Designing a Generalized Sonification Environment

Alberto de Campo [*], Christopher Frauenberger[†], Robert Höldrich[†]

[*] Academy of Media Arts Cologne (KHM)
adc@khm.de
[†] Institute of Electronic Music and Acoustics, Music University Graz
frauenberger@iem.at, hoeldrich@iem.at

Abstract

This paper proposes a design for a generalized sonification environment ("SonEnvir") that aims to be flexible, easily extensible, and practically useful for researchers from many scientific domains (target fields) that deal with data analysis and exploration.

SonEnvir is intended to allow for very quick development of new sonification designs, based on a library of existing sound processes that have multiple perceptually independent control parameters. By using modular software architecture which decouples components like basic data handling objects, data processing, sound synthesis processes, mappings used, playing approaches, and real-time interaction possibilities, all the individual aspects of one sonification design can easily be re-used as starting points for new designs.

Prototypes for this sonification environment are being developed on the SuperCollider3 (SC3) and PureData (pd) platforms (see SC3, pd).

1 Introduction

Sonification has been around for some 20 years now, and has been successfully used for many purposes: among others, Exploratory Data Analysis for a large number of scientific disciplines, real-time process monitoring systems, navigation systems, as well as straightforward data representation for the visually impaired, as documented at (ICAD). All of these uses have many commonalities, and in fact many implementations for specific domains/data sets use similar software techniques.

We believe that a generalized Sonification Environment (SonEnvir) that supports many of the approaches in common use, with the features described here, can be a valuable contribution to sonification research in a number of ways.

Probably the most challenging task in sonification is to design software that is useful for scientific researchers who feel that sonification could help them explore and analyze complex data, in order to understand inherent patterns and/or to classify the data by perceptual means.

In order to explore these unknown areas, the tools have to be well-understood and tested for reliability and validity of results. In order to achieve this, the obvious approach is to start with representing well-known data, verifying that their known content comes across clearly in user perception, and that the sonification process does not introduce any spurious information (which, in unknown data to be explored, would lead to misinterpretations).

In the course of tackling these tasks, we expect many findings that will be applicable for auditory display, real-time monitoring, design of auditory user interfaces, and other uses where known information is to be communicated by sound.

The design, implementation, and practical testing of such an environment in multiple iterations is the object of the main author's PhD thesis in progress.

This paper discusses the features we consider useful for this kind of environment, gives more detailed plans for each of the subtasks involved, and reports on progress with the implementation of prototypes.

2 Basic Problems

How can one best convey data content ('meaning') with non-speech sound?

Some concepts we use to evaluate the appropriateness and success of specific sonification approaches with are these:

- Similarity between data domain and sound domain,
- Conceptual or metaphorical proximity of individual data dimensions and sound parameters
- Levels of indirection between data and sound (pre-processing, reducing dimensionality, domain-based models)
- Representation vs. Exploration
- Sensory Richness vs. Immediate Intelligibility
- Perceptual Transparency of Sonification Processes

For very similar domains, no translation process is required, as can be seen in Auditory Seismology (Dombois 2002): Planet Earth is a resonating physical body, being excited with different kinds of energy input (various types of earthquakes, explosions), seismic recordings are physical movements recorded in time: Audification only needs to scale amplitudes and time steps to a suitable range for human hearing, and one can apply everyday auditory knowledge (how different energy impacts make different bodies resonate) to analyze the acoustic behavior of a scaled physical object – eventually down to very subtle details.

For time series data, the data time dimension is nearly identical to the time used in audio synthesis: In most cases, sonifications will map data time to sound time. There are cases where deciding against this can be heuristically interesting, and certainly navigation along the time axis is a valuable option.

For data with less immediate connections to audible qualities, i.e. less 'natural' mapping choices, **metaphors** have to be employed. E.g. a data dimension temperature could be mapped to pitch, overall activity, e.g. density of events, or overall brightness of a sound texture; all three seem to express the character or essence of temperature reasonably well, and the choice depends largely on context.

The more conceptual distance, the more arbitrary mappings between data and sound can and will seem; and the more short-term learning is necessary to understand what which property of the sound stream 'means'. The more dimensions are directly represented in the sound, the more complex the audible result will be, and the more difficult to analyze and understand.

