SONART : THE SONIFICATION APPLICATION RESEARCH TOOLBOX

Oded Ben-Tal, Jonathan Berger, Bryan Cook,
Michelle Daniels, Gary Scavone

The Center for
Computer Research in Music and Acoustics
Stanford University
Stanford, CA USA
brg@ccrma.stanford.edu

Perry Cook

Computer Science Department
Princeton University,
Princeton, New Jersey, USA
prc@cs.princeton.edu

ABSTRACT

The Sonification Application and Research Toolbox (SonART) is an open-source effort to develop a platform-independent collection of methods to map data to sonification parameters. A set of graphical user interface tools that will provide practical and intuitive utilities for experimentation and auditory display. SonART aims to provide publicly available, well-documented code that will be easily adapted to address a broad range of sonification needs. The effort will build upon the Synthesis ToolKit (STK) [1]. In this paper we describe SonART's parameter engine framework, its interface to STK, and relevant recent developments in STK. We present an example of sonified stock market data to illustrate the principles of SonART.

1. INTRODUCTION

Data sonification is increasingly contributing to scientific research and education. Many current research areas produce enormous amounts of data, setting new demands and challenges for data analysis and mining. Beyond the sheer enormity of many data sets lies the need for new methods to search and analyze data of great complexity. Preliminary investigations on the efficacy of auditory display suggest that this is a promising domain albeit in need of refinement and further investigation. The most problematic aspect of data representation is that interpreting data is influenced and possibly biased by the method in which it is accessed. This is equally true for visualization as for sonification. The impressive sensitivity of the human hearing mechanism and human ability to discern subtle variance in sounds and sound patterns suggest that sonification of data may provide a useful addition to visualization tools for data analysis and interpretation. In numerous real world situations sonification is a standard means of data interpretation. Examples include a physician's interpretation of cardiovascular and breathing sounds through a stethoscope, and an automotive mechanic's detection of mechanical problems by hearing mechanical irregularities. Sonification has been applied to:

- The interpretation of highly dimensional data;
- Interpreting data when visual clustering is confusing;
- Topological structure analysis;
- Revealing trends (such as in tracing the progress of neural network learning);
- Finding patterns in long term dynamic temporal data (such as seismic data);
- Interpreting short-term data in which compression or expansion may be readily achieved acoustically.

Furthermore the inherent hierarchical nature of music can be used to distinguish structurally salient information from complex multidimensional data. A number of types of sonification are currently used. These include:

- Alarms;
- Acoustic representations of place within a data tree;
- Metaphorical use of sounds to represent a state;
- Acoustic feedback in human computer interface [2];
- Parameter mapping (in which a sound's attributes are driven by data values) [3]

By classifying sonification methods, SonART will provide researchers with the means of exploring parameter mapping with the same high-level control afforded by many data visualization packages. Synthesis and sound processing parameters can be classified by general acoustic or musical properties; or by synthesis specific parameters. The parameter mapping engine provides a framework in which the wide range of sonification possibilities can be conveniently organized and easily accessed. The overall framework of SonART consists of three components (fig 1):

- the Synthesis ToolKit.
- the parameter engine.
- the scheduler.

SonART is differentiated from existing toolboxes in the scope of sound synthesis and processing capabilities, and in the flexibility of parameter mapping strategies and methods. The *MUSE* toolkit [4], and its predecessor *Listen* [5] provide methods for high level musical parameter mappings (timbre, rhythm, tempo, volume, pitch, and harmony) with preset sounds and mappings for each type. For example, timbre is bound to a limited number of instrument and vocal sounds. *Personify* [6], provides an open-ended framework for creating and exploring data mappings in soundspaces. The sounds are perceptually linearised allowing for the creation of scales. STK, described in the next section, provides SonART a means of extending this open-ended framework by providing platform independence, adaptability and extendable synthesis and sound processing capabilities.

