POSTER

PRELIMINARY STEPS IN SONIFYING WEB LOG DATA

Mark Ballora

School of Music/Department of Integrative Arts
The Pennsylvania State University
30 Borland Building, University Park, PA 16803
ballora@psu.edu

Brian Panulla

The Pennsylvania State University
College of Information Sciences and Technology
Center for Network Centric Cognition and
Information Fusion
brian@panulla.com

Matthew Gourley

The Pennsylvania State University
Administrative Information Services
mmg207@psu.edu

David L. Hall

The Pennsylvania State University
College of Information Sciences and Technology
Center for Network Centric Cognition and
Information Fusion
dhall@ist.psu.edu

ABSTRACT

Detection of intrusions is a continuing problem in network security. Due to the large volumes of data recorded in Web server logs, analysis is typically forensic, taking place only after a problem has occurred. We are exploring the detection of intrusion signatures and patterns via an auditory display. Web log data is parsed and formatted using Python, then read as a data array by the synthesis language SuperCollider [1], which renders it as a sonification. This can be done either for the study of pre-existing data sets or in monitoring Web traffic in real time. Components rendered aurally include IP address, geographical information, and server Return Codes. Users can interact with the data, speeding or slowing the speed of representation (for pre-existing data sets) or "mixing" sound components to optimize intelligibility for tracking suspicious activity.

1. INTRODUCTION

While the primary task of the sciences may be exploration and the discovery of new knowledge, a critical issue currently facing scientists and researchers is in the area of presentation -- the ability to introduce their discoveries effectively, both to laypeople and to fellow researchers. There is an emerging area of interest in representation of scientific information, and how the use of multi-media technologies, an essential component in disseminating new information, can in turn shape and influence scientific thought [2].

The problem is not only that of dealing with new forms of information, but also with unprecedented quantities of it. In our Information Age, new forms of gathering information are constantly being created. However, this does not necessarily lead to increased understanding. In particular, managing crisis situations or monitoring infrastructures requires the ability to interpret incoming information from multiple sources. With new sources of information constantly becoming available, the challenge becomes how to process it effectively, avoiding the condition described by informatics researchers as *cogmenutia fragmentosa* [3].

Penn State's Center for Network Centric Cognition and Information Fusion (NC^{2}IF) [4] housed in Penn State's College of Information Sciences and Technology, explores the information chain from energy detection via sensors and human observation to modeling, signal and image processing, pattern recognition, knowledge creation, information infrastructure, and human decision-making [5].

The Center's Extreme Events Lab (EEL) is intended to allow researchers to run end-to-end experiments that improve situational awareness and enhance their ability to optimally leverage all available sensors, human observers, and technology in order to escape "information overload" and extract the true meaning hidden within the vast mountains of available data [6].

There is a body of work in the field of cyber security that examines intrusion detection in terms of information theory, noting that the complexity of network activity drops during intrusion attempts [7, 8].

Here we describe initial steps in an experiment created for the EEL in which Web log data is rendered as a sonification. Our goal is to determine whether intrusion attempts produce recognizable patterns that can be detected aurally, either in real time, or as an after-the-fact analysis. Results of this work will become part of a collective pool of methodologies used in ongoing data rendering experiments carried out by the center.

2. PREDECESSORS

This project is related to earlier work [9-11] involving network activity sonifications. Most directly related was the Peep Network Auralizer System [9], which played back various recordings to reflect network conditions such as incoming and outgoing email, load average, number of users logged in, etc. Through various sound libraries, akin to SoundFonts, listeners could be placed in a variety of listening environments – rainforest, desert, and so on. The amount of rain might represent load average, the flow of a waterfall might represent email traffic, and so on. The creation of nature-inspired soundscapes was an effective design choice, as it resulted in a pleasant and unobtrusive listening environment. However, the limitation of this approach is the same as found in sampling synthesizer instruments: simple *triggering* of audio files lacks "control intimacy" (as described by Moore in [12]) whereby variations in the data create variations (often subtle) in the creation of sound, in a manner akin to a musician's many microscopic gestures that affect the sound and character of an instrument during performance.