One solution to this complexity is to introduce levels of indirection. This can be done by listening only to some selected data dimensions at a time, by pre-processing or filtering and thus collapsing several data channels into one, by using lower dimensional projections of the data space for sonification, or by introducing complex models (from the data domain or the sound domain) to mediate between data and sound synthesis.
All of these strategies serve the same fundamental purpose:

Reducing data complexity by means of pre-interpretation; making informed choices of what to represent in the sound and what to leave out, in order to arrive at sound results that are easier to understand.

We believe it is essential to make this tradeoff between complexity and immediate intelligibility of the audible results explicit, and to give sonification users fine control over this critical balance. In fact, the whole point of sonification as an exploratory strategy is to allow the sensory richness of the sound to transport latent patterns in the data that cannot be detected otherwise, so more complexity (or better, control of the degree of complexity) is desirable.

We believe that it is possible to carefully design synthesis processes that are sufficiently free from common problems like unwanted event fusion, masking of important detail, ambiguities between parallel streams, and misleading perceptual artifacts. Such synthesis processes maximize the degree of complexity that still can be understood by listening.

3 Feature Overview

Following Extreme Programming methodology (Beck 1999), here is a list of "stories" that name desirable features in a general-purpose sonification environment, and note importance.

A Sonification Environment should:

Read data files in various formats. The minimum is human-readable text files for small data sets, and binary data files for fast handling of large data sets.

Perform basic statistics on the data for user orientation. (e.g. for every data channel, minimum, maximum, average, standard deviation, simple histograms, etc.) This functionality should be user-extensible in a straightforward way.

Provide basic playback facilities like ordered iteration (in effect, a play button with a speed control), loop playback of user-chosen segments, zooming while playing, data-controlled playback timing, and 2D and 3D navigation along user-chosen data dimensions. Later on, navigation along data-derived dimensions such as lower-dimensional projections of the data space is also highly desirable.

Support multiple types of synthesis processes to be used for sonification. For a first iteration, simple audification, driving parameters in a continuously synthesized sound texture from data, or continuously generating new sound events for every data vector, as well as mixed approaches will suffice. Flexibility to use more advanced approaches later should be planned for.

Supply an initial collection of well-understood synthesis processes, along with test data to verify these for perceptual linearity and perceptual independence between control parameters. These test data can be re-used for evaluating and fine-tuning other candidate synthesis processes. Users should be able to add voices to the library, and write their own if desired.

Store sonification designs in human-readable text format: This allows platform independence, provides possibilities for informal rapid exchange, and can be an appropriate and useful publication format for sonification designs that employ user interaction.

Serve to build a library/database of high-quality sonification designs made in this environment, with real research data coming from a diverse range of scientific fields, developed in close collaboration with experts from these domains.

A large base of sonification designs written on the same basic platforms allows for generalizing commonalities into the main versions of the software, thus making approaches and ideas developed for a specific domain available to a larger group of users with little additional effort.

More generally, the implementation should be kept as lightweight, open, and flexible as possible to accommodate evolving new understanding of the design issues involved.

4 Practical Considerations

4.1 Data formats

The most general data format is a list of numerical vectors, with the order in the list often being another data dimension, e.g. order for time-series data. Any higher orders within the data can be represented as extra slots in each individual vector; e.g. a 3D spatial order of points can simply have slots for x, y, z.

In practical terms, data can be stored in human-readable text files such as SC3 code (rtf), plain text (e.g. Excel, SPSS, Matlab export formats, etc.), and .mtx format for pd; for larger data, binary data files in 32bit integer or floating point formats, and multichannel 32bit (floating point or integer) soundfiles for faster and more convenient handling.

4.2 Data Pre-processing

For convenience, all data channels are pre-analyzed for numerical range, linear and exponential averages, standard deviation, as well as user-customizable analysis functions.

Based on these operations, the SonEnvir can suggest mapping specifications, which are useful for first-pass data screening, and as a start for a sonification 'design/tuning session'.

