Implementing Accessibility in a Widely Distributed Web Based Visualization and Analysis Platform – Weave

Franck Kamayou, Heather Granz, Merve Tuccar, Shweta Purushe, Georges Grinstein

University of Massachusetts Lowell, Department of Computer Science

fkamayou@cs.uml.edu, hgranz@cs.uml.edu, merve_tuccar@student.uml.edu,

shweta_purushe@student.uml.edu, grinstein@cs.uml.edu

Mike Paciello, Graeme W. Coleman

The Paciello Group LLC

mpaciello@paciellogroup.com, gcoleman@paciellogroup.com

Abstract

Visualization tools provide authors with the ability to present large amounts of data in a way that allows the user to gain understanding of the data through a simple glance. This strategy, while useful to the sighted population, presents obvious barriers for blind or visually impaired individuals. Identifying a solution to this problem has become more vital, as ever more publicly funded agencies turn to data visualization as a tool for conveying information to the public. In this paper we present a solution based on previous research that allows a system to conduct automatic analysis of a line chart visualization to extract and then present it’s intended message. Previous advancements in this area, an implemented prototype of the proposed solution and a description of the platform in which it was built are presented, as well as a discussion of the implications of this research and future work.

Keywords

Weave, visualization, accessibility, blind, screen reader, disability, universal design

Introduction

Type the term "visualization" into a web-based search engine, and it is not surprising that more than fifty million results are returned. Clearly, visualization is a huge component of this golden age in multimedia. There are tremendous visualization tools available. These tools are colorful, aesthetically pleasing, even glamorous - but most importantly, they are (and should be) informative. Visualization tools give authors the ability to present huge amounts of data in a way that allows the user to gain an understanding with just a glance, and provides an appealing solution to a worsening problem known as *information overload*. Information visualizations are increasingly becoming adopted as a solution for large data dissemination. Though information visualization as a method for information dissemination is incredibly useful for individuals with the necessary visual faculties, this method presents obvious barriers to understanding for individuals who are blind or visually impaired. This problem, a lack of accessibility by this population to the presented information, is often ignored by organizations and website developers. Recent legal action by groups advocating for comprehensive website accessibility has many organizations, particularly those who receive public funding, scrambling to provide accessibility [1] that meets the WCAG 2.0 guidelines, which are currently the gold standard [3].

In this paper, we present a description of a novel user interface prototype, for the creation and manipulation of accessibility features directly within a visualization and analysis system. This work is implemented within Weave, which is an open-source web-based analytic and visualization platform. Weave is an ideal candidate for this project because it is open source, freely available, and is currently deployed and widely used by a diversity of organizations (many of which are publicly funded). The users of the sites of many of these organizations are potentially also individuals who currently do not have access to the information being provided using Weave [4]. The work described here is meant to be employed by website administrators as a way to advance a website towards realization of WCAG 2.0 guidelines.

The user interface provides the administrator with tools to both manipulate and access either system -or user- generated textual descriptions of visualizations within Weave. Access to these descriptions must be practical and cost effective. Users cannot be expected to spend any more time or money than a typical user would have to, to obtain the same information. The resulting information is therefore designed to be accessible by technology that is already widely adopted by the target population; screen reading software, particularly JAWS [13]. However, because of the nature of the implementation, this work is compatible with all types of screen reading software.

The user interface can be used to create visualizations and Weave overview descriptions using a manual method, an automatic method, or a combination of these. The basic case provides a novel method for allowing the user to create and adjust manual description components. The advanced case allows the inclusion of automatically generated description components. This system prototype is loosely based on the message category and recognition work done by Greenbacker et al [5][6]. Once a message is identified, it is made available via the systems graphical user interface as an option for inclusion in the final description. More detailed descriptions of these functionalities and the theory behind them follow in section 4. We will discuss how the chosen software platform extends the usefulness and functionality of related work, and creates new research possibilities in section 5.

Related Work

There are many approaches to providing alternative methods for accessing graphics, both practically and in recent research. On the Internet, current standards call for website designers to provide a short description via the *alt* attribute [17], but developers may not always provide this information. Manual creation of descriptions, although a great start, may be inaccurate and clearly does not scale. The automatic components we propose in this paper addressed these issues.

