Interpretable sonification of HTML and CSS, an exploration.

Jeroen Oliemans
LIACS
Media Technology program, Leiden University
jeroenoliemans@gmail.com

ABSTRACT

Is it possible to make an interpretable sonification of the web? To answer this question the Musical Browser was build, The Musical Browser makes it possible to explore the web, and sent the HTML to the parser. The parser translates this model into data which a sound synthesizer understands. The synthesizer software turns this model into sound. This separation made it easy to experiment with the sonification.

Several tests were conducted with custom created, preselected and random webpages. These test provided a granular approach to interpretation. From webpages with a distinct property, (very dark webpage) to more average webpages (colorful webpage). For the last test predictions were made about webpages after hearing the sonification of these webpages.

The result is that the musical browser's sonification is interpretable for the HTML tags and some style properties of a webpage. But the sonification is not sufficient to make an exact prediction of the visual display of the webpage.

1. INTRODUCTION

While thinking about the enormous mass of HTML on the World Wide Web, the idea to use all these pages as an inspiration for art came to mind. I am a big fan of electronic music, this, combined with my knowledge as a front-end developer of webpages, inspired me to build a musical browser.

The musical browser is not expected to create fantastic melodies, but it is intriguing what the possible outcome of sonification might be and what the structure of the compositions from all the different websites will be like. One of the challenges is to find a way to parse the web. The web has been here for a while now and has seen many modification and enhancements which are still available on the web today.

Another question is; what are the limitations in sonification of a webpage? And, what are the elements that could be extracted from a webpage?

2. HTML, CSS AND SEMANTICS

What is a webpage, how is it built and why is it structured the way it is?

A website exists mainly of HTML, CSS files, and javascript for enhanced interaction and behaviors. Webpages can inherit multiple techniques such as flash and java applets. This research has the focus on HTML and CSS because these are the essential two languages to make a webpage.

Every webpage on the web needs HTML to display their content. HTML is the structure for the content of the page. HTML describes when to place a picture or a piece of text on the page. HTML consist of a document full of tags and these tags can contain content and inform the browser what type of content it can expect.

Tags are called elements and are formed by the element name enclosed into angular brackets. For example the paragraph element is formed as follow <p>some paragraph text</p>. Other tags require attributes to be rendered for example the image tag; .

The stylesheet (CSS) controls how the tag or the content of the tag is displayed. CSS can control color, placement, margins, and font-styling. CSS is developed to separate the content of a document from the presentation of the document. For example the following code sets the font type for a header in a webpage.

```
h1{ font-family: Arial, Helvetica, sans-serif; }
```

The semantic web [1] is the promise for the web. The use of semantics will increase in the future. Semantics make it possible to provide aid to human and machine for understanding the meaning of content.

HTML consists of structural, semantic and interactive tags [2]. A small part of HTML is entirely meant to support invisible functionalities of a web page. There are for example tags which assist the search engines finding their results <meta>, tags to display tabular date <table> and for web forms <input>. Semantic tags give their content special meaning, for example the <strong> tag provides emphasis to the content enclosed in the tags.

The list of available tags will change by future versions of HTML. Today the most used version is HTML4, but many browser producers have already begun implanting HTML5 which will be cleaner and uses more semantic tags.

3. RESEARCH QUESTIONS AND TOPICS

This research tries to find answers to the following questions:

1. Which elements or properties of a webpage are suitable for sonification?
    a. Why are these elements or properties chosen for sonification?
    b. In what way will the elements and properties be interpretable?
    c. Are there resemblances between the code and the sound? And if so, which resemblances?
2. To what extend is it possible to interpret the sonification on the properties chosen?
3. Is it possible to recognize subjective characteristics, like color and font styles, without learning?

It is also interesting to see if it is possible to understand the differences in the sonification after an introduction about the elements which are sonified. In a way that users can make a prediction how a webpage could sound.

The actual semantic value of the semantics in the HTML will not be interpreted. Because these values are highly abstract, for example the tag for paragraph (<p>) has a web semantic value for paragraph. This tag lets the world know that it contains “text”. To transfer the meaning “text” into sound is beyond the scope of this sonification project.