To support more complex data preprocessing, the SonEnvir should provide for adding secondary data channels, derived by processing with external tools and importing results, with analysis functions provided in SonEnvir, and most importantly, with user-written functions, simply because the most meaningful data pre-processing will likely be domain-specific.

In the long-term perspective, more complex domain model worlds can be implemented to preprocess data in real-time in order to test domain theories, as well as to exploit analogies between data models and acoustic/synthesis models.

4.3 Playback Modes

For many applications, simple ordered iteration (i.e. a play button with a speed control) is appropriate and useful. For exploring greater detail, loop playback of user-chosen segments, and zooming while playing allows for user interaction, and generally contributes to the feeling of being able to "handle" the data. A further aspect to explore here is the idea of providing minimally intrusive tools for time orientation (e.g. using single sample clicks in a repeating accent pattern), and pitch orientation (e.g. a soft background drone, possibly akin to a tanpura resonating in octaves only).

Playback timing is a really interesting parameter to be data-controlled, and for making subtle timing variations audible, a well-designed continuous time reference will be useful. Mappings from data channels to time will typically involve pre-processing, typically difference to the previous data value.

More complex questions will be raised when polyphony is used for parallel data channels, e.g. how to make streams clearly identifiable while keeping them expressive of many data properties, and how to avoid unwanted event fusion or masking.

2D and 3D navigation along user-chosen data dimensions for both position and orientation will allow for more direct user interaction, and a more intuitive experience of the data space. We expect that navigation in data-derived dimensions such as lower-dimensional projections, while using rich multi-parametric synthesis methods, will prove to be really helpful.

Finally, gestural or haptic user input to a domain model analogy (as suggested in e.g. Hermann 2002), can be provided for. The idea of using actions like touching, knocking, scratching, or otherwise exciting real world physical objects that resonate, thus employing implicit everyday acoustic knowledge of users as perceptual data exploration tools, is indeed promising.

4.4 Synthesis Processes ('Voices')

Depending on the domain, many different kinds of acoustic behaviors can be considered useful to represent data content. Simple audification can be useful for ultra-fast data screening, so treating data channels as audio has to be supported. Generating one sound event per data vector is another very simple requirement, as is driving the parameters of one or more continuously running processes from data playback.

Obviously, there is a large repository of candidates for usable synthesis processes in the Computer Music world (Boulanger 2000, Roads 1996, Cook 2002, and many others]; in fact, there the inverse problem exists, namely, how to generate large masses of meaningful control data for complex sound generating processes. Especially for experimental software instruments, it is difficult to get to know their possibility space well enough, let alone find 'magic' zones of highly differentiated acoustic complexity that is really perceptible. Here, sonification can be a heuristic approach that is truly fruitful for both sides.

A particularly promising domain is Granular Synthesis, or more generally Microsound; e.g. (Roads 2002) documents a large variety of Particle Based Synthesis methods. For large data sets, or for first 'overviews' of smaller data sets within a few seconds, this world is ideally suited; and more detailed studies of the perceptual properties of these synthesis methods will make them more accessible and expressive for computer music purposes as well. Furthermore, models of real world granular processes can be applied to related science fields.

One of the perceptually strongest sound properties is spatial position, and thus current state of the art spatialisation techniques such as Higher Order Ambisonics, including the binaural approach (Zmölnig et al 2003, Noisternig et al 2003, Sontacchi 2001), which support scalable output formats from headphones to small speaker arrays to larger size venues, should be available to use in the synthesis processes. This technology has been researched and used

extensively at IEM Graz, and prototypes in pd are being used to explore these aspects of the SonEnvir design. With SC3 being open source, these custom pd objects can also be ported to SC3.

The literature for sound synthesis based on physical models is another promising source of ideas for synthesis processes, e.g. [9]. Common implicit knowledge of everyday physical sounds is quite subtle and differentiated, and approaches exploiting these perceptual capabilities will make sonification more accessible to people with little or no musical training.

4.5 Perceptual Linearity and Independence - Intonation

It is well known that sound parameters interact in human perception. The simplest example is the loudness differences humans hear between sine tones of equal intensity and different frequency, i.e. the Fletcher-Munson curves. For a sine wave synthesis voice, compensation based on simplified Fletcher-Munson curves is easy to implement, and compensation for e.g. short durations could also be found in the literature. This can be compared to having a craftsman set the intonation on a piano to ensure that it plays and sounds evenly through all registers.

