SONIFICATION OF A STREAMING-SERVER LOGFILE

Wolfgang Hauer

University of Music and Performing Arts Graz
Institute of Electronic Music and Acoustics
Inffeldgasse 10/3, 8010 Graz, Austria

w.hauer@student.tugraz.at

Katharina Vogt

University of Music and Performing Arts Graz
Institute of Electronic Music and Acoustics
Inffeldgasse 10/3, 8010 Graz, Austria

katharina.vogt@kug.ac.at

ABSTRACT

This paper presents the sonification of a multimedia streaming server based on its log file. The server is used at a technical university to provide lecture recordings. Due to the large number of datapoints and various data categories it's a promising approach to use sonification as a display method instead of the traditional visual representation.
The SuperCollider script allows to simulate a realtime scenario as well as to monitor a past period of time.

1. INTRODUCTION

With growing internet bandwidth the number of multimedia streaming services increased in the last few years. Besides entertainment services, more and more informational and educational content is available online. To provide this content reliably, multimedia streaming servers have to be managed and monitored almost permanently.
Because of the large and complex log files of multimedia streaming servers it can be advantageous to use an auditory display instead of traditional diagrams to represent relevant data. Although there are a few programs that allow monitoring network data with sound (such as "NeMoS" [1] or "Personal Webmelody" [2]), there are no programs that sonify log files of streaming servers. Both, NeMoS and Personal Webmelody, create realtime datastreams by using the standard web protocols SNMP or HTTP and associate MIDI tracks to predetermined events. Personal Webmelody additionally offers the feature of mixing an external music source with system-generated music.
In this paper the sonification of a technical universitys streaming server log file is presented. The log file contains a large number of datapoints (approx. 9000) each whith many catgories like date and time of access, amount of data streamed between server and client, the clients ip-address and user-id, the name, size and type of the streamed file etc.
The sonification, which was created during a university seminar, represents two chosen categories:

- the amount of data streamed from server to client per access
- seven different lecturers

The script was written for the SuperCollider programming language [3] and is able to simulate realtime monitoring or to represent data of a chosen period of time using logfile data.

2. SONIFICATION APPROACH

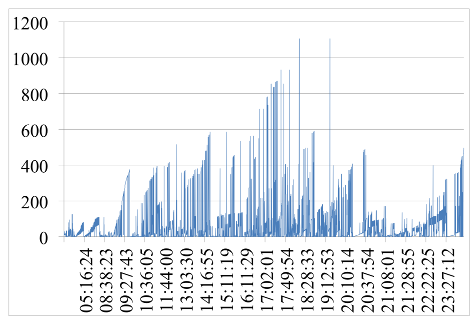

Figure 1: Plot of data streamed from server to client in Megabytes over a period of one day, where all seven lecturers are mixed together.

The log file needs to be adapted by hand before the SuperCollider script is able to read it. The most important and time consuming parts of editing are to delete irrelevant datapoints like internal server events and to anonymize user information (like IP addresses or user-IDs).

For the chosen data categories, two sounds are generated in real time and played simultaneously. Data of each parameter are normalized within the range of the whole data set. The first sound is a continuous, pitched/ resonated noise, called "Whistlesynth". The second sound is based on granular synthesis of the lecturers themselves, i.e., short excerpts of the voices of the lecturers, which we call "Grainsynth".

2.1. Whistlesynth

Here, sound is generated by a resonator (`Klank`) which is triggered by brown noise. The frequency of the resonator depends on the amount of data streamed from server to client (see Fig.1) - the higher the amount of streamed data, the higher the resonance frequency of the resonator. This generates a whistle like sound between 1000 Hz and 4000 Hz.

Controllable parameters are:

- resonance frequency
- amplification
  the more data is transmitted the louder the sound
- position in the stereo panorama
  to allow temporal orientation, the sound can move in a continuous way from left (00:00) to the center (12:00) to right (00:00)

2.2. Grainsynth

To sonify different lecturers a granular synthesizer (`GrainBuf`) is used. The grains are generated from sound samples which again are taken from corresponding lecture video clips. Attempts were made to create a sound which can be recognized as voice but does not allow understanding words. Each speaker is assigned to a fixed position: this allows to get an idea about different speakers even if the grains are very short and the succession of different lectures downloads is quick.

Controllable parameters are:

- length of grains
  the duration of grains depends on the amount of streamed data. If many data are downloaded from one speaker, the grains become longer and are better perceived as speech-like.
- playback position in the grain
  can be chosen either to create comprehensible language or to create a more abstract sound [4].
- position in the stereo panorama
  each speaker is assigned to a fixed position

3. APPLICATIONS

The script is intended to be used by multimedia streaming server administrators in their daily work.
Two possible practical applications are for example:

3.1. Realtime monitoring

To enable realtime monitoring the script has to be adapted so that it is possible to process data received directly from the server (see [5]). Up to now, it is only possible to simulate realtime, which means that the synthesizers are not triggered by actual server accesses but from a random process with approximately the same pace (approx. 2-6 accesses/second).
In this way, the sonification could be received in the backround all the time and therefore enables permanent monitoring of server activities.

3.2. Monitoring past periods of time

The second application is to get an overview over a period of time using a saved log file. While this could also be done with traditional graphical interfaces, the advantage of the sonification is the possibility to monitor several categories simultaneously.

4. CASE STUDY

Three versions of parametrization of this sonification design have been developed and tested in an informal questionnaire. In version 1 Grainsynth generated a very abstract sound while in version 2 a sound similar to voice and in version 3 comprehensible language was generated [4].
The 5 test participants were asked how many lecturers they were able to identify, to draw approximately the amount of streamed data shown in Fig.1 and to judge whether the sound of each version was rather pleasant or unpleasant.
While none of the participants was able to identify all 7 different lecturers (the answers varied between 4 to 6) all of the drawings were very similar to Fig.1.
Three participants assessed the first version as rather pleasant and the second and third version as rather unpleasant. Two participant assessed version 1 as very unpleasant, version 2 as rather unpleasant and version 3 neither unpleasant nor pleasant.
While this evaluation is only a short pilot test, it gives some feedback about the sonification's usability and sound quality. It showed that it is principally possible to distinguish and memorize different speakers and to visualize the temporal development of the streamed data.

5. OUTLOOK

The next step is to adapt the script so it's possible to receive and sonify data directly from the streaming server. Moreover it's crucial to find a better trade off between a pleasant sound which can be listened to over a long period and a more analytic sound.

6. ACKNOWLEDGMENT

I would like to thank the seminar participants for their input and their participation in the survey. Furthermore I would like to thank IOhannes Zmoelnig for his suggestions of similar projects.

7. REFERENCES

[1] D. Malandrino, D. Mea, A. Negro, G. Palmieri and V. Scarano, "NeMoS: NETWORK MONITORING WITH SOUND" in *Proceedings of the 2003 International Conference on Auditory Display*, Boston, MA, USA, July 6-9, 2003.

[2] M. Barra, T. Cillo, A. De Santis, U. Ferraro Petrillo, A. Negro, V. Scarano. "PERSONAL WEBMELODY: CUSTOMIZED SONIFICATION OF WEB SERVERS" in *Proceedings of the 2001 International Conference on Auditory Display*, Espo, Finland, 29 July - 1 August 2001.

[3] Wilson, S., Cottle, D. and Collins, N. (eds). 2011. The SuperCollider Book. Cambridge, MA: MIT Press

[4] `http://iaem.at/kurse/ss15/sonifikation-sound-of-science-se/sounds-server`