The translation of the visible part of the webpage will be researched marginally. The visible parts that are taken into measure are the average brightness, the temperature of the color (red - blue/green) and if the average font of the webpage is with serif of sans-serif.

There are many ways to change the visual display order of a webpage. For example; JavaScript tricks, large images and server-side scripting could alter the way the page displays to the user significant, while the order of HTML remains the same. Therefore the music will be based on HTML and not the visual display of the webpage.

4. MUSICAL BROWSER

The application is constrained to make it easier to compare different websites. For example, for each webpage only the HTML and the first stylesheet are collected. Some webpages serve multiple stylesheets but that would make the application unnecessarily complex. The application would also be slower, because all the additional HTTP requests and processing. The consequence is that only webpages with one relevant stylesheet can be compared.

The musical browser is an application consisting of three elements: a browser with a user interface built in Adobe Air [3], a parser build with Processing [4] and a musical interpreter built with max/MSP [5].

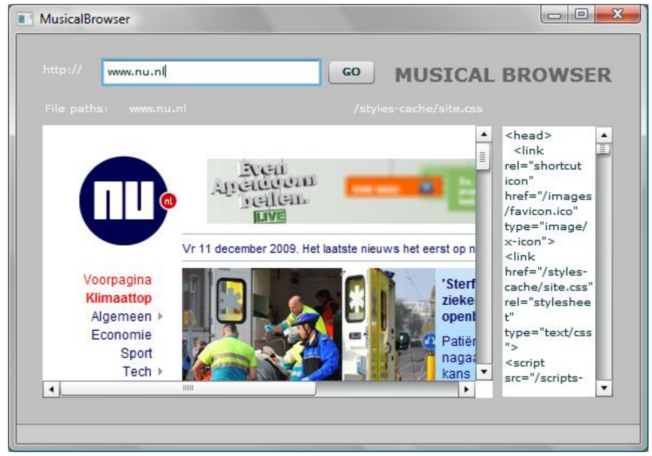

figure 1: musical browser interface

The browser makes it possible to interact with the web just like a normal browser reads the HTML and searches for the stylesheet link. This data is passed to the parser with an XMLSocket. The browser also displays the HTML in a text window next to make it possible for the user to follow along with the generated tag sounds.

The parser parses all the tags available in the HTML. The parser contains a matching list with all the available tags, each tag in the list is coupled with a pitch duration harmony and an amplitude value. All the tags in the webpage are compared to this list. This technique is known as parameter mapping sonification [6][7].

These parameters per tag are used in the Musical Interpreter to create a specific sound for each tag. After reading all the tags, the parser searches for the first stylesheet reference in the webpage, and retrieves the stylesheet. The parser sends the stylesheet and de HTML to the WebPageModel. The WebPageModel calculates an average of all the color values and searches for all the font styles and checks whether the style is “serif” (Times New Roman) or “sans-serif” (Helvetica).

The difference between serif and sans-serif is the most obvious separation in font-styles, these two types are always used in a stylesheet as a fallback mechanism. First you define the font-names you suppose to be available on the user’s computer, if the fonts are not available the computer chooses its own “serif” or “sans-serif” font. The created model is then sent to the MusicInterface. The MusicInterface translates the webpagemodel, including the tags, average color and the fontstyle into data which the MusicalInterpreter understands. This last separation makes it also possible to use multiple MusicInterfaces for other programs for example SuperCollider.

The data from the musical interface is packed into a list and send with OpenSoundControl (OSC) [8] to the Musical Interpreter, see section Sonification for details. See figure 2 for a flowchart of the parser.

