EXPLORING THE INTERFACE EFFECT IN DISTANT SONIFICATION

Iain Emsley

School of Media, Film and Music,
University of Sussex,
Falmer, BN1 9RH, United Kingdom
I.Emsley@Sussex.ac.uk

ABSTRACT

I introduce ongoing research into the method that I am calling distant sonification as a response to understanding abstractions created through computational reading. My aim is to explore the interface effect and situate it in sonification and media theory. Discussing existing prototypes, I contextualise the visible interfaces within the wider design models, such as patterns, and computational materiality. Reflecting on experiments in media specific analysis, I suggest that there are different models with their own specificities that are brought together to create the interface. They might exist separately or are combined to create a wider effect that I explore through models and grammars. I suggest that there are different models with their own specificities that are brought together by humans and machines through layers.

1. INTRODUCTION

Culture is becoming increasingly digital and requires new forms of practice and reading. Computational practices - such as distant reading [1], viewing or listening – use abstractions, such as maps and graphs, to analyse culture. Moretti's provocation that "distance… is a *condition of knowledge*: it allows you to focus on units that are much smaller or larger than the text" [7] suggests a strategy of not reading. Algorithms and data structures are central to this remediation of culture.

My research uses a method that I am calling distant sonification. It is a method of listening to these abstractions to aid interpretation and exploration. In particular, I want to develop Berry and Fagerjord's [14] view of critical Digital Humanities to explore the materiality of the medium's role in the sonification of cultural data. As the computational enables and remediate culture, I use sonification as a critically reflexive practice as well as representational method.

My research questions are:
1. How might the interface effect exist in sonification?
2. How might materiality affect sonification?
3. What role might sonification take in revealing its grammars and models?

In this paper, I discuss ongoing research into how the interface effect affects the sonification of cultural data. Using experimentation as a pragmatic approach, I advance a theory that the final abstraction is created from models that are remediated by other models, such as design considerations. I raise questions about what we might consider an interface and how it might be reflected as a construction.

Beginning through outlining the theoretical approach that I take; I reflect on the early experiments that explore distant sonification as a critical practice. From this, I begin reconsidering the design decisions being made and how these create an effect. I present experiments in using sonification as a tool for media specific analysis to raise questions about the hidden models. In the final section, I reflect on role of the computational in sonification and move towards the questions that this raises for future work.

2. A MEDIUM BASED APPROACH

In this section, I want to reflect on the role of the computational medium in sonification. My current focus is on Galloway's [3] interface effect, where it is transformation and transformative. Interfaces might be either software interfaces or the User Interface (UI), either graphical or aural. Building on design and listening approaches [19], my focus is on the materiality of the computational. I particularly want to explore the idea of the models and grammars that combine to create the effect and develop an argument that through understanding their role in the abstract and concrete models.

I explore the possibility that understanding the models supports the transformations. Grossman [4] suggests sonification is an extension of the human. Through considering the materiality, I want to use models to move from perception to consider how sonified cognitive models create and are part of an interface effect. Through considering this effect, I suggest that listening with *machines* becomes listening *with* machines.

I want to consider the models and grammar through their semantics and syntax. Vickers [2] raises questions about different design grammars and how they are interpreted by different actors within the development of the sonification. I use this to consider how these affect the transformation from data into sound within the digital medium. The models that both transform and are transformed into the abstraction and how the machine understands culture in its own milieu are constrained by the design considerations and grammars, themselves limited by syntax and semantics. Through this, I raise questions about how sonification might be used to understand these changes as a critical practice and as a reflexive critique. A critical question might be how might the object that is a sonification be de-reified? Instead of hearing it as an object, how might it be considered as an interface effect?

I show how layers of models combine and are altered by processes to create a façade. Taking a medium based approach, I argue that these layers can be interrogated to reveal hidden machine-based models created through remediation. I see interaction as a way of questioning the given options, leading to questioning the materiality of the medium.

3.   USER INTERFACES

In this section, I consider the developed prototypes to explore graphical interfaces - the dashboard and the live editor. Both use different interfaces on a common architecture, shown in Figure 1, and shared data sources, the Early English Books Online [17] and Russian Twitter troll data [16]. I consider the interface as an effect of its underlying mediations but also through external considerations, such as purpose.
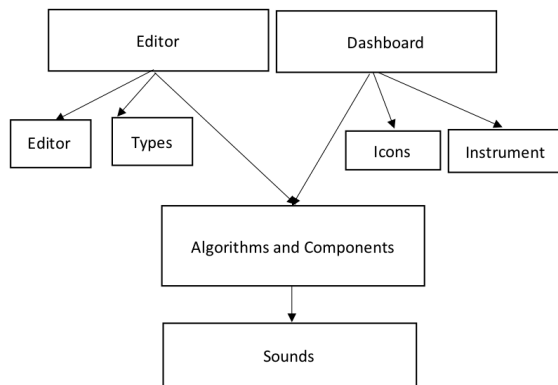