2. STK: THE SYNTHESIS TOOL KIT

The Synthesis Tool Kit is a collection of classes in C++ intended for the rapid creation of sound synthesis and audio processing systems. Specific design goals have included: cross-platform functionality, ease of use, instructional code examples, and real-time control. STK classes can be divided into three categories: 1) basic audio sample sources and manipulators called unit generators, 2) sound synthesis and audio signal processing algorithms built from unit generators, and 3) control signal and system-dependent handlers. Sound synthesis algorithms include additive (Fourier) synthesis, subtractive synthesis, Frequency Modulation synthesis of various topologies, modal (resonant filter) synthesis, a variety of physical models including stringed and wind instruments, and physically inspired stochastic event models for the synthesis of particle sounds. STK classes form the ”building blocks” for much of SonART's data processing and sonification components. Unit generators include numerous filter classes, envelope generators, audio file input and output classes, and other data sources/sinks. STK synthesis algorithms are used to generate the actual sonified results. The SonART parameter engine is necessitated by the fact that the STK architecture imposes no particular parameter control mechanism on its classes and algorithms, allowing maximum flexibility in the design of an external control structure. STK unit generators can be used in the design of such a structure to provide, for example, parameter smoothing functionality. Recent additions to the algorithmic suite include a new generic physical model of conical bore instruments, simplified wind-based models such as a blown Helmholtz pop-bottle, a 2D waveguide mesh, greatly expanded shaker/particle instruments, and expanded modeling of rigid modal structures using Banded Waveguides.

3. THE PARAMETER ENGINE

Every sound synthesis algorithm or synthesizer has a set of parameters that define how the sound should be generated. The user creates dynamic sounds by changing these parameters with a set of controllers. Data sonification and sound synthesis depend greatly on the parameter-mapping problem of how to map the set of controllers onto the set of synthesis parameters. The most basic part to parameter mapping is determining what controls will affect what parameters, and in what way. It can be useful both to have one control map to several parameters, and to have one parameter be a function of any number of controls.

In order to allow for maximum flexibility SonART uses a parameter Matrix Engine to interactively set the mapping. It provides a simple, intuitive real-time graphical interface that allows for arbitrary mapping of controllers to parameters and definition of the functional mapping of the output based on the set of inputs. Conceptually, the Parameter Engine is similar to an audio matrix mixer. It mixes the set of mapped controls together and assigns it to the output. Across the top of the matrix is the list of available control parameters. For data sonification, this is where the data is fed into the matrix. Along the right side is the list of output parameters. These are usually the synthesis parameters for the output synthesizers. Each input has a data bus that intersects the data bus for each output. Each of the input controls can be mapped to any or all of the output parameters by setting a patch point at the intersection of the two buses. At each of these patch points there is an arbitrary mapping from the input range to an output range, called a Channel FX.This allows for linear and non-linear mapping of the control data. In addition, a Channel FX can be placed to affect the whole input or output bus. The final mix is the sum or the product of all mapped controls depending on whether the output is set to sum or multiply. One key to the parameter matrix is the fact that it keeps track of not just the current value of a parameter, but also keeps track of the range of the parameter. And when the parameter passes through a Channel FX, it not only modifies the value but also makes note of how the range of outputs changes. This allows the user to very easily mix several parameters together but then ensure that the final range of the control is one that is appropriate for the output parameter. The parameter matrix addresses the key issues of the parameter-mapping problem. The matrix layout gives a simple and intuitive interface for mapping controls to parameters, and gives a simple way to create powerful data mappings. the scheduler SonART also provides an extension to the parameter engine's capabilities through the use of an external scheduling module. Because the parameter engine simply provides instantaneous mapping of inputs to outputs, but does keep track of clock time. This limits its abilities as a tool for mapping to parameters such as the duration and onset times of a sound. The scheduler maintains an internal clock and acts as an interface between the parameter engine outputs and changes to the resulting synthesized sound. If parameters related to timing of sound events are included as matrix outputs, the scheduler then uses those parameter values to control the rhythm of sounds. The scheduler then has the ability to create a varying number of layered voices corresponding to one set of data. This introduces greater flexibility into the parameter engine's role, since the number of voices need not be predicted and set in the matrix before experimenting with a set of data. In addition, the scheduler's abstraction from the parameter engine interface allows the user to easily set controls for rhythm and timing in a more intuitive way, similar to the generation of a musical score.