A closer level of control intimacy is achieved with *parameter-based* sonifications [13], which link the data to the sound at a deeper acoustic level than is possible with simple triggering. The data values are mapped to synthesized sound characteristics such as oscillator frequency, filter cutoff frequency, volume, stereo panning, etc. This methodology runs the risk of turning into "bleep bloop" music that can be a trying listening experience. The key to success lies in effective orchestration strategies. A multi-dimensional data set is sonified as a multi-instrumental synthesizer ensemble. The design challenge is to create timbres that complement each other well when they are combined to represent data dimensions. Parameter-based sonifications also have the potential limitation of being arbitrarily contrived, so that users may have difficulty learning which auditory characteristics reflect which data dimensions, which may have no apparent intrinsic connection.

A further level of control intimacy is gained with another approach, termed *model-based* sonification [14]. This involves mapping data values to resonances and/or mechanics of a *physical model*, typically in a form that can be explored interactively by the analyst, creating inextricable relationship between the data and the resulting sound event. A physical model is a computer synthesis technique based on wave equations that describe vibrating objects. An example model-based sonification might be a multi-dimensional data set mapped to a theoretical grid of masses and springs, simulating a virtual instrument. As the data iterates, the character of the instrument's vibrations changes. This allows the possibility of subtle patterns to emerge within the quality of the sound field that would be lost with realization based on simple triggering, and are more inherently integrated than parameter-based sonification methodologies.

The modeling idea is explored in a simplified form in our sonification, in that we use a simple physical model, although without the high-dimensionality and user navigability of many model-based sonifications. We create additional renderings that are parameter based for a more qualitative realization, as well as triggered sounds that are used for certain alerts. The synthesis parameters were chosen for aesthetic reasons, in some cases to create a pleasing sounding musical instrument, in others to create a pleasant nature-like soundscape.

3. WEB LOG DATA

When someone tries to load a Web page by typing a URL, clicking a link, or by submitting data via a Web-based form, that person's browser sends an HTTP Request to the Web server, which in turn sends back an HTTP Response in the form of a Return Code. The Return Codes consist of a numeric ID, often followed by a text explanation. The ranges of the numeric IDs indicate various levels of Success (2xx), Redirection (3xx), Client Error (4xx), or Server Error (5xx). This exchange is typically invisible to users, although one commonly seen Return Code is the familiar *404: Page Not Found*.

The data set used for our sonification consists of 11,350 entries, which originate from a filtered set of HTTP Requests made to the Web server at Penn State's College of Information Sciences and Technology. The requests span a 24-hour time period. Each point in our data set is an array consisting of information taken from an HTTP Request as well as the corresponding Return. Each array entry includes:

- a timestamp,
- the Request's source IP address,
- the latitude and longitude of the Request,
- the Return Code sent by the server.

4. SONIFICATION STRATEGIES

4.1. Iteration

The data set is loaded into SuperCollider as an Array. A Task process is run, with each iteration loading values from the next array member into variables. Synth objects are then instantiated that use the variables as controls of its various aspects (frequency, modulation rate, and so on).

4.2. Time Stamps

Each data array triggers a sound event in the sonification. The rendering for sound events is based on the relative times between timestamps, multiplied by a scalar. Thus, periods of higher or lower relative activity can be easily recognized, depending on whether one hears sparse, occasional events or a flurry of sound. A pre-existing data set consisting of many hours of activity can be heard over a timescale on the order of minutes or seconds, depending on the iteration rate the listener chooses.

4.3. IP Addresses

The principal focus of this work involves translating the IP addresses of HTTP Requests into sound. Our question is whether a coordinated set of Requests from a single address or a set of related addresses would create an auditory signature of some kind.

IP addresses are typically written in "dot-decimal notation," which consists of four octet values derived from a 32-bit network identification number, as shown in Table 1.

a 32-bit binary identification number

01100100100101101100100011111010

is divided into four 8-bit octets

01100100.10010110.11001000.11111010

commonly written with each octet as a
decimal number in the range of 0-255

100.150.200.250
most significant octet least significant octet

Table 1. IP Address Structuring

We treat the octets separately, thus treating each source IP address as a four-dimensional data point, with the most significant values represented by the leftmost octet, and the least significant values represented by the rightmost.

The binary nature of IP addresses suggests a compatibility with the numbering system used in MIDI (Musical Instrument Digital Interface), a common software protocol understood and transmitted by synthesizer instruments. A note number of 60 is assigned to middle C, with successive values above or below 60 corresponding to half steps above or below middle C.