Figure 1. Diagram of the Distant Sonification Components Architecture

3.1.   Dashboard Interface

The first developed prototype is a dashboard, reflecting Manovich's cultural analytics [19] approach. The interface displays different facets to the underlying data, such as histograms and line graphs. Icons suggest the shape of the audio to appear to orient the listener. The initial theory is that the icons and components suggest limitations that are designed into the User Interface.

Each component provides a view for the data. These might be altered through widgets, such as filters, to identify particular elements of interest. The interaction that these filters support allow for user to alter the sonification for their own purposes. Two components focus on one particular hashtag which can be chosen through a filter. The first counts the main tag's appearances and those with an edit distance of two symbols between them. Processed into an abstraction of the term and its associated count, it is presented as a histogram. This sonification design suggests an additional interface that reflects design constraints. The second sonification takes the results from the same filter and it maps the initial position of the tag within a text string and maps this to a pan position based on the screen size. The first component is bound by accepted conventions of the histogram and its intentions, where the second uses the

text position in the data to allow the listener to make their own interpretation.

Through this, I might begin to think about how the interface effect is created. Hermann and Hunt [5] argue that interactive sonifications might be regarded as a type of virtual instrument that supports the interaction. Initial observations suggest that the dashboard allows the listener to create their own understanding through assigning a frequency to an event. Interaction allows a human model to work with the designed components, creating an apparent reflection of both worlds but with differing semantics, limited by a design context. These initial observations suggest critical angles to approaching and thinking with these models through the presented artefacts and their design.

3.2.   Live Editor Interface

The second prototype is an interface that supports the interaction with the processes within a live environment. Replacing the icons with the Ace editor [20], it can either show an empty editor or it can be given an algorithm from a library used behind the dashboard icons. When the mouse is moved away from the editor, it runs the code in the browser and stores the change onto the disk.

The interaction works with the revealed code and the editor's syntax model. By removing the icons, the editor and language become interfaces, reflecting Blackwell and Aaron [6]. The grammar of the library used potentially limits the interactions through its semantics. The underlying software is revealed but it requires the ability to read and reason with them. Unlike the dashboard, the code is able to be altered so embedding the changes at a deeper level, reinforcing the sense of the working with the machine. This reveals the library which might be considered not only as a way of making but itself is a model and is designed for a purpose. Showing the code requires the listener to understand the components created from the algorithms and the language that they are written in, such as JavaScript.

Replacing the icons with an editor alters the human and machine relationship. Where the icons project a set of options, the live editor provides a closer relationship to the machine. It not only requires different forms of reading but also an understanding of the language and library's own models and grammars as part of the generative process that also constrains. The editor seemingly removes an abstraction but demands a deeper understanding of the code and the machine, showing it to be an abstraction for a language through modes. Removing one model allows the coder to create a new one using the available grammars and to place their own context into the new code, provoking questions regarding Tanimoto's concept of liveness [11]. It raises questions about the effect of languages and designs that they impose on the environment and are imposed on them by semantics and syntax as a new abstraction.

The ongoing research will explore these issues through writing a constrained language for the existing sonifications and to consider the role of design.

4.   MEDIA SPECIFIC SONIFICATIONS

In the previous section, I discussed the existing prototypes and began uncovering the models and grammars within them. In this section, I want to think about sonification as

critical practice for understanding how the computational remediates itself to present different interfaces. This approach builds on Hayles's [8] consideration that print is flat and code is deep to explore how the medium's specificity and materiality can be sonified. I also want to think about it as a reflexive experimental practice.

4.1. Sonifying mark-up

An early experiment examines sonification as a critical practice to explore changing mark-up elements when a page is interacted with. The browser is an everyday manner of reading HTML and it also mediates the structural and style elements. The aim is to create a note each time that an event alters the HTML markup so that we can begin to explore what has taken place. These events may be human derived or from the web page. From this, we can begin to question how the change took place and what it might mean. Is a new component, such as a form loading, or are elements being removed, such as in infinite scrolling pages?

The probe is implemented as a web extension for the Firefox browser [12]. Using JavaScript's MutationObserver API, the extension listens for changes to the Document Object Model and maps these to a note. The sonification uses the Web Audio API so that a single JavaScript file can run in the browser. The sounds are microtones to suggest the speed of the changes. Changes to attributes, such as the accessibility attributes when a button is activated, are given a tone of 340.25Hz and detuned. The event also listens to the number of elements changed and whether these are being added or removed. This direction is used to calculate new frequencies from a base of 260.25Hz if added or 440.3Hz if removed, using the calculation for new notes to change frequency according to the number of changes.