Researchers involved in the SIGHT project [2] have a well-developed system for line and bar charts providing high-level summaries of these graphs when they are encountered in documents, with additional details on demand. This research is ground-breaking, but is only suitable for static images, thus missing out on the potential to provide users with a better awareness of a visualization that has interaction or animation. It is also focused on providing a graph summary after the visualization has been created as an image, thus missing out on the potential benefits that our system may offer to a visualization designer, particularly one with a specific goal in mind. SIGHT is also implemented as a browser extension, thus requiring the user to download and install additional software, contrary to the system prototype presented in this paper. Further graph summarization methods have been researched, however these are either too granular [10] or too general [11] for practical use with a screen reader.

In the SAGE [14] system, an automatic caption generator for charts was implemented. The communicative goal of the graphics is gained by given data points. SAGE differs from SIGHT in that the communicative goal is deduced by reasoning about the communicative signals in the graphic [16].

The iSonic system, a sonification system for data exploration that was built for blind users, allows the user to explore geo-referenced statistical data [12]. This approach takes into account interaction but not animation, is limited to geographical data, and is more of a data exploration tool rather than a high-level summarization tool.

Implementation Platform

Weave

Weave is a visualization and analysis platform that is web-based and open source. It was developed by a team of researchers and developers at UMass Lowell in collaboration with the Open Indicators Consortium, a diverse team of public and non-profit organizations from across the United States. The size and diversity of the Open Indicators Consortium as a group of system stakeholders has resulted in a system that performs well for a multitude of different purposes. This has aided progress toward realization of the Weave mission: to enable anyone to visualize any available data for any purpose.

Progress toward this goal however has been directed at the 85% of the population that is not permanently disabled. This focus on the able-bodied potential user population is also the case for most software development projects [7], however by ignoring the 15% of the population that is permanently disabled; the Weave project not only loses out on a large potential user population, but also fails to achieve its mission.

Weave is built as a windowing system in much the same way as Microsoft Windows. In any instance of Weave, there can be as many or as few visualizations and data sets as the user wants or requires. Users can add, delete, resize, and minimize windows. Windows usually contain visualizations or visualization components but may also contain explanatory user-generated text or more complicated functionality such as statistical tools. As an open-source system with a diverse set of stakeholders and users, it is necessary for site administrators to have the ability to configure the level of complexity available to the end users.

Thus, Weave can be as simple or as complicated as necessary, depending on the use case which ranges from basic visualization for display/presentation purposes, with all or almost all complicated functionality disabled, to data exploration and analysis, with functions such as interactive probing, window configuration, data import/export/subsetting and statistical tools, in addition to much more available to the user. Weave instances can be viewed either as an independent web page, or they can be embedded into webpages as just one of many components.

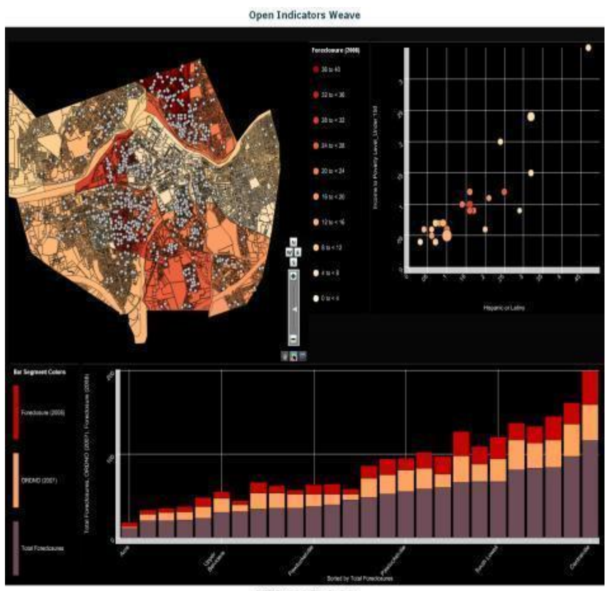

Fig.1. Weave visualization of Foreclosures in Lowell, MA

Weave's Users

Weave was brought to fruition with the support of the Open Indicators Consortium (OIC) [4]. The OIC funded and guided most of the development of the Weave platform by participating in an agile development process. Due to the diversity of individuals and organizations within the OIC, Weave has been developed to fit the needs of a broad set of users. As a result, Weave is suitable for users ranging from a random internet-browsing individual who happens to land on a web page that is running Weave, to an academic researcher using Weave to analyze complicated heterogeneous data. Three types of Weave users are defined with an accompanying description of how the user interface will uniquely benefit this type of user.