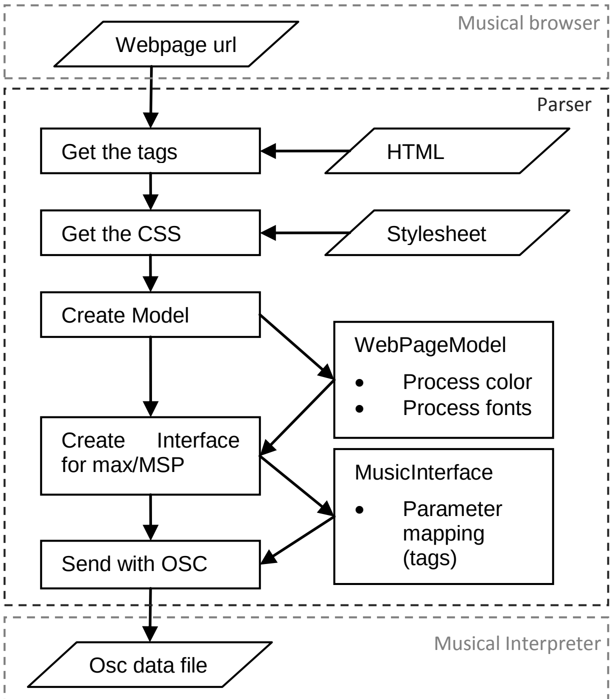

figure 2: parser flow chart

5. SONIFICATION

This research of the sonification of webpages tries to get to the heart of the HTML page by translating the basic elements of a webpage, the tags, the color, and the font-styling into sound. Therefore the content is left out, the content is not the thing that makes a webpage a webpage; content could also be revealed by paper or another medium. This research should make the diversity of the tags and the grouping clear. The sound should make the user aware of the structure and of the patterns used in the HTML of the website. In short, the user should hear the world below the visual representation of the webpage.

Frequency modulation synthesis (FM) [9] proved to be a good technique to transfer the tags. FM is very flexible and makes it possible to create interesting sound with a low performance cost. The tags play on after one another and form the “melody” of the sonification. All the available tags are divided into three groups, to make it clear that HTML provides three main groups of tags; structural, semantic and interactive tags. This makes it easier to test the resemblance between the HTML code and the sounds. A single FM-generator for all the tags is chosen to create unity between the tags, but still diverse enough to recognize types, groups and patters of tags.

Structure tags are the foundation of any webpage, like <body>, <div>, and <ul>. These tags have a pitch in the lower spectrum 200-500Hz and. The most important structural tags like <body> and <div> have a long duration.

Semantic tags give extra meaning to content, for example the <em> tags, stands for emphasis which gives the content more impact. The tags have an average pitch 700-1100 HZ. Their duration is shorter than the structure tags, semantic tags used for large content, like the <p> tag have a longer duration than tags used for specific content for example the <strong> tag.

The interactive tags are the <a> “link” tag, but there are a lot of others, forms are built with interactive tags like <input> for a button or <select> for a dropdown menu. Interactive tags are one of the main reasons that the web has grown so big. These tags have a high pitch 1300-1500Hz and a short duration. This way they can be easily separated from all the other sounds. For this project the sonification of the tags is not influenced by the stylesheet. Otherwise the sound of the tags would differ between websites making it impossible to learn to recognize the tags (and tag patterns).

To make it possible to interpret the font-style two sounds are synthesized for each style, the average font-style in the webpage determines which sound plays. For the serif fontstyle (with small horizontal strokes) a fragile, nervous sound is generated, because the font-style is more “nervous” than sans-serif. Every 3 seconds a high pitch sine wave with an alternating amplitude envelope. The sans-serif fonts are more clear and simple, if this style is more common on the webpage visited then a bass drum will sound every 3 seconds. Both sounds are easy to distinguish in the sonification.

A crossfade is implemented between a set of multiple saw-waves and a set of sine to make interpretations of the color values possible. These two are chosen because the saw polyphone sound gave me a dark and sinister feeling, the sine wave polyphone is more vivid, sparkling and light. Both sets provide a complex and layered soundscape, playing continuously because the whole webpage is influenced by color. The crossfade is controlled by the brightness of the total color. If the color is dark (low RGB values) then the saw waves will be more prominent. If the color is light the sine waves will be easier to hear. The tempo of these sounds will be influenced by the color as well. The sound application compares the average value of the green and blue color to the red color value. The highest value is processed further. The offset from the highest color value compared to the median 127 (255/2) is subtracted, in case the red had the highest value, or added (if green and blue have the highest value) to the base line values for the duration of the waves. The result is that a red color increases the tempo, and a blue/green color decreases the tempo of the polyphone sounds.