As the browser code is event triggered, it needs to consider running in near real time. This places a constraint on the choice of timing mechanism. By allowing the event to drive the sonification, the machine is made more prominent and suggests that the sound is being made in near real time. The timing relies on synchronizing the various audio components for one event. This means that time is a model of sonification in itself, but it is a linking mechanism for other models.

The decision to use the AudioContext time allows the data to be used in near real time to demonstrate the amount and type of change that is taking place. The intention is to demonstrate the fast pace of largely invisible changes to suggest the addition of new components, such as forms, or changes to the markup to think about the materiality of the markup language. It does raise the question of what is being the sonified: the changed markup by the website or the browser?

The sonification is triggered by changes to the structural model. It shows that the machine part of the interface may change through intended, or otherwise, interaction in the browser, such as clicking a button or scrolling. The code relies on a language API and there is a map between the type of change and the note emitted. Although the data and transformation are both machine languages, a human mapping is required to create the relations with the endpoint. The sonification suggests that the interface might not just respond to the human action but also a machine driven one. Set into a browser extension, the sonification reveals the browser mediating a changing model.

4.2. Sonifying types

Having explored the interface as components, I turn to the syntax of the distant sonification abstraction. The second experiment explores the way that machine represent cultural data. I want to go beyond the algorithm to understand their effects. As data is being remediated in the initial components, it has to be represented in the algorithms.

The data is converted from textual strings into machine readable objects that can be manipulated. The approach taken echoes languages such as Sonnet [7] and Caitlin [9]. These languages show the operation of a running programme, the type language is aimed at helping the listener understand the way that the cultural form is altered by the algorithms. Yet it goes further than just the show the algorithm but how it represents the data for its operations.

Using reflection, an internal function library tests builds a tree of both time and types with the function that called and uses this to sonify the types, such as array, object, or float.

The library is a simple JavaScript file that use the Reflection API and Web Audio for sonification. Similar to the web extension, this library is designed to run within a coding environment. The live editor in section 3.2 is used to test and run the library, storing the models to allow me to read the underlying data later.

It shows the way that the abstract model presented as the interpretation is itself a mediated model. The second model is the human linking of time which is used here for the analysis. There are to options. Either the sound can be sequential, going through the objects as they appear in strict order of appearance or the nature of the objects can be used to simulate their interactions.

As an example, the filter component was sonified. This component takes a string for the URL to fetch data from and presents the received data as an array. As the array is iterated, it reveals the objects that are converted into numbers before being sonified. As the data is being iterated, it is being tested by the filter and only parts of it sonified. A further extension to this is to test for a named or anonymous function to reveal more about the design.

Through both experiments, we see that the materiality of the digital alters the sonification. In the first, we see the markup changing to reveal or hide components. It also reveals the way that websites alter their representation to support engagement. In the second, we can see the type and time models and the algorithms that alter these through processing. Using a filter to allow for the human model to suggest what should be sonified but revealing the types shows the changes and how the data structures changes through the processes. I want to think about these experiments as a critical practice, to not only test the theory but also how one might think about the practice of sonification.

5.   DISCUSSION

The computational object itself is part of the interface effect. The remediation of the data into the abstraction – the map or the graph - suggests that it is constructed of grammars and models that require critical reading.

Through such a reading, the dashboard becomes an assemblage of components, revealing a series of data processes. It is a combination that is reliant on the design

associations, such as the method of analysis or within the components shown. The component itself can be rethought as bringing an analytical model into being. The given interaction provides a way of using these models as experiments themselves in using parts of the model. In the editor, the interface alters from visual to one of code and the editor itself. The language given is an interface to the operating system and with the libraries. It is a more subtle one that relies on a deeper understanding of the computational and hints at the alteration of the human and machine relationship. Interaction suggests a controlled use of grammars to create a new model for sonification using the interfaces.

The reflexive use of sonification to analyse the medium begins a reading the abstraction as a construction. I use this reflection to build on Hogg and Vickers's [10] consideration that even pure data needs mapping. As the mark-up is transcoded into a sound model, the browser extension uses a simple mapping between the type of event and the sound. Although it has no transformation of state, the sound requires a form of link between an attribute in the data defined by a designer and a frequency. The focus on types shows the medium making its own changes and on what is being transformed from the original data to the sonification. The abstraction used is itself a model of types, times and data that is brought into being through both software processes and the design constraints. A media specific approach begins to reveal the digital grammars that are used to construct the abstraction. The computational object requires intervention to sonify it.