Use Case: Typical Internet User

The simplest case we present here is when an individual with average computer skills visits the website of a publicly funded entity, such as a city or town website. In this case, it is reasonable to assume that the user is visiting the site to retrieve information or to perform an action that they know this site provides. This person may encounter an instance of Weave embedded into the HTML of this page. Regardless of the complexity of each instance of Weave embedded within any website, the visualization(s) within it have likely been placed there to convey a message, probably a message pertaining to the relative data and mission of that organization. In this case, the system described in this paper will aid the user who is accessing this page with screen reading software that retrieves the intended message of each instance of Weave within that page - extending the reach of that website to include internet users with impaired vision. It also assists the user who does not have the necessary visual literacy to interpret a particular visualization or group of visualizations.

Use Case: Data Analyst

This use case is “advanced” in that the user is exploring a dataset to try to uncover meaning, and in doing so, she or he will probably make use of Weave’s more advanced statistical features and visualizations. Exploration of data sets (visual or otherwise) is a cognitively taxing and complex task. In fact, system design and methods that aid in the task of knowledge discovery and data analysis are a major research area and are being approached from many angles. The system described in this paper is uniquely beneficial to this type of user as well. As a visual analyst explores a data set, s/he is constantly changing and refining the visualization with which s/he is working. The automated component of the system described produces dynamically generated message components that adjust to reflect changes made to the visualization. This may support the data analyst by providing a textual interpretation of the visual scene.

Use Case: Website Author or Administrator

In this case, the user may be charged with the responsibility of creating or maintaining the website of a publicly funded entity such as that of a city or state government or a government agency such as the Department of Education. This user will be using Weave as a website component in order to tell a story or to provide a constituent with information, and this person may also find it absolutely crucial to ensure that the information or story being provided visually using Weave is also available in a more accessible form. This can be achieved by providing the information within the HTML of the webpage, however; design considerations may prohibit this. Here, the system being presented allows this user the freedom to make use of both manual and automatic components in order to provide the website viewer with a description or story that matches exactly with her or his goals. This description, once created, may have some, all, or none of its components automatically generated by the system and, and therefore will change dynamically. Once the description is satisfactory, the website administrator can access it as a string variable, using it to populate any text-based areas of a webpage, including regular text areas, popup text, or as the *alt* and *longdesc* attributes associated with that instance of Weave. Automatically generated components of that description will propagate through Weave, updating the variable, thus being reflected in the webpage wherever that variable is being used.

Discussion

Screen Reader Compatibility

Weave is a visualization system designed to run on the web. Most screen readers today are compatible with HTML and Flash content, and are able to read textual information from within them. However, screen readers are not capable of reading images, charts or other graphical elements by themselves. The web standards state that alternative text descriptions must be provided for those visuals via the *alt* attribute. Those attributes are usually non visible on a webpage and read by screen readers in place of the graphic they represent. Through the Weave Javascript API, Weave is able to communicate a *sessioned variable* back to its containing interface to provide the textual information for screen reader. A sessioned variable is simply any runtime variable within Weave that is to be stored in the session state. We created two variables of this sessioned type, one to represent a short description (basic description, explained in section 4.2) and one to represent a long compound description. A test webpage was set up to test the functionality of this framework, confirming that these string variables are indeed accessible outside of the instance of Weave itself from the webpage using common HTML elements. This test webpage was then accessed using the Jaws screen reading software as well as the Mac built in screen-reading software, VoiceOver [15], to confirm that screen reading software can indeed gain access to the text and properly utilize it. Additionally, we were able to dynamically link the session variables to the *alt* attribute content for a more streamlined description into the page.

Automatically Generated Descriptions

With any graph, there are often a number of levels of descriptions that are possible. Consider for example scatter plots. First is the overall trend which one can think of as the trend or regression line and its range, useful for a first pass overview. A user may want the spread, one or two standard deviations; one may want to see an interpolation curve, perhaps focusing on inflection points and varying spread. A user may want to select one or more points and have a description of these points, much like probing or brushing. Each requires a different set of messages, each with more detail and requiring more user interactions or specification.

In our implementations, we create mathematical algorithms for each description category. As stated above, trend messages (increasing, decreasing, steady etc.) are generated using regression lines. Once a best-fit model is obtained, we can further study the inflection points of the chart, using first and second derivatives.