As a preliminary step, the IP octet values are mapped to MIDI note values. Each octet (0-255) is remapped to a value within a span of 23 half steps. Since the initial range of 0-255 is much larger than the mapped range of 0-23, the resulting values are floating point, meaning that most of the mapped frequencies are microtones, falling between the half-steps represented MIDI integer values.

4.4. Vibraphone

A set of four resonances can describe the timbre of a vibraphone, as shown by the spectrogram in Figure 1. It can be observed that the timbre consists of four resonances that closely correspond to the fundamental, fourth, 10th and 17th harmonics.

We start with the quartet of MIDI note values derived from the source IP address, described in the last section, and transpose them further, such that they function as four harmonics falling roughly within the ranges of a vibraphone's partials. These values are then used in an instantiation of an vibraphone-like instrument that is created with SuperCollider's Klank unit generator, which is a simple and general physical model consisting of an arbitrary number of resonant frequencies with relative amplitudes and ring times:

```
SynthDef("ipVib", {arg fund=293,
        formant1=1173, formant2=2930, formant3=4986,
        v1=1, v2=1, v3=0.3, v4=0.3, pos=0.0;
e=Env.new([0, 1, 1, 0], [0.01, 1.5, 0.01], 'linear');
k=DynKlank.ar(`[[fund, formant1, formant2, formant3],
                [v1, v2, v3, v4],
                [1.5, 1.0, 0.25, 0.1]],
                Impulse.ar(0, 0, 0.1));
p=PanAz.ar(~numChans, k, pos, 1, 3);
Out.ar(0, p*EnvGen.ar(e, doneAction:2))
        }).send(s);
```

Each time the Task iterates, a single impulse (the digital audio equivalent of a percussive strike) is sent to an instantiation of the vibraphone instrument, with four resonances mapped from the values of data's IP address. Thus, each HTTP Request in the data set triggers an instance of the vibraphone-like instrument in the sonification.

The result is a quick, active vibraphone melody. The four octet/frequency values are not heard as a discrete chord, but rather they fuse into a unified timbre with shifting resonances. A short two-channel excerpt can be heard online at http://dl.dropbox.com/u/4128606/vibraphone-like.wav.

4.5. Babbling Brook

The Peep system, mentioned earlier, created a pleasant and informative nature-scape. In an effort to appropriate this idea, an alternate rendering is created of each IP address that is meant to sound like a brook or creek.

This sound model is derived from a synthesis example titled

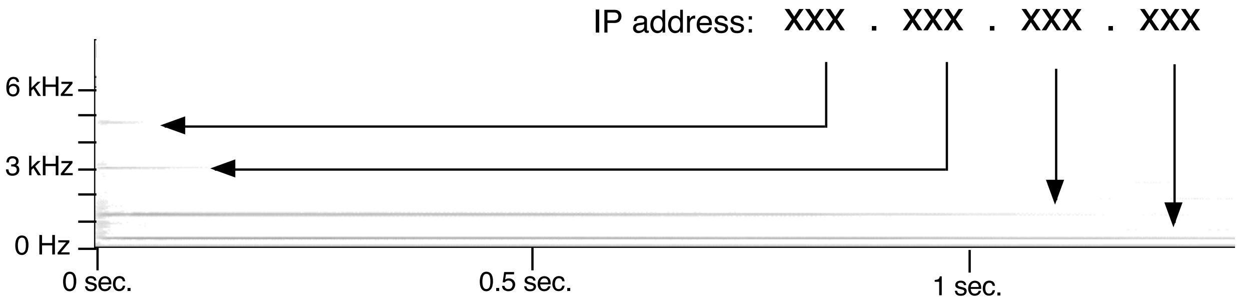

Figure 1. Spectrogram of vibraphone Middle D (293 Hz), with the assignment of IP octets to the most significant partials.

"Babbling Brook," which was created by SuperCollider's inventor, James McCartney, and is included as an example that comes with the SuperCollider program's documentation. The patch consists of filtered noise, with a modulating value controlling the cutoff frequency of the filter.