The mapping decisions have a role in the consideration of how the interface effect is created, from the type of interaction to the type of computational models that are created. I contend that these considerations supply the conditions for interpretation. By understanding the materiality of the computational reading, we can begin to understand and (re)create it in different ways.

Audio is also part of the interface, though perhaps under theorised in this context. In work providing access to artefacts for visitors with a visual impairment [15], a tablet was used to provide audio information in response to being touched. A sonification alerts users to the activated button and before the voice played. Although screens and paper interface exist, they become invisible through the haptic and audio process. They create an interface through remediating events, models and concepts into sound, reflecting audiation [21]. The aural responses raise questions about the emitted sound as a central concern for sonification.

6. CONCLUSION

I present distant sonification as a method to understand digital culture. Using existing prototypes, I contextualise the visible interface within the wider design models, such as patterns, and the materiality of the computational. The specificity shows the machine creating its own structures that are remediated. I suggest that there are different models with their own specificities that are brought together by humans and machines through layers.

7. REFERENCES

[1] F. Moretti, *Distant Reading,* London, UK: Verso, 2013.

[2] P. Vickers, "Ways of Listening and Modes of Being: Electroacoustic Auditory Display", *Journal of Sonic Studies*, vol. 2, 2012, arXiv:1311.5880

[3] A. Galloway, *The Interface Effect*, Cambridge, UK: Polity, 2012

[4] J. Grossman, "From Metaphor to Medium: Sonification as Extension of our Body" in *Proc. of International Conference of Auditory Displays (ICAD)*, Washington D.C., USA, 2010, pp 145-152.

[5] T. Hermann and A. Hunt, "The discipline of interactive sonification". In *Proc. of the International Workshop on Interactive Sonification*. Bielefeld, Germany. 2004.

[6] A.F. Blackwell and S. Aaron, "Craft practices of live coding language design" in *Proc. of First International Conference on Live Coding*, Leeds UK, 2015, pp 12-22.

[7] D.H. Jameson, "Sonnet: Audio-enhanced monitoring and debugging" in *Auditory Display*, G. Kramer, Ed. Vol. XVIII. Santa Fe Institute, Studies in the Sciences of Complexity Proceedings. Reading, MA: Addison-Wesley, 1994

[8] N. K. Hayles, "Print Is Flat, Code Is Deep: The Importance of Media-Specific Analysis". P*oetics Today 25*, 2004, pp 67–90. https://doi.org/10.1215/03335372-25-1-67

[9] P. Vickers and J.L. Alty, "CAITLIN: A musical problem auralisation tool to assist novice programmers with debugging" in *Proc. of International Conference of Auditory Displays (ICAD)*, Palo Alto, USA, 1996

[10] B. Hogg and P. Vickers, "Sonification Abstraite/ Sonification Concrète: An "Aesthetic Perspective Space" for Classifying Auditory Displays in the Ars Musica Domain" in *Proc. of International Conference of Auditory Displays*, London, UK, 2006

[11] S.L. Tanimoto, 2013, May. A perspective on the evolution of live programming. In *Proceedings of the 1st International Workshop on Live Programming* IEEE Press, pp. 31-34

[12] https://github.com/iaine/mutate

[13] F. Moretti. *Graphs, Trees, Maps: Abstract Models for Literary History,* London, UK: Verso, 2007

[14] D.M. Berry and A. Fagerjord. *Digital Humanities: Knowledge and Critique in a Digital Age*. Cambridge, UK: Polity, 2016

[15] I. Emsley, T. Graven, N. Bird, S. Griffiths, J. Suess and L. Shaw, "Please Touch the Art: Experiences in Developing for the Visually Impaired". *Journal of Open Research Software*, vol. 7 no. 1, p.4, 2019, http://doi.org/10.5334/jors.231

[16] https://github.com/fivethirtyeight/russian-troll-tweets/

[17] https://github.com/textcreationpartnership/Texts

[18] L. Manovich, "Cultural analytics: visualising cultural patterns in the era of "more media"". *Domus* 923 March 2009

[19] F. Grond and T. Hermann, "Interactive Sonification for Data Exploration: How listening modes and display purposes define design guidelines", *Organised Sound*, vol. 19 no. 1, 2014, pp.41-51.

[20] https://ace.c9.io/

[21] G. Kramer "Some Organizing Principles for Representing Data with Soundin *Auditory Display*, G. Kramer, Ed. Vol. XVIII. Santa Fe Institute, Studies in the Sciences of Complexity Proceedings. Reading, MA: Addison-Wesley, 1994