Spreads and/or cluster information are identified using various clustering algorithms, and using various clustering metrics such as the Dunn Index. These metrics allow the system to determine whether a given cluster or data partitioning are perceptually relevant to a potential user. For line graphs, the same holds true: overview and range, spread, details on demand, and user selection and interaction response [18].

Thus we see a pattern of information to be conveyed for any visualization. This can also be extended to provide some perceptual interpretations. For example one could say that the slope of the regression line is 2 whereas the data's actual slope is 1 thereby implying that an exaggeration in the actual representation of the data has been made.

The Weave architecture is based on the concept of *session states*. All actions made within the system, including visual parameter settings, tool properties and user interactions are recorded and this "snapshot of history" is stored as a session state [9]. The session state contains all of the information required to restore a given Weave visualization, or instance, to the screen. To create a visualization that is accessible to visually impaired individuals, it can be used to generate a description of the visualization.

The session history feature of Weave allows the programmatic creation of specific descriptions of visualizations on the fly, even as they are being modified or animated, since the parameters in the session state change when the visualization changes. To do this we use the session history to automatically detect certain features of the visualization such as data change and columns replacement. This data information is then sent to the R statistical package using a plug-in to the Weave architecture, which then computes and sends back the necessary Weave statistical information that is used to determine and construct messages about that visualization. This process is inspired by and partially based on the work of Demir et al [2]. To date we have slope information computed statistically, minimums, maximums and standard deviations. Currently this process has been implemented for simple instances of line charts for the messages of increasing, decreasing, and stable trends as message categories defined in the work of Greenbacker et al [5]. An example of an automatic visualization description component produced by our prototype system is: “Scatter plot showing GDP by Date, The chart has an increasing trend.”

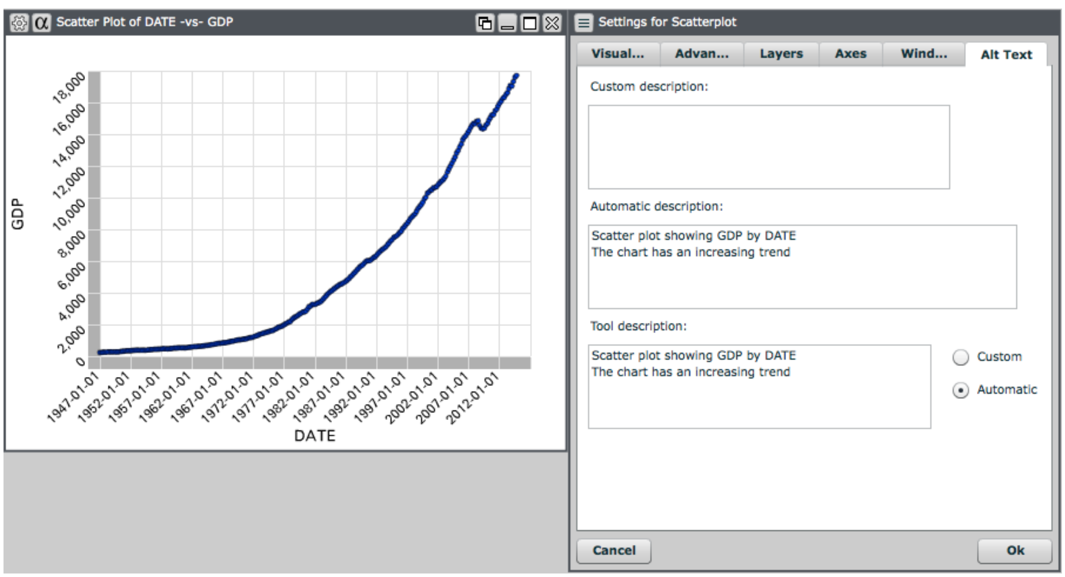

Fig. 2. A simple scatter plot showing GDP with system generated description, *Weave.*

User-Generated Descriptions

A basic graphical user interface tool was built to facilitate the creation of user generated textual descriptions. This functionality is provided both for the creation of Weave instance summarizations, within the main Weave preferences, and for the creation of individual visualization summarizations, within each individual tool's preferences. This GUI allows the user to manually type in descriptions of visualizations or the entire instance of Weave that is accessible from outside of Weave.