Here is a synthesis voice in SC3, with compensation for pitch-amplitude dependence. For 200 to 4000 Hz, this is quite usable:

```
SynthDef(\sineTing2, { arg out = 0, freq = 440, amp = 0.2,
        attack = 0.01, decay = 0.1, pan = 0.0;
        var ampComp, env;
            // axis is middle C, every 3 octaves down by 6db.
        ampComp = (261/freq) ** 0.35;
        env = Env.perc(attack,decay, amp * ampComp);
        Out.ar(out,
                Pan2.ar( SinOsc.ar(freq), pan)
                * EnvGen.ar(env, doneAction: 2)
        );
}).load(s);
```

Obviously, for any more complex synthesis process, the best weightings for approximate perceptual linearization will be unclear, and thus it will be necessary to conduct listening tests that compare parameters for possible interdependence. By collecting enough user responses and “intonation” suggestions to allow for reasonably reliable averaged compensation factors, specially tuned compensations can be found for every single synthesis voice. There are obvious limits to the attainable precision (e.g. differences between individual listeners, and differences between audio systems). The best place for these compensations is in the synthesis voices themselves, such that each voice has known and tested recommended parameter ranges, within which good parameter independence and linearity can be expected, and that it is thus straightforward to use.

These test data can be made available as part of the software documentation, so users can test their own playback systems, as well as their own perception, and contribute to improving compensation fine-tuning over time. Furthermore, these test data will be good starting points for evaluating and fine-tuning other candidate synthesis processes.

This will facilitate including user-written synthesis voices into the general library of voices; without compromising reliability. The psychoacoustics literature focuses on real-world sounds and music, and explains perceptual phenomena like event fusion, stream segregation, spatial hearing etc. in great detail in natural sound contexts; how these phenomena apply to the immense variety of synthetic sounds is not at all clear. Thus, there will be plenty of interesting research work to do here.

4.6 Storage

Human-readable code/text as a storage format of sonification designs has many advantages: being executable code, it is very easy to exchange, e.g. for discussion in teams with dislocated partners. The code is also technical documentation of the exact processes used, which can be useful for platform independence, and longevity, facilitating re-implementations on later platforms. Furthermore, this is probably the only really practical publication format for sonification designs that require a significant amount of user interaction, which we believe will become more and more important, as can be seen at conferences like [14].

By being playable, with access to all the mapping and playback control parameters, these designs can even be directly used for suggesting alternatives and variations, making design tuning a potential part of the general discussion in the field.

4.7 Library of Sonification Designs

Once fully functional prototypes exist, one can start building up a library of high-quality sonification designs made using this environment. The ideal process here is having computer music/sound programming experts work directly with researchers from a wide range of scientific fields, who bring current research data they find good candidates for sonification experiments. We have started working with scientists from medicine, particle physics, nonlinear systems, and sociology as a representative cross-section. Pooling domain expertise in sound and target science fields makes this a truly interdisciplinary venture.

In the long run, a collection of strong sonification designs on the same platforms will allow for reuse of all components of the environment: Well-tested synthesis processes; playback, navigation, and interaction strategies; and data preprocessing/ model building approaches. In effect, generalizable commonalities between designs deemed successful by the respective target domain scientists can be adopted into the general software package, thus making successful heuristics easily accessible to other users. We hope that this approach can contribute significantly

to wider recognition of sonification as a research tool, and thus be beneficial to the sonification community at large.

4.8 Implementation

The platforms we have chosen for implementations of the design proposed above are both mature high-level audio programming environments: SuperCollider3 (SC3) [1] and PureData (pd) [2]. Both are open source projects supported by active developer communities and available for multiple platforms (Mac OS X, Linux, Windows), both have rich audio synthesis and processing libraries, and there is pre-existing work on both pd (Zmölnig 2003, Noisternig 2003, Sontacchi 2001) and SC3 (PulsarGenerator, CreatoVox, see Roads 2002, pp 154, pp 193-6).

