A Model for Data-Driven Sonification Using Soundscapes

KatieAnna E. Wolf
Princeton Univeristy
Princeton, NJ, USA
kewolf@princeton.edu

Genna Gliner
Princeton University
Princeton, NJ, USA
genna@princeton.edu

Rebecca Fiebrink
Goldsmiths University
London, UK
r.fiebrink@gold.ac.uk

ABSTRACT

A sonification is a rendering of audio in response to data, and is used in instances where visual representations of data are impossible, difficult, or unwanted. Designing sonifications often requires knowledge in multiple areas as well as an understanding of how the end users will use the system. This makes it an ideal candidate for end-user development where the user plays a role in the creation of the design. We present a model for sonification that utilizes user-specified examples and data to generate cross-domain mappings from data to sound. As a novel contribution we utilize soundscapes (acoustic scenes) for these user-selected examples to define a structure for the sonification. We demonstrate a proof of concept of our model using sound examples and discuss how we plan to build on this work in the future.

Author Keywords

Mutltimedia UIs; Soundscapes; HCI; End-User Development; Cross-Domain Mappings.

ACM Classification Keywords

H.5.5 Information Interfaces and Presentation: Sound and Music Computing; H.5.2 Information Interfaces and Presentation: User Interfaces

INTRODUCTION

The field of *end-user development* is defined by [12] as "a set of methods, techniques, and tools that allow users of software systems . . . to create, modify, or extend a software artifact." One framework for end-user development is *metadesign*, a design approach concerned with opening up solution spaces to users rather than complete solutions and includes a co-adaptive process between users and systems [4]. While metadesign allows users to contribute their own visions and objectives to a system, it also adds additional work for a user who may not have design experience. Automatically generating designs is one way to overcome a user's inexperience and incorporate expert knowledge (e.g., generating magazine covers [8], web designs [11], and user interfaces [3]). Additionally, allowing users to communicate their design goals through examples is helpful for users to express their design

objectives. For instance, the *Bricolage* algorithm retargets web designs from one web page to another based on an example web page specified by the user [11]. We propose a similar technique where a user specifies an example input to define the design objectives and to guide the generation of an output design. However, in our design space, users also specify data from a completely different domain that will be incorporated into the output design. This adds additional complexity as users may not be experts in both of these domains and may need additional help to create the cross-domain mappings.

The design space we explore is sonification design. Sonification is generally defined as the representation of data using sound. Many types of sound have been used including music, synthesized sounds, and recorded sound clips. Sonification is part of the larger category of auditory displays that have been developed to take advantage of the communicative power of sound and to overcome some of the shortcomings of visual interfaces [10], such as difficulty displaying high-dimensional data, lack of usability for users with certain physical impairments, or just the requirement that users stare at a screen. More specifically, sonification ". . . refers to the algorithm that is at work between the data, the user and the resulting sound [5]." Creating these algorithms is not an easy task as there are many aspects to consider: an individual's aesthetic preferences, multiple dimensions of sound, complexities of data (often only understood by experts), and principles for representing data with sound. Therefore, sonification design is commonly performed by sonification experts, rather than domain experts who will ultimately use the system [6].

A few sonification platforms allow users to specify mappings [16] and interact with the playback of the sonified data [14], but there is little work that allows end-users to closely interact with an intelligent system to automatically generate data-to-sound mappings. While work in [6] applies interactive optimization by allowing users to input perceptual quality ratings to generate "child" sonifications, we seek to allow users to play a more direct role in the creation of mappings.

In this work, we focus on a particular technique of sonification called Parameter Mapping Sonification where the algorithm at work is a mapping between data features and sound features. We are interested in exploring how end users who are experts in the data can play a role in the creation and generation of these mappings without the need for extensive knowledge about algorithms, programming, or sound. We present a novel, comprehensive model that allows for new approaches to end-user design for automatically generating cross-domain mappings that incorporate the use of sound-

scapes and theoretical knowledge of how best to convey data using sound. In this paper we first define soundscapes, describe why they are useful in sonification, and present our model for representing them. We then define our data model and our sonification model for mapping data to sound. As a proof of concept we apply our model to real-world data from Twitter and demonstrate the results with sound examples.