A Weave *instance summarization*, or an overview summarization, is considered in this interface to be a basic description and is referred to as such. This description is expected to have the least amount of detail as it may describe a complicated Weave instance with multiple visualizations and is restricted in length to fifty or fewer characters. An example of a basic description is: "Three visualizations showing Obesity prevalence data in the US."

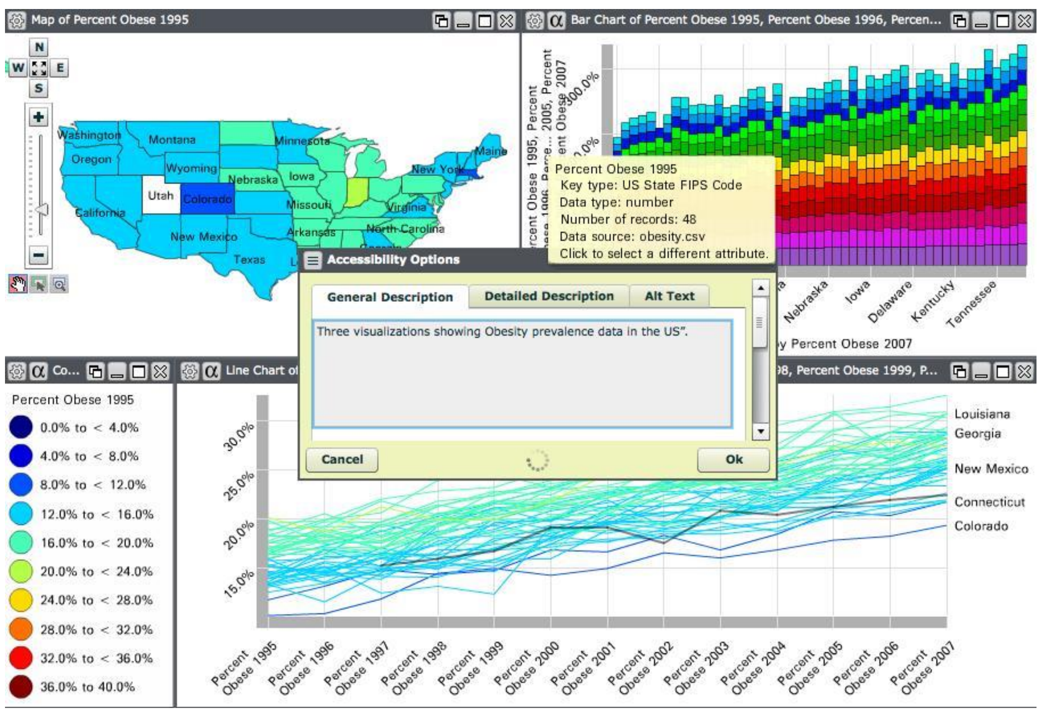

Fig. 3. General Weave Descripton: Obesity Prevalence in United States – *Weave.*

An individual visualization (or tool) description is referred to as a *manual visualization* long description component. A manual visualization long description component may be more detailed than a basic Weave description because it is only representative of, at most, one visualization and thus is expected to summarize potentially fewer details. A manual visualization long description component is also restricted to fifty or fewer characters. An example of a manual description is: "Map showing the obesity prevalence in 1995 in Massachusetts." This type of manual description may be desirable for the interjection of the author's domain knowledge, to emphasize specifics, or human readability.