Currently, the core set of the functionality described exists in a prototype in SuperCollider3, as well as some of the more advanced features proposed. This prototype is kept as small and flexible as possible; exploring next features to include is done in separate studies first, putting them into the main version only when they are understood and tested well enough.

We expect to make adjustments and extensions to the design described based on requests from and ideas developed with the target field users; particularly data pre-processing and domain modeling approaches depend essentially on input from target domain scientists.

Sonification designs from earlier work with sociological data have been ported to the prototype, which has proven painless, and the new incarnations have become more flexible. E.g., getting suggestions for matching scaling ranges of data dimensions and synthesis parameters is really helpful, and it is much easier now to start with a blank design and to iteratively add more and more complexity to the evolving design.

Currently, the control interface is code only, which is a very interesting approach to real time intervention while a sonification is sounding. By using Just In Time Coding style (Rohrhuber 2002f, Collins 2003, Just In Time Coding), interactions are usually very precise and can easily be kept to document the tuning process of a sonification design; and all aspects of the playing sonification are equally accessible and changeable form the code interface. While simple GUIs have been used for some purposes, building a general GUI that makes all the options available is impossible for an engine of this degree of generality.

Furthermore, a very simple and clear code interface may make the software easily accessible for visually impaired users.

5 Future Work

The immediate next step is to finish the basic versions in SC3 and pd, including a well tested library of sound processes.

Then, we can begin serious work with target field partners, using real data, arriving at sonification designs that have real value for the respective target domains, as well as being part of the described library of sonification designs.

Designs for including more advanced spatialisation will require test porting some of the advanced pd Ambisonic objects and turn them into SuperCollider UnitGenerators.

There is also plenty of literature research to do, particularly in psychoacoustics and perception (Bregman 1990, Snyder 200) as well as data analysis (e.g. Kantz 1997) to verify the hypotheses posed here.

6 Conclusions

We have presented a design for a generalized sonification environment, which is currently being test-implemented, and which we hope will be useful for both the sonification community and researchers interested in experimental analysis tools.

References

Beck, K, *"eXtreme Programming explained"*, Addison-Wesley, Boston, Mass., 1999.

Boulanger, R. (ed.), *"The Csound Book"*. MIT Press, 2000.

Cook, P.R., *"Real Sound Synthesis"*, A K Peters, Mass., 2002.

Dombois, F. (2002). *"Auditory Seismology – On Free Oscillations, Focal Mechanisms, Explosions and Synthetical Seismograms"*, ICAD Proceedings 2002, Kyoto, Japan.

Hermann, Th., *Sonification for Exploratory Data Analysis*. PhD Thesis, University Bielefeld, 2002.

ICAD. http://www.icad.org

PD. http://www.pure-data.org

Roads, C., *Computer Music Tutorial*, MIT Press, 1996

Roads, C. (2002), *Microsound*, MIT Press, 2002

SC3 (2002f). http://sourceforge.net/projects/SuperCollider

Zmölnig, J.M. et al, *"The IEM Cube"*, in ICAD Proceedings, Boston, USA, July 2003, pp. 127-130

Noisternig, M., et al, *"A 3D Realtime Rendering Engine for Binaural Sound"*, in ICAD Proceedings, Boston, USA, July 2003, pp. 107-110.

Sontacchi, A., Höldrich, R., *"Further Investigations on 3D Sound Fields Using Distance Coding"*, DAFX Proceedings, 2001.

Hermann, T., Hunt, A : www.interactive-sonification.org Conference Jan 2004, proceedings forthcoming.

Rohrhuber, J. (2002f) JITLib, class library published with SC3 download, available at (SC3).

Collins, N., et al., *"Live Coding Techniques for Laptop Performance"*, Forthcoming for Organised Sound 8:3

Just In Time Coding. http://swiki.hfbk-hamburg.de:8888/MusicTechnology/566

Bregman, A.S., *"Auditory Scene Analysis"*, MIT Press, 1990.

Snyder, B., *"Music and Memory"*, MIT Press, 2000.

Kantz, H., Schreiber, Th., *"Nonlinear Time Series Analysis"*, Cambridge University Press, 1997.