All the interpretable elements are listed in the table (table 1).

Interpretable elements/ properties	*Result*
HTML tags: Structural tags (<body>)	Low pitch, 200-500Hz
HTML tags: Semantic tags (<p> <em>)	medium pitch, 700-1100Hz
HTML tags: Interactive tags (<a> <input>)	high pitch, 1300-1500Hz
Font style: serif	Sine wave every 3s
Font style: sans-serif	Bass every 3s
Color: average element brightness is light	Polyphone wave is dominant
Color: average element brightness is dark	Polyphone saw is dominant
Color: average element tint towards red	Increasing tempo of polyphone
Color: average element tint towards green/ blue	decreasing tempo of polyphone

table 1: interpretable elements/ properties

The decision was made not to let the context information (font and color) influence the way the tags sound, to keep the tags interpretable. Otherwise it would be too difficult to learn to notice the different tags. This way the tag-sounds sonification resembles the HTML and the context sonification resembles the CSS just the way it works with a webpage.

6. TEST AND RESULTS

To verify the research questions, three tests were taken with the application. The first test sonifies predefined and controlled webpages. The second test is a field test on the web with preselected webpages with one obvious dominant property. The third test concerns random picked webpages without a dominant property. The last test concerned predictions of how the webpage would look after listening to the sonification.

First test: The sonification of the first test can be found here: http://www.wonderolie.nl/musicalbrowser/#tests

For the first test several webpages and stylesheets were made. The testpage includes almost all the available elements in HTML. The test isolates all the sonification aspects and makes it clear if the outcome sounds as expected.

The results of the first test were perfect. All the six pre-created webpages resulted in the expected outcome. For example the difference between the testpages with serif and sans-serif fonts is very clear in the sonification of these pages.

Second Test: The sonification of the second test can be found here: http://www.wonderolie.nl/musicalbrowser/#tests2

For the second test webpages are selected to isolate one of the sonification properties. For example a website with a lot of white elements or a webpage styled mostly with serif fonts. After these tests several random webpages were picked and a prediction was made what the sonification would sound like.

The second test proved that it is already more difficult for preselected webpages on the web with a dominant property to predict the outcome of the sonification. For example for the webpage of "the guardian" it is clearly to be heard that the webpage uses a lot of serif fonts. However the colorfulness of the webpage cannot be extracted from the sonification of this webpage. But if a website is analyzed by the user in advance than it is possible to make a good prediction, see for example the test of www.gothicfestival.be .

Third Test: The sonification of the third test can be found here: http://www.wonderolie.nl/musicalbrowser/#tests3

The third test sonifies some random selected websites, and for each website a prediction was made what the sonification would sound like.

It appears that the predictions became more accurate after a few predictions; this is probably caused by a learning effect. It does however indicate that the outcome is consistent and therefore predictable.

The fourth test concerns predictions the other way around. Is it possible to predict the visual layout of the HTML and the style properties, after listening to the sonification.

Fourth Test: The sonification of the fourth test can be found here: http://www.wonderolie.nl/musicalbrowser/#tests4

For this test the browser was positioned with the web window of the screen. The several sites were visited and recorded. Based on the sonification predictions were made. After the prediction the color values and the webpage itself were matched to the prediction.

It seemed possible to give a proper estimation about the average color of the elements, the position of clustered links and the font-style.

During the four tests the interpretation of the tags remains the same. The question was whether the sound resembles the HTML code, and if it is possible for someone familiar with HTML to detect patterns? It is possible to detect patterns that are in the HTML, mostly combinations with links and containers for example:

link `

The <li> tag is a structural element which has a low pitch and the <a> is an interactive tag which had a high pitch. These patterns are easy to detect. To detect all the other tags and other patterns is more difficult. Another aspect of the tags that is easy to detect is the beginning of a webpage, most often very structural, thus a lot of low pitches.