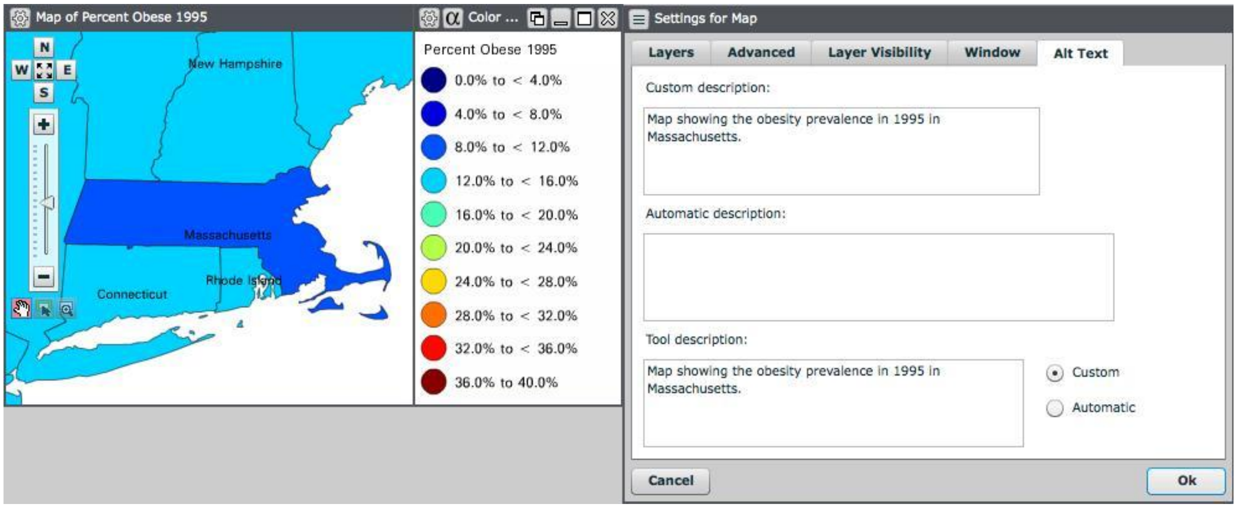

Fig. 4. Custom tool Description: Obesity Prevalence in Massachusetts – *Weave.*

Compound Extended Descriptions

Long descriptions are a user-determined combination of the basic description, manual visualization long description components, and automatic visualization long description components. For each component that is defined using the interface, a control is dynamically created within the long description (aka compound description) interface. These controls are in the form of radio buttons that only appear in the interface once that particular description component is defined by the user or generated by the system. These are designed to allow manipulation of the final long description text by toggling the individual description components into or out of the long description.

The order of the components in the long description is determined by the order in which they are added or subtracted using the GUI. The end result is a series of sentences appended together, which may or may not contain the overview sentence or any of the individual tool description (manual or automatic) sentences, according to the user's choice. An example of a long description that has a basic description, a manual visualization tool description, and an automatic visualization tool description included may be: "Three visualizations showing BMI data in United States. Line chart showing skyrocketing obesity. This line chart has an overall increasing trend from 1995 to 2010."

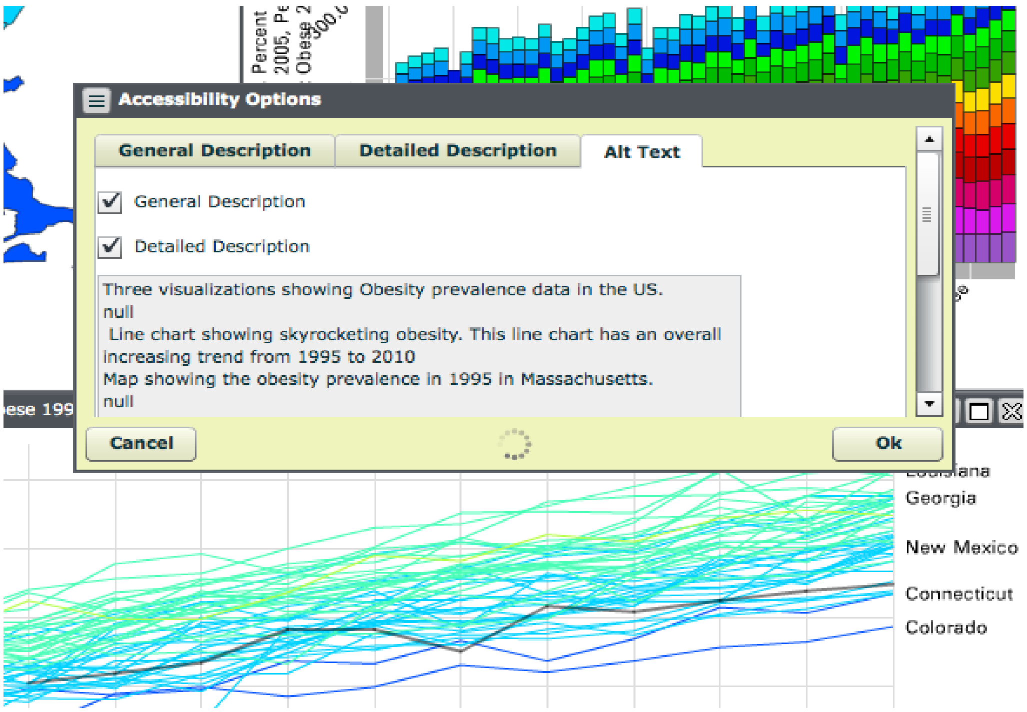

Fig. 5. Compound Extended Description: Obesity Prevalence in United States - *Weave*