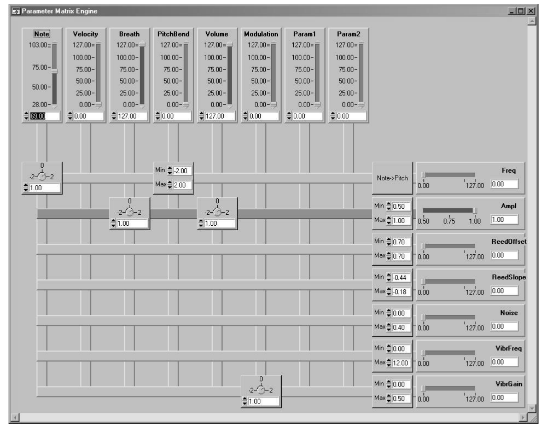

Figure 1: *Graphic interface of the parameter engine matrix.*

4. INITIAL EXPERIMENTS

To date we have experimented with oceanographic and financial data using the STK framework to interpret sonified data with a format filter instrument that approximates humanly produced vowel sounds. Oceanographic data from the Mediterranean used five data sets from five locations. The data included temperature and salinity measurements for depths of up to 100 meters, at 1-meter intervals, for each location. Temperature was mapped to filter bandwidth (thus getting colder is heard as a change from whisper to whistle) and the salinity to the center frequencies (higher sounds indicate higher salinity). The values are normalized such that the mean values correspond to a vowel sound (*ow* in this case). Twenty sample points at each location were successively sonified providing a correlation of depth and time. Sonified stock market data was achieved by representing a single trading day as a single sound burst. The closing price for the day is mapped to the center frequency, and the volume of trade to the bandwidth. These values are scaled such that the parameters for the last day of trade in each period corresponds to the reference vowel (/it A).

5. OBJECTIVES AND EXPECTED SIGNIFICANCE

The primary objective of the SonART initiative is to provide useful and accessible tools for a broad range of scientific applications. The project is seen as laying the foundation for an ongoing open-source collaborative effort. Formulating a framework and toolbox of sonification methods and providing a functional set of examples covering a variety of data types and analyses is of great potential significance for the scientific computing community and for scientific research in general. A prototypical portable real-time sonification device will be a significant addition to field research. It is hoped that establishing and maintaining a well documented and publicly accessible repository of sonification development tools will lead to further research and development in this promising domain of data mining. The long-term goals for real time acquisition and sonification on portable devices will assist field research with potential applications for medical analysis, geophysical surveying, defense and entertainment systems.

6. SUMMARY

In virtually every domain of scientific research data sets have grown dramatically both in size and complexity. Data analysis and data mining demands new tools for discovering patterns, trends and anomalies, SonART will provide intuitive and flexible development tools for enhancing data manipulation. We will present examples of data sonified using different methods, as well as demonstration of the interactive mapping allowed by the parameter engine.

7. REFERENCES

[1] P. Cook, and G. Scavone, Gary P. ”The Synthesis ToolKit (STK)”, in *Proceedings of the 1999 International Computer Music Conference*, Beijing, China, 1999, pp. 164–166.

[2] N. Friedlander, K. Schlueter, and M. Mantei, “A Bullseye Menu with Sound Feedback,” in *Proceedings of HCI International ’97*, San Francisco, CA, 1997, pp. 379-382.

[3] G. Kramer, “Some Organizing Principles for Representing Data With Sound,” in *Auditory Display: Sonification, Audification and Auditory Interfaces*, edited by G. Kramer, SFI Studies in the Sciences of Complexity, Proceedings Volume XVIII, Addison-Wesley Publishing Company, Reading, MA, U.S.A., 1994. p. 197-202.

[4] S. K. Lodha, J. Beahan, T. Heppe, A. Joseph and B. Zane-Ulman, “MUSE: A Musical Data Sonification Toolkit,” in *Proc. of the 1997 Int. Conf. on Auditory Display (ICAD97)*, Palo Alto, CA, 1996, pp. 1119-1134.

[5] C. M. Wilson, and S. K. Lodha, “Listen: A Data Sonification Toolkit,” in *Proc. of the 1996 Int. Conf. on Auditory Display (ICAD96)*, Palo Alto, CA, 1996, pp. 1119-1134.

[6] S. Barrass.”Personify: a toolkit for perceptually meaningful sonification,” in *Proceedings of the Australian Computer Music Conference ACMA ’95*, Melbourne, Australia, 1995.