In the end it can be said that proper knowledge of HTML is essential to make anything out of the tag sounds. The user needs to be able to create a mental model of the webpage and the code structure to make it possible to detect the resemblances. Then the user has the ability to make a prediction about the sonification of the tags or the "melody". For example a blog listing has a lot of semantic tags, like headings (<h1>) and paragraphs (<p>). This melody is different than the homepage of a news website which mainly contains a lot of links (<a>) in structure tags ( ).

7. CONCLUSIONS AND DISCUSSION

In short the conclusion is that the musical browser's sonification outputs interpretable information about the HTML tags and some style properties of a webpage. But this information is not sufficient to form a proper prediction of how the webpage will look.

Sonification of elements and properties of webpages is possible. The musical browser makes it possible to extract some of the properties of a webpage by sound. It is possible to some extent to predict the outcome of the musical browser if someone is familiar with HTML. It is for example possible to hear where clusters of links are positioned within the webpage (group of high pitched tag sounds), or if the webpage has a complex layout (lots of low pitched sounds).

Without HTML knowledge it is more difficult, for example the color value. The problem is that most of the time dark elements go together with a light background, making the visual experience of the webpage light as well, which does not correspond with the "dark" saw sound. The solution might be to capture a screenshot to analyze the brightness of the pixels of the webpage.

The expectation of browsing several websites and generating different pieces of music did not came true. The sonification is less different from website to website than expected. This may be caused by the fact that almost all the websites sounds more or less the same because of ongoing standardization on the web. Another reason may be that websites are much alike without images and background images.

In the introduction I ask myself the question whether the data on the web could be used for inspiration for (traditional) arts. Unfortunately this is not the case with the current application.

How could the system be improved? To my opinion the current application needs to add the ability to make a sonification of the design. The code of websites may be more the equal than expected, the design still differences from each site. The current application will be better suited for HTML and CSS analysis than the visual appearance of the webpages.

The musical browser could be improved by implementing a system to calculate the impact of the properties. For example a webpage has 3 paragraphs, 1 very long, black paragraph, and 2 very short, yellow paragraphs; the current application says that the website is “light”. The user however perceives the webpage as “dark”. The font-style suffers from this same principle. This will make the sonification more accurate. Another addition could be bitmap analysis; analyzing the color of a screenshot of the webpage could increase the accuracy of the sonification of the website from a users’ perspective.

Another addition could be the sonification of the content for example the type of the content could be played by a type of instrument and the duration could be determined by the length of the content.

Is there a purpose for sonification of the web? Sonification of total webpages will not be that useful, nevertheless sonification of special webservices could prove very useful. For example it could be useful to make a sonification for weather messages. The sound could inform (trained) users very quickly. Or one could build a system which informs stock brokers for the several stocks they are interested in.

8. ACKNOWLEDGEMENTS

The author likes to thank Edwin van de Heide for project supervision, motivation and technical support.

9. REFERENCES

[1] Berners-Lee, Tim; Hendler, James; Lassila, Ora (17 May 2001). "The Semantic Web". *Scientific American*.

[2] W3C tags

[3] Adobe Air, http://www.adobe.com/products/air

[4] Processing, http://www.processing.org

[5] Max/MSP, http://www.cycling74.com

[6] Thomas Hermann; Helge Ritter, Sound and Meaning in Auditory Data Display, 2004, Proceedings of the IEEE special issue: engineering and music.

[7] C. Scaletti, “Sound synthesis algorithms for auditory data representations,”in *Auditory Display*, G. Kramer, Ed. 1994, Addison-Wesley.

[8] OpenSoundControl, http://opensoundcontrol.org

[9] Charles Dodge, Thomas A. Jerse , Computer Music - Synthesis, Composition, and Performance- 2nd edition, 1997