In addition to the GUI, the long description is also manually editable. This allows the user to adjust the flow of the long description once the necessary elements have been edited. This allows complete control over the final long description without forcing the user to re-type or remember each of the components. An example of how this may be used would be to edit the example long description sentence above to read "Three visualizations showing Obesity data in the US. Line chart showing skyrocketing obesity, an overall increasing trend from 1995 to 2010.

Alternative approaches – sonification

The research presented in this paper thus far has primarily considered the use of text-based equivalents of graphical information. However, one of the disadvantages of using text alone is that, while it can make the content *technically* accessible, the results can place significant cognitive demands on the user, particularly when describing extremely complex charts and graphs containing a large amount of data that updates dynamically over a short period of time. Indeed, we recognise that providing a literal, detailed, text equivalent may even have a **negative** effect on accessibility, as users may have to trawl through vast amounts of data to identify particular values or to gain an understanding of detailed trends, which may be time consuming and possibly even likely to be unsuccessful. Hence, in order to approach parity with visual users by moving away from mere descriptive feedback towards fostering exploration of the data, additional approaches may be required.

One possible approach is the concept of *sonification* [20] in which the underlying data is presented through non-speech audio rather than visually or through text. An example is the Sonification Sandbox hosted at the Sonification Lab at the Georgia Institute of Technology [21]. Using sonification, data are mapped to multiple auditory parameters (such as pitch and timbre/tone), thus essentially providing an auditory "sweep" of the underlying information. For example, the shape of a chart can be mirrored through a dynamically changing pitch (i.e. low pitch tones for low values and high pitch tones for higher values), thus providing the user with an overview of where there are peaks and troughs, and hence allowing the user to very quickly identify where there are outliers or any other oddities which may be difficult to convey in text format – particularly if the data is regularly updated at a rapid rate.

However, while a significant amount of research has been carried out investigating the potential for sonification [22], its application to the Web in the wild is arguably still highly experimental at best, and too time consuming and/or expensive to develop at worst. This is compounded by the fact that, while there are some very helpful resources to assist developers to provide appropriate equivalents for graphs and charts [23], such resources often only consider "flat", reasonably simple, and predominately static images which convey only a small amount of data that can be easily provided in alternative formats such as a simple text-based description or a table. On the other hand, resources which provide practical and pragmatic advice on sonifying highly complex, dynamic, graphs and charts to foster exploration rather than description are fewer and far between. Furthermore, there is no "true" standard for creating graphs and charts, making it technically very difficult to apply a single approach to sonifying data across different environments; for example, graphs may be created as scalable vector graphics (SVGs), using tools such as Flash, or even presented as a series of animated static images in JPG or PNG format.

Yet, as graphical information on the Web becomes more complex, there is an argument that moving beyond mere text-based equivalent descriptions is becoming increasingly vital to ensure that all users are able to understand and benefit from the underlying information.

Conclusion

In section two of this paper we presented a comprehensive review of screen reading software, sonification, and graph summarization, and explained how these works differed from and contributed to the work presented in this paper. Section three described the software platform in which this work was implemented, presenting three different use cases and explaining how the work presented in this paper is useful in each of those three cases. In section four we described the implementation and theory behind this work and gave examples of the output produced by each part. In section five, we argue that additional methods, in particular sonification, may be required when content becomes increasingly difficult (not to mention demanding on the user) to describe in text format.

The work presented in this paper describes one of the first automatic visualization to text graphical user interface implemented within a visualization and analytic system. It was designed specifically to work with the tools available and already being used by visually impaired individuals, specifically screen-reading software.

However, there are several areas to explore as future work. First, the work will be extended for advanced line charts, which need more message categories for description. This work provides the user with the ability to both manually and automatically adjust a description of visualization, but the implications of this hybridization of content contribution have not yet been studied. Since this system automatically generates message text, it may also have interesting implications and applications in report generation. Because the Weave software platform was built using a session state based architecture, this work has access to both the visualization and analysis parameters in addition