In our adaptation, each source IP address provides parameters for four instances of the water-like instrument, each with its average cutoff frequency and modulation rate based on the value of its source octet.

```
SynthDef("ipnoisedroplet",
    { arg f1=800, f2=17, vol=1.0, dur=1.0, pos=0.0;
      e=Env.new([0, 1, 1, 0], [0.01, dur, 0.01],
       'linear', nil, nil);
      r=PanAz.ar(~numChans,
       RHPF.ar(OnePole.ar(BrownNoise.ar, 0.99),
       LPF.ar(BrownNoise.ar, f2+5)*f1+f1, 0.03, 0.003),
       pos, vol, 3);
      Out.ar(0, EnvGen.ar(e, doneAction:2)*r);
   }).send(s);
```

The result is a diffuse water-like soundscape that is abstractly related to the IP addresses in the dataset. Since there is no inherently musical relationship between the octet values in IP addresses, the option of creating the babbling brook soundscape allows the possibility of a more qualitative rendering that may be preferable to the musical relationship somewhat imposed by the vibraphone rendering. A short two-channel excerpt can be heard online at http://dl.dropbox.com/u/4128606/water-like.wav.

4.6. Vocal Synthesis

Since a set of 3-5 formants (consistent spectral peaks applied to a signal containing many harmonics) is sufficient for basic vowel synthesis, a third rendering is created that models vowels created from each source IP address. This is somewhat similar to the idea of the vibraphone-based timbre, with the distinction that the four values are mapped to different frequency regions so that they fall within the ranges of vocal formants instead of vibraphone harmonics. The four frequency values are also not given relative ring times, but rather they all sound continuously. A continuous sound that modulates as the data set iterates creates a vocal-like drone that has a shifting vowel quality.

4.7. Server Return Codes

In addition to creating sounds mapped from source IP addresses, an additional annotation is given to each data point by creating a sound based on the Web server's Return Code. Different percussive sounds are assigned to the various Return Codes, such as a model of two river stones clicking together, or noise bursts to suggest splashing, or a pitched ringing sound. These are meant to highlight occurrences of different Return messages, so that a repeated error message, for example, can be made salient.

4.8. Request Location

Since SuperCollider allows sounds to be panned over an arbitrary number of channels, we also sonify longitude of each Request as stereo localization, panning the sound event within an octaphonic ring of loudspeakers, placing the listener in "the center of the world." At this writing, a rendering of latitude is being explored. A likely design will be a model resembling a gong, so that different "elevations" can be represented by differing strike pressures, so that a low tapping can represent lower latitudes, with a louder, ringing strike indicating higher latitudes.

5. INTERFACE

The overall soundscape of the sonification is controllable by a mixing-board like interface. Its functionality follows the design of our earlier work in cardiac data sonification [15]. Users can start, pause, and reset playback via buttons; one slider can speed up or slow the rate at which the dataset is traversed; another slider allows the dataset to be scrubbed so that playback may start from any arbitrary point, with the timestamp of the current position displayed visually. Each sound component has separate volume slider, allowing the overall mix to be controlled.

The interface is displayed on a Lemur LCD controller [16]. This customizable device can send messages to the IP addresses of musical devices via Open Sound Control (OSC). A Lemur can potentially control a synthesizer device placed anywhere on the World Wide Web. Lemur interfaces are created in software on a computer, with objects taken from a palette of knobs, sliders, and other interface elements, and each named individually.

SuperCollider has a class called Lemur, by which an interface can be recognized by its IP address. It can read information from the named objects in the Lemur interface (for example, the position of an object called Slider1), and assign it to a variable within the synthesis patch (for example, a variable assigned to an oscillator's volume value).

6. INITIAL RESPONSES

Development and testing of the sonification system is being carried out as this is being written. But it is apparent that two base conditions have been met. One is that coarse changes are audibly obvious: for example, repeated Requests from a single location are readily apparent, even to the least musically trained ears. Another is that of persistence: the sound quality makes a pleasing and unobtrusive backdrop. This was informally assessed during a wine and cheese event commemorating the tenth year of the College of Information Sciences and Technology. Visitors were invited to visit the EEL and other facilities. As people wandered in throughout the evening, the initial reaction of visitors was quite favorable. Granted, this soft anecdotal evidence hardly constitutes academic merit in and of itself. However, a pleasant listening experience is an essential component of a successful sonification, and is therefore an essential criterion for evaluating merit at this initial stage of the work.

7. FUTURE WORK

The work to date has been on design issues, exploring how the information can be mapped effectively. The next step is to establish proof of concept by evaluating our design with data sets containing known intrusion attempts. Well-documented case sets are publicly available [17], which allow for initial proof of sonification concept.

It is likely that other characteristics of a Web log data set can be usefully sonified, beyond simple renderings of the requesting IP address. Subsequent iterations of this work will explore anomalous log entries, such as unusually long requests, which are often associated with attempted database intrusions.