A main contribution of this work is to lay the foundation for users to create new sonifications simply by selecting an example soundscape and a data set. These models of data, soundscape, and sonification make it possible for the computer to constrain and guide user-generated sonifications, or even to automatically generate new sonifications using optimization.

SOUNDSCAPES

We use the term soundscape to identify an acoustic scene (e.g. sounds at a seashore or in a forest). Using sound recordings in Parameter Mapping Sonification is difficult as there are limitations on how the recording can be manipulated to encode data while still sounding realistic [9]. However, using recordings rather than synthesized sounds allows end users to work with terms they already know like *bird tweet* or *running water* rather than needing specific terminology like *frequency* and *timbre*. One work found that natural soundscapes had potential for use in sonification because of their ability to be easily distinguished from the background, while still being able to fade out of attention without being tiring or obtrusive [13]. The authors found that users listening to such sonifications even found the natural sounds to be "relaxing". Another work states that soundscapes can be "effective communication channels at the same time as being environmentally compatible and less fatiguing [15]." While Vickers et al. has yet to develop their sonification for network monitoring using soundscapes [15], Mauney and Walker used the triggering of sound samples to indicate that different thresholds had been reached in response to percentage changes from the average stock price [13]. Our process for mapping data to sound is more complex (allowing for user interaction and automatic mapping selection) and will be described later. We first outline our model for structuring soundscapes. By identifying and analyzing the structure that defines how individual sounds fit into an input soundscape $\mathcal{S}^{in}$ that is pre-segmented, we can determine how to recombine those segments to create a similar output soundscape, $\mathcal{S}^{out}$ that also encodes the data.

Soundscape Model

A *soundscape* is a sequence of *soundscape events* S_t where each event is a structure containing *sound groups* and *sound samples*. Each soundscape has a set of features that distinguish one soundscape from another (Table 1). A soundscape is then broken into a sequence of soundscape events $S_t \in \mathcal{S}$, where each event contains a finite set of all possible sound groups $G_j \in S_t$. Figure 1 demonstrates the structure of a forest soundscape with three different sound groups: a river flowing G_3, birds chirping G_2, and frogs croaking G_1. These sound groups also have a finite set of features that define whether the sound group is playing during that event and other properties about that particular sound group (Table 1). Each feature is defined as static or dynamic depending on

whether it may change from one soundscape event to the next. In Figure 1, the bird group, G_2, is playing during event S_1, so the on/off static feature is marked as on, but it is off in S_3.

Each sound group is a finite set where an element $g_{j,k} \in G_j$ is a short recording which we define as a sound sample. In the example in Figure 1 each distinct bird chirp recording would be a separate sound sample ($g_{2,1} \cdots g_{2,4}$). These different samples also have a finite set of features that contain information about that sample, such as the sample's start time and duration. Additionally, a sound sample contains features that define how the sound sample is played, which is important when we want to encode the information with the sound. For instance, the gain for the sample playback can be adjusted from 0 to 1, or the panning can be anywhere from left to right (-1 to 1). These features are optional, thus can also have a *Null* value if no adjustment is used. Table 1 displays some of the features and their possible values for the sound samples.

Soundscape Features	Value Range	Type
Number of Sound Groups	[0, max]	Static
Duration of Soundscape	[0, max]	Static
Sound Group Features	**Value Range**	**Type**
Type	Interval, Instant, Both	Static
Percentage of Soundscape Duration	[0, 100]	Static
Number of Samples in Group	[0, max]	Static
Group On/Off	0, 1	Dynamic
Sound Sample Features	**Value Range**	**Type**
Duration of Sample	[0, max]	Static
Start of Sample	[0, max]	Static
Sample On/Off	0, 1	Dynamic
Gain	Null, [0, 1]	Dynamic
Panning	Null, [-1, 1]	Dynamic

