source position through a pair of coordinates. “Not point source” is defined by a rectangle, when the listener is inside it the concept amplitude does not change, while when the listener is outside it, the spatialisation engine computes the distance between him and the closest point of the rectangle; this position type is useful for specific sources or atmospheres that cannot be assumed to be located in a point (e.g. a non point source is a waterfall, whose sound is the compound of a multitude of parts). “Point source random generated in an area” is a useful position type to model

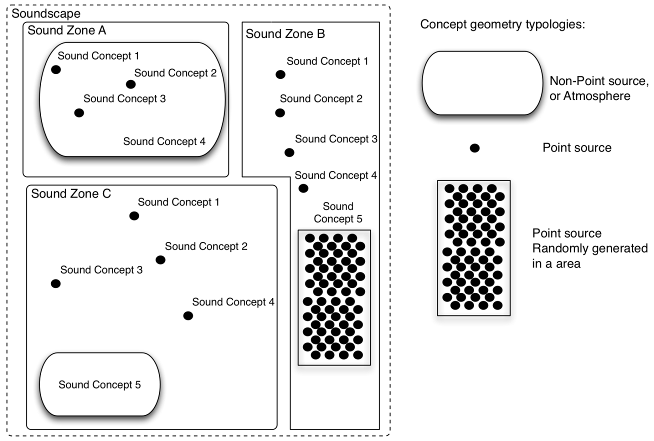

Figure 3. Sonic space description

common soundscape point sources that frequently appear in several positions moving along a specific area. For instance it is a powerful tool to model voices or waves. Concepts also has two position attributes: “clone” and “listened area”. If a concept is located in several fixed positions, clone is a more sensible choice than “point source random generated in an area” to model multiple concept occurrences in space. Clone allows to copy the parameters of a concept and to reuse its samples to place the same concept in another location. For instance it is useful to model several flags flapping under the action of wind. “Listened area” attribute de-emphasizes the perception of distance, it controls the distance from which the spatialisation engine applies a Low Pass Filter to the sound event in order to set a target concept more clearly audible also from far distances.

3.2 Sequencing and mix

Once the sonic space is defined, the user focuses on sequencing and mixing. The mix, basically, is the initialisation of the zones and each concept “gain” parameters. Those values must be expressed just if the user needs to amplify or de-amplify the elements, thus when the gain differs from one [1]. Concepts can be also “active” or “inactive”. The sequencing is described by a set of concept parameters. A “continuous” parameter produces a never ending stream of events. The sequencing of a continuous concept does not present pauses, but it can be deactivated, while when it is active it is always audible in the soundscape, if the listener is close to it. Examples of continuous concepts are wind, fountains or general water streams. Non-continuous concepts are described by “probability”, “multiple path” and “arrhythmic generation”. “Probability” describes the number of concept occurrences per hour, but the same probability could represent very different sequencing behaviour depending on the pauses between the generation of two consecutive events and the number of possible contemporaries triggering process. “Multiple path parameter” informs about the number of contemporaries triggering process (i.e. the number of graph actants, as described in section 2). “Arrhythmic generation” is a param-

[1] The software uses an linear amplitude representation where 0.125 = -18dB, 0.25 = -12dB, 0.5 = -6dB, 1 = 0db, 2 = +6dB.

eter that controls the irregularity of the triggering pattern. The graph generation algorithm is detailed in section 4 .

The spatial and sequencing information can be both written in the SuperCollider language (see section 4) or in KML files. The KML, namely Keyhole Markup Language, is a description open format based on XML used to display geographic data developed by Google and used on Google Earth and Google Maps. KML files can be created with the Google Earth user interface, or using an XML or a simple text editor to enter raw KML from scratch. As KML allows to declare extension data, we created a **soundscape KML** scheme that augments the basic KML informations adding all the compositional parameters previously explained.

Refer to the system documentation page [1] for a practical guide for editing the scheme instances, here we provide a general introduction.

We use the tag "Placemark", a tag with associated geometry, to declare zones and concepts. Placemark has name, description and two types of geometry elements, "Point" models point sound sources and "Polygon" models area sound sources and zone geometries. KML allows to create "Folder", a container element, we use a parent Folder to declare the soundscape, which in turn contains the zone Folders and each zone Folder contains a collection of Placemarks. We declare three extended elements with scheme identifiers "soundscape", "zone" and "concept" that allows to explicitly generate system specific data. Soundscape has four attributes: name, width, height and database type (the two supported database types are explained above). Zone has four attributes: gain, close ambient, scale and geometry, which allows to use a meter relative reference system instead of the KML standard latitude and longitude. Concept has nine attributes: geometry (again, meter relative reference system), gain, random generation, continuous, multiple path, probability, arrhythmic generation, listened area and clone. The possibility to use relative and absolute coordinate system allows the system to serve two types of application, respectively based on real locations or virtual spaces.

We chose the KML because it is becoming a popular format on online maps and earth browsers, thereby enabling interoperability of earth browser implementations. Programs such as AutoCad or 3D model editors easily export or convert their data in this format. We are developing a web GUI editor that allows users to place zones and concepts in a virtual space, to define parameters and to export a KML file.

3.3 Events database

Once the two previous step are accomplished, in a final step the user searches sounds that represent the concept events, creating the database. The system supports two types of databases: "pre-segmented events" or "annotated on-site recordings". "Pre-segmented event" format allows the user to specify a list of segmented audio samples that populate the concept instances. The XML minimal schema starts with the element "soundscapeDatabase", which contains the nestled "soundConcept" elements that in turn have the child elements "name" and a set of elements "event". The "event" element has six attributes that store the information about the database path (URL [2]), the segmentation (start and end frame), the sample rate of the audio file (sampleRate), and two types of normalisation attribute, the event distance of recording in case it is different from the standard of 5 meters (recDistance) or the normalisation volume. The system uses the latter attributes to normalise all the concept event with the same loudness before the spatialisation. Users that performed on-site recordings could prefer the distance of recording while users that search audio materials on online repository are more comfortable with a perceptive loudness normalisation.

A soundscape XML database annotation file example is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MTG soundscape composition system database>
<soundscapeDatabase>
  <soundConcept>
   <name>Wave<\name>
    <event URL="http://www.freesound.org/samples/id101429.aiff"
     start="0" end="12381696" sampleRate="44100"/>
<!-- New File-->
<event URL="http://www.freesound.org/samples/id101459.aiff"
 start="23456" end="234556" sampleRate="44100" recDistance="2"/>
<!--Same file, further event, further sound concept instance -->
<event URL="http://www.freesound.org/samples/id101459.aiff"
 start="234999" end="333000" sampleRate="44100"/>
  </soundConcept>
  <soundConcept>
   <name>Children<\name>
     <event URL="http://www.freesound.org/samples/id18929.aiff"
      start="0" end="12381696" sampleRate="44100"/>
<!-- Local Host path-->
<event URL="file:///Users/mattia/samples/smiling.aiff"
 start="23456" end="234556" sampleRate="44100" recDistance="10"/>
  </soundConcept>
   <!--...many concepts -->
</soundscapeDatabase>
```

Currently the XML annotation should be created using an XML editor. His step will be extended with a web GUI interface where users can easily select events, segment interesting audio file regions and export the annotation. The web interface shall make use of the Freesound [9] repository using an extended concept based search we previously developed [10].

The second format is "annotated on-site recordings". It is an annotation specifically for users that are interested in recreating the sonic environment of a real location. The annotation procedure is achieved using SonicVisualiser [3] . This software allows to create annotation of "Region" layer on audio file. Users have to manually select and label the segments in the recordings that belong to a concept with the associated name he previously defined. The annotation is exported as CSV file. The generative module creates an event for each concept segment annotation and it counts the number of occurrence per concept computing its probability.

4. GENERATION MODULE

The generative software is implemented in the audio programming language SuperCollider (SC) [11], which features a high-level, object-oriented, interactive language together with a real-time, efficient audio server. In the system documentation web page [1], we provide the gener-

[2] A sample path expressed as URL is in the form [subprotocol]://[node]/[sample path].

[3] An application for viewing and analysing audio files content http://www.sonicvisualiser.org/

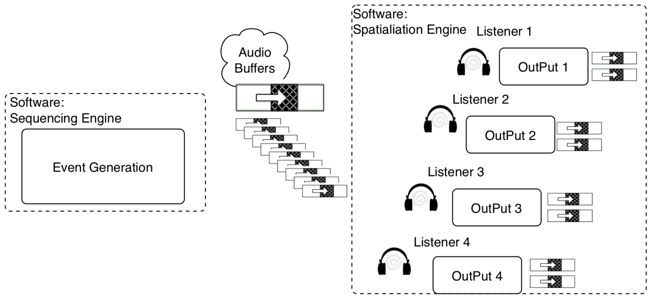

Figure 4. Sound events sequencing engine and spatialisation engine

ative software as a SC library, that uses two other external libraries in turn: "GeoGraphy" for the sound object sequencing management [8] and the "XML" SC library that implements a subset of the DOM-Level-1 specification for XML parsers.

As the application is designed for being a streaming web service module it presents high performance technical features: low requirements of CPU and RAM, high stability and robustness. Thanks to the sequencing engine that streams sound directly from disk using a small buffer window (32768 frames), the system allows for a minimal RAM usage while simultaneously playing dozens of audio files. The sequencing engine also manages the sound event generation process driven by graphs: each time an actant triggers a new sound object, the sequencing retrieves the associated sample from the database and adds it to the list of active sound object initializing its streaming buffer. The engine applies to each activated sample a control envelope that performs a fade in and a fade out proportional to the sample duration. The spatialisation engine manages the list of listener objects: each listener processes the list of active buffers, creates its own temporary copy of the buffers and performs its relative soundscape mix considering its position and orientation. The spatialisation engine produces a stereophonic stream but being split from the sequencing, the engine could easily re-use external libraries for other spatialisation set-ups or models. Both engines deals with multi-threading events synchronization.

The application implements parsing methods that initialise the soundscape software representation starting from the user space, sequencing and database annotations in the several formats previously discussed. During the initialisation phase the application performs the automatic graph generation per each concept. For continuous concept, it creates a graph whose edges have durations proportional to the duration of the vertex (i.e. the sample) they start from, so that the resulting triggering path will generate a continuous stream of concept events. The durations are not equivalent but proportional because the edge duration is slightly shorter in order to superpose events and to provide a cross-faded concatenation. The arrhythmic generation (AR) is a parameter that controls the number of input and output edges from each graph vertex. The number of edges is balanced: the system always sets the connection between the vertices that are less connected. The more

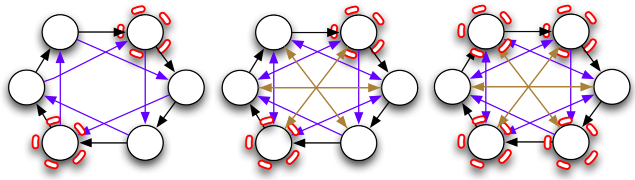

Figure 5. The type of sequencing for a 3 graph having the same global probability $= 80$, just changing the AR and MP parameters

edges each vertex has the less predictable the sequencing will be.

For a non continuous concept, AR also represents the irregularity in the sound object generation process. In fact, for this type of concept, the edge duration is proportional to the general concept probability and the multiple path parameter (MP), considering that each further actant doubles the effective probability. The edge durations are computed recursively, in the sense that each further cycle of edge connections between vertexes have the same values. Considering the simple case where edges probability are uniform, this means that edges probability are equal to the number of output edges from a vertex (i.e. 1/AR). Starting from the case where AR $= 1$, each vertex has just one output (and input) edge, this means that all the pauses between sample activations are the same and the triggering pattern is completely regular, like a metronome. The number of vertices does not change the probability, but the more vertices (i.e samples) a concept has, the less repetitive and poor in nuances its generation will result. First we consider the effective probability p taking into account the actual number of actants, which differs from the declared probability p_d provided by the user in the concept parameter:

$$p = \frac{p_d}{MP}$$

Thus, the duration of all graph edges per AR $= 1$, considering $d =$ edge duration in second and $h = 3600$ seconds, is:

$$d = \frac{h}{p}$$

If AR $= n$, vertexes has n number of input/output edges and each further cycle i of edge creation has a duration that is proportionally lower than the previous one, always granting the overall probability:

$$\sum_{i=1}^{AR} d_i = \frac{h}{p}$$

where

$$d_i = d_{i-1} * 2^i$$

Summarizing, in figure 5 and in table 1 we show how the sequencing can generate completely different triggering pattern having the same global probability. In the example $p = 80$ occurrences per hour.

Once the application parsed the annotation, thus creating the soundscape and the graph associated to each concept, the audio generation starts using the sequencing and

MP	*AR*	d_1	d_2	d_3	d_4	d_5
2	2	120	60			
2	5	58	29	14.5	7.25	3.62
4	5	464	232	116	58	29

Table 1. The edge duration d_i per each further creation cycle i with AR the number of cycle (input/output edges per vertex)

spatialization engines. At this stage the application provides a OSC interface along with a GUI to allow both user or external application to interact in real-time with the graphs and the synthesis models controlling the soundscape performance. Externals requests can:

- Create listeners
- Update listeners position and orientation
- Control zones amplitude
- Control concepts amplitude
- Control concepts probability requiring more multiple generation processes (i.e. increasing MP)
- Control concepts probability requiring to change the edge durations (i.e. recomputing the graph)
- Update concept position
- Create pre-set: map a global soundscape status to an high-level general description

Pre-sets allow to save all the soundscape parameters (positions, mix and probability) mapping them to an high-level description that could be activated on request in real-time. For instance it is a powerful tool to model typical time period or particular state of the soundscape, like: night, day, windy, holiday, winter and so on.

As a conclusive note, Iain McGregor [4] conducted a qualitatively experiment [12] to test the perception of spatialisation in a Canary Island soundscape use-case generated with the proposed application. The experiment was conducted with 20 listeners. The results were very positive but the main problem observed was the perception of distance. Often users wrongly perceived very far sound objects considering them closer than the actual soundscape design. We believe that this issue relies on the lack of reverberation cues in our spatialisation engine that just uses the implicit audio sample reverberation without computing a procedural re-synthesised reverberation based on distance and on the acoustic space features specific to each zone.

5. STREAMING MODULE

The generative application runs on a public server hosted by the UPF at `http://mtg110.upf.es`. The server

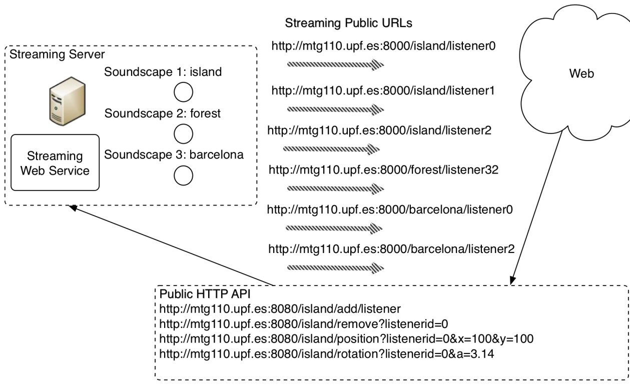

Figure 6. Streaming server and public API

service is developed using Icecast [5] as audio streaming server and the python based Twisted framework to manage the networking engine [6]. This module delivers the produced soundscapes as MP3 streams and it implements a public interface for the SuperCollider application that allows for easily access, interact and control the soundscape generation through public HTTP API. The server has 2 types of API, one that creates new stream on demand and one that modifies the streams. Each new soundscape has its own streaming URL, then each "Add Listener" request creates a nested streaming URL as figure 6 shows. The "Remove listener" request frees the streaming resource. Instead the "Listener position update" requests, that controls the spatialisation engine, and the "Performance update" requests, that controls the sequencing engine, are translated on OSC and routed to the generative application.

6. CONCLUSIONS

There are not completely similar approaches in soundscape generation. What we propose is an innovative work, both in the idea of providing a web service for delivering listener based spatialised soundscape, and an interface to simplify the soundscape design also encompassing user-generated recordings. An interesting work that proposes an automatic sonification system of geographic sound activity controlled by acoustic, social and semantic informations is [13], but it is not aimed to high quality sound design standard and the system can not be used directly in multimedia productions, in spite of being an interesting sound events geographical browsing application. Outside from the environmental sound context, [14] presents a powerful tool to control the generation of musical events from a web map interface, mapping geo-related data to a musical interface. Furthermore, a related approach that focuses on the techniques for environmental sound retrieval is [15] but it does not take into account the generation part.

We imagine that possible clients of our system are Virtual Social Worlds (e.g we design a dedicated sonification in Second Life for a Metaverse project use-case, see section 7), Online games, web Mapping Viewer (e.g. Google maps) and augmented reality tourist application delivering

[4] PhD Student at School of Computing Edinburgh Napier University working on a mapping tool for comparing auditory display interfaces with listeners experiences

[5] `http://www.icecast.org/`

[6] `http://twistedmatrix.com/`

Soundscape Radio through the GPS technology (e.g. cars, mobile)[7]. Also architectural rendering for project presentation or urban planning makes use of video game technology to create 3D real time graphics simulations and do not consider the audio cues. Finally expert users, sound designer and composer, could also deploy and integrate the system in concert, sound art or interactive installation using the generative module as a local application.

The proposed system provides a standard composition format, making soundscape design more easily accessible and controllable by others applications or clients.

6.1 Future steps

An online platform that dynamically manages several generative module instantiations along with a soundscape web graphic editor are the future steps to realise the soundscape web service. From this platform clients could explore the already available composition scores searching for a desired space, and in case of need, proceed with a dedicated new design.

7. ACKNOWLEDGMENTS

This work was partially supported by the ITEA2 Metaverse1 Project `http://www.metaverse1.org`.

8. REFERENCES

[1] Universitat Pompeu Fabra, "`http://www.mtg.upf.edu/technologies/soundscapes`," 2010. MTG Soundscapes project web page with documentation and resources.

[2] A. Farnell, *Designing Sound - Practical synthetic sound design for film, games and interactive media using dataflow*. London: Applied Scientific Press Ltd., 2008.

[3] S. Kersten and H. Purwins, "Sound texture synthesis with hidden markov tree models in the wavelet domain," in *The 7th Sound and Music Computing Conference*, (Barcelona), 2010.

[4] D. Schwarz and S. Norbert, "Descriptor-based sound texture sampling," in *The 7th Sound and Music Computing Conference*, (Barcelona), 2010.

[5] Google, "`http://code.google.com/intl/ca/apis/kml/`," 2010. KML format.

[6] A. Valle, V. Lombardo, and M. Schirosa, *Auditory Display 6th International Symposium, CMMR/ICAD 2009, Copenhagen, Denmark, May 18-22, 2009. Revised Papers*, vol. 5954, ch. Simulating the Soundscape through an Analysis/Resynthesis Methodology, pp. 330–357. Berlin: Springer, 2010.

[7] R. Murray Schafer, *The Tuning of the World*. Toronto and New York: McClelland & Steward and Knopf, 1977.

[8] A. Valle, "Geography: a real-time, graph-based composition environment," in *NIME 2008: Proceedings*, pp. 253–256, 2008. Antonio Camurri and Stefania Serafin and Gualtiero Volpe.

[9] Universitat Pompeu Fabra, "`http://www.freesound.org`," 2005. Repository of sounds under the Creative Commons license.

[10] G. Roma, J. Janer, S. Kersten, M. Schirosa, and P. Herrera, "Content-based retrieval from unstructured databases using an ecological acoustics taxonomy," in *Proceedings of the International Community for the Auditory Display Conference*, 2010.

[11] S. Wilson, D. Cottle, and N. Collins, eds., *The SuperCollider Book*. Cambridge, Mass.: The MIT Press, 2008.

[12] I. McGregor, G. LePlâtre, P. Turner, and T. Flint, "Soundscape mapping: a tool for evaluating sounds and auditory environments," in *The 16th International Conference on Auditory Display*, (Washington D.C.), 2010.

[13] A. Fink, B. Mechtley, G. Wichern, J. Liu, H. Thornburg, A. Spanias, and G. Coleman, "Re-sonification of geographic sound activity using acoustic, semantic, and social information," in *The 16th International Conference on Auditory Display*, (Washington D.C.), 2010.

[14] S. Park, S. Kim, S. Lee, and W. S. Yeo, "Composition with path: musicla sonification of geo-referenced data with online map interface," in *Proceedings of the International Computer Music Conference (ICMC)*, (New York), 2010.

[15] B. Mechtley, G. Wichern, H. Thornburg, and A. Spanias, "Combining semantic, social, and acoustic similarity for retrieval of environmental sounds," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 2402 – 2405, 2010.

[7] In this sense an interesting example is `http://www.soundwalk.com/`