Larger-scale analyses will likely need some higher levels of abstraction in rendering, as the density of data may become unwieldy or incoherent when each point is sonified. While having this microscopic level present is desirable to ensure the integrity of the rendering, it is also desirable to be able to introduce various types of averaged and statistical data.

We also project creating a real-time renderer. This will likely be in the form of a daemon written in Python that receives copies of Web log entries, parses them, and reformats them as OSC messages, which it then sends to SuperCollider.

8. CODA

It is unlikely that Web system administrators will feel inclined to purchase octaphonic sound systems and Lemur interfaces. However, there has been interest expressed in using some form of auditory monitoring of server traffic at Penn State, and certainly scaled-down versions could easily be created for practical implementations. But it is more to the point to bear in mind that NC^{2}IF functions as a collective, whereby ideas are regularly shared among people working on a variety of projects. It is entirely possible that an interesting idea created for one project may turn out in practice to be more suitable for another project, as various forms of data renderings and data fusion are explored. It is thus in our interests to explore all possibilities for representation, rather than potentially limit ourselves by setting out to create a fixed product for this rendering.

9. REFERENCES

[1] http://supercollider.sourceforge.net/

[2] *Visualizing Science: Image-Making in the Constitution of Scientific Knowledge (an interdisciplinary symposium)*. October 24, 2007, Brandeis University. http://culturalproduction.wikispaces.com/visualizing_science

[3] McNeese, M. D. and Vidulich, editors, *Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogmenutia Fragmentosa,* Dayton, Ohio, Wright Patterson Air Force Base, CSERIAC Press, 2002.

[4] http://nc2if.psu.edu/

[5] Hall, D., C. Hall, S. McMullen, M. McMullen, and B. Pursel, "Perspectives on Visualization and Virtual World Technologies for Multi-Sensor Data Fusion," *Proceedings of the 11th International Conference on Information Fusion,* Cologne, Germany, June 30- July 03, 2008.

[6] Hall, D., B. Hellar, and M. D. McNeese, "The Extreme Events Laboratory: A Cyber Infrastructure for Performing Experiments to Quantify the Effectiveness of Human-Centered Information Fusion." *Proceedings of the 2009 International Conference on Information Fusion (Fusion 2009),* Seattle, Washington, July, 2009.

[7] Evans, S.C. and B. Barnett. "Network Security through Conservation of Complexity." *Proceedings of MILCOM 2002 Military Communications Conference*. October 7-10, 2002, Anaheim, California, IEEE.

[8] Eiland, E. E., and L.M. Liebrock, "An Application of Information Theory to Intrusion Detection." *Proceedings of the Fourth IEEE International Workshop on Information Assurance.* IEEE Computer Society, 2006.

[9] Gilfix, M. and A. Couch, "Peep (The Network Auralizer): Monitoring Your Network with Sound." In *2000 LISA XIV*. December 3-8, 2000 – New Orleans, LA, pp. 109-117.

[10] Chafe, C., and R. Leisikow, "Levels of Temporal Resolution in Sonification of Network Performance," Proc. 2001 Intl. Conference on Auditory Display, Helsinki, 2001.

[11] Chafe, C., and S. Wilson, D. Walling, "Physical Model Synthesis with Application to Internet Acoustics," Proc. 2002 Intl. Conference on Acoustics, Speech and Signal Processing, Orlando, 2002.

[12] Moore, F.R. *Elements of Computer Music*. Englewood Cliffs, NJ: PTR Prentice Hall, 1990.

[13] Kramer, G., ed. *Auditory Display: Sonification, Audification, and Auditory Interfaces*. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. XVIII. Reading, MA: Addison Wesley, 1994.

[14] Hermann, T. *Sonification for Exploratory Data Analysis.* Ph.D dissertation, Bielefeld University, 2002.

[15] Ballora, M. and B. Pennycook, P. Ch. Ivanov, L. Glass, A. L. Goldberger. 2004. "Heart rate sonification: A new approach to medical diagnosis." *LEONARDO* 37 (Feb. 2004): pp. 41-46.

[16] http://www.jazzmutant.com/lemur_overview.php

[17] Lippmann, R. P. [et al.]. "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-Line Intrusion Detection Evaluation." *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX), Vol. 2. 2000*: IEEE Computer Society Press: Los Alamitos, CA. p. 12-26.