Table 1. Some of the features used to describe a soundscape, sound group and sound sample, along with their type and values ranges.
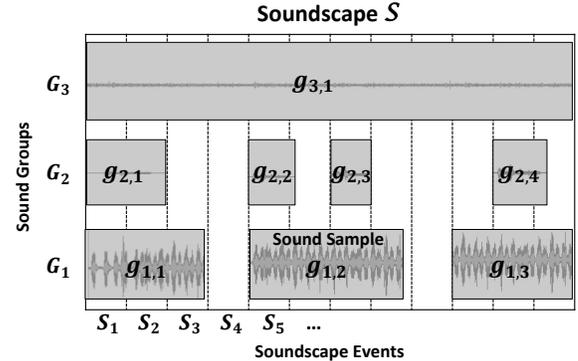

Figure 1. An example soundscape model with three sound groups $\mathcal{G}_j$, each with a different number of sound samples $g_{j,k}$.

Sound Group Type

The *type* of sound group is important in our sonification technique as it constrains the set of valid data-to-sound mappings. Sounds can be classified as either *interval*, *instant*, or *both*. Interval sound groups contain sound samples that have a temporal periodicity and can be looped (a river flowing). In contrast, instant sound groups contain samples that are best played in a single instant (a bird chirping). Some sounds can be classified as both, for example frogs croaking for a short time could be instant, but while looped could be interval.

SONIFICATION

Since a sonification is the transformation of data to sound, the data plays a key role. Here we define our model for the data and present how that data model combined with our soundscape model fit into our sonification model.

Data Model

Each dataset is made up of *data points* $d_{i,t}$ that have a finite set of features which are dependent on the domain of the data. A *data group* D_i is a collection of data points that are related by a specific feature(s). An example of this is a data set collected from different sensors over time. The data from a particular sensor i would be a data group D_i, and an individual reading from that sensor at a specific time t would be the data point $d_{i,t}$. The finite set of all data points that occur at a single time t can also be organized in a *data event* E_t. A visual representation of our data model is shown in Figure 2.

There are a few requirements for any data set that allow for an optimal "display" of the data using sound. One requirement is that the data must be ordered and have a feature that can be used as time. This is important in sonification to take advantage of a human's sensitivity to perceiving changes in sound over time [10]. Additionally, every data group must contain a *type* that states how the data will be "displayed". A *single* data group will represent each data point $d_{i,t} \in \mathcal{D}_i$ as its own discrete event, while an *aggregated* data group will aggregate an aspect of the features and continuously update that value.
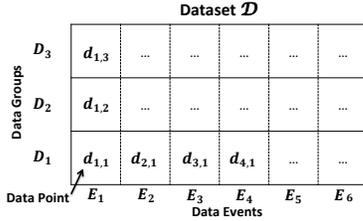

Figure 2. Our data model of data set $\mathcal{D}$ containing data points $d_{i,t}$ that each belong to a data group D_i and a data event E_t.

Sonification Model

This model is inspired by a model created by Hermann et al. [6]. Formally, we define the mapping $\Upsilon : [\mathcal{D}, \mathcal{S}^{in}] \mapsto \mathcal{S}^{out}$ that determines how the dataset $\mathcal{D}$ and input soundscape $\mathcal{S}^{in}$ will cause the creation of an output soundscape $\mathcal{S}^{out}$. The output soundscape is then defined by the sequence of all sonification events $\mathcal{S}^{out} = [S_1, S_2, S_3, \ldots S_n]$ over the timespan of the data. Each of these sonification events is a sum of acoustic events $\phi(d_{i,t}, \mathcal{S}^{in})$ that occur at time t:

$$S_t(\mathcal{D}) = \sum_{d_{i,t} \in E_t} \phi(d_{i,t}, \mathcal{S}^{in}). \qquad (1)$$

These acoustic events are based upon the input soundscape, the data, and the *policy*, ϕ.

Policy

The policy ϕ defines the "rules" that are used to create the data-to-sound mappings. We use the parallel structures of our data and soundscape models in order to create guidelines for specifying the mappings. This is supported by Kramer [9], who states "...important features will be more discernible if the display is structured to reflect structures in the data."

We restrict the policy by stating that data groups should be mapped to sound groups: $D_i \mapsto G_j$. In this way a data group will be represented by the distinct sound (timbre) of a sound group (also supported by Kramer who suggests mapping data families to sound streams [9]). Additionally, we use the *type* attribute we defined for both sound groups and data groups so that a data group of type *single* is mapped to an *instant* sound group, and an *aggregated* data group is mapped to an *interval* sound group. Any data group can be mapped to a sound group of type *both*. We based this decision on work done by Ahmad et al. [1], who proposed that instant-based temporal information (in our case 'instant' sounds) are used to specify a point in time, where as interval-based temporal information ('interval' sounds) are used to indicate status or progress.

Once the group mapping is created, we need to determine how each data point in that data group $d_{i,t} \in D_i$ will be represented by a sound sample in that sound group $g_{j,k} \in G_j$. Depending on the data, there are several ways in which this can be done. In the next section we present a simple rule-based policy for doing this that is motivated by principles developed in the auditory display community [9]. Our future work will examine other ways to build on this policy, e.g. using stochastic optimization via reinforcement learning.

MODEL APPLIED TO TWITTER DATA

Twitter data has been used for sonification in the past: *Tweetscapes* sonifies real-time Twitter messages from around Germany for radio broadcast [7], and *I Hear NY4D* presents a modular auditory display platform that uses Twitter data taken from different locations in New York City [2]. Neither of these applications allow for users to interactively modify the sonification to fit their personal data or interests. Since Twitter is one example domain where people's data and sonification preferences are likely to be diverse, we seek to develop an interactive sonification system where users can select the Twitter information they wish to monitor in real-time. However, our current version seeks to simply demonstrate our basic model, so we use a curated set of off-line Twitter data.

We define a Twitter data set in terms of our data model as follows. A data group D_i, or *tweet group* in this case, is a collection of tweets that the end user is interested in monitoring (for instance, tweets from a specific user or tweets that contain a specific hash-tag). Each *tweet* then represents a specific data point $d_{i,t}$. Some of the features of tweet groups and tweets are listed in Table 2. In this model, we determine the *type* feature of the data group based upon the number of tweets per second (TPS) of that data group and the shortest duration (SD) of all the sound samples. If $TPS \times SD < 1$ then the tweet group would be single, otherwise it would be aggregated.

End-user sonification design with a simple policy

In our current implementation, a user chooses a dataset of selected Twitter groups $\mathcal{D}$ and chooses a soundscape $\mathcal{S}^{in}$. The user is then prompted to choose a 1:1 mapping between each tweet group D_t and a sound group G_j restricted by the data group *type* and available sound groups that have a corresponding *type* as described in the *Policy* section above. Once all group mappings are determined, we use additional rules to

Tweet Group Features	Value Range
Type	Single, Aggregated
Total Number of Tweets	[0, inf]
Average Length of Tweets	[1, 144]
Maximum Length of Tweets	[1, 144]
Average Tweets per Second	[0, inf]
Number of Twitter users	[1, max]
Tweet Features	**Value Range**
Author of Tweet	" * "
Content of Tweet	" * "
Length of Tweet	[1, 144]
Time of Tweet	[0, max]

Table 2. Some of the features used to describe a Twittter group and tweet along with their values ranges.

generate the mapping between specific data points and sound samples. An example in the case of the single-instant mapping is a tweet-sample mapping where the username of the tweet is used to select the sample from the group. If there are more authors than sound samples, we can place the sample at different spatial locations around the user using the panning feature of the sound sample. For aggregated-interval mappings, we use the number of tweets per second to change the gain of the looped sample. The full description of the policy can be found at `http://research.gold.ac.uk/11275` along with the recording of the input soundscape and the generated output soundscape representing a collection of tweets taken from the 2014 Super Bowl.

FUTURE WORK

We have presented a novel model for sonification that uses the structure of a user-selected soundscape to drive the creation of mappings. We hope to expand on this model in the future to incorporate more intelligent mapping schemes as well as more user interaction. We plan to allow users to also generate the data-to-sound group mappings as well as iteratively refine their designs. Encouraged by work in [6], we hope to use interactive optimization as one way to balance the trade-offs between designing a sonification where: (1) the output soundscape is as close to the input as possible, (2) the data can be easily interpreted from the output soundscape, and (3) the output soundscape matches a user's aesthetic preferences.

ACKNOWLEDGMENTS

REFERENCES

1. Ahmad, A., Stanney, K., and Fouad, H. Theoretical foundations for integrating sound in interactive interfaces: Identifying temporal and spatial information conveyance principles. *Theoretical Issues in Ergonomics Science 10*, 2 (2009), 161–186.

2. Boren, B., Musick, M., Grossman, J., and Roginska, A. I hear NY4D: Hybrid acoustic and augmented auditory display for urban soundscapes. In *Proc. Int'l Conf. on Auditory Displays* (2014).

3. Gajos, K., and Weld, D. S. SUPPLE: Automatically generating user interfaces. In *Proc. IUI* (2004).

4. Giaccardi, E., and Fischer, G. Creativity and evolution: A metadesign perspective. *Digital Creativity 19*, 1 (2008), 19–32.

5. Hermann, T. Taxonomy and definition for sonification and auditory display. In *Proc. Int'l Conf. on Auditory Displays* (2008).

6. Hermann, T., Bunte, K., and Ritter, H. Relevance-based interactive optimization of sonification. In *Proc. Int'l Conf. on Auditory Displays* (2007), 461–467.

7. Hermann, T., Nehls, A. V., Eitel, F., Barri, T., and Gammel, M. Tweetscapes: Real-time sonification of twitter data streams for radio broadcasting. In *Proc. Int'l Conf. on Auditory Displays* (2012), 113–120.

8. Jahanian, A., Liu, J., Lin, Q., Tretter, D., O'Brien-Strain, E., Lee, S. C., Lyons, N., and Allebach, J. Recommendation system for automatic design of magazine covers. In *Proc. IUI* (2013).

9. Kramer, G. *Auditory Display: Sonification, Audification, and Auditory Interfaces*. Addison-Wesley, 1994, ch. Some Organizing Principles for Representing Data with Sound, 185–221.

10. Kramer, G., Walker, B., Bonebright, T., Cook, P., Flowers, J. H., Miner, N., and Neuhoff, J. Sonification report: Status of the field and research agenda. Prepared for the National Science Foundation by members of the Int'l Community for Auditory Display, 1999.

11. Kumar, R., Talton, J. O., Ahmad, S., and Klemmer, S. R. Bricolage: Example-based retargeting for web design. In *Proc. CHI* (2011).

12. Lieberman, H., Paternó, F., Klann, M., and Wulf, V. *End-User Development*, vol. 9 of *Human Computer Interaction Series*. Springer, 2006, ch. End-User Development: An Emerging Paradigm, 1–8.

13. Mauney, B. S., and Walker, B. N. Creating functional and livable soundscapes for peripheral monitoring of dynamic data. In *Proc. Int'l Conf. on Auditory Displays* (2004).

14. Pauletto, S., and Hunt, A. A toolkit for interactive sonification. In *Proc. Int'l Conf. on Auditory Displays* (2004).

15. Vickers, P., Laing, C., Debashi, M., and Fairfax, T. Sonification aesthetics and listening for network situational awareness. In *Proc. Conf. on Sonification of Health and Environmental Data* (2014).

16. Walker, B. N., and Cothran, J. T. Sonification sandbox: A graphical toolkit for auditory graphs. In *Proc. Int'l Conf. on Auditory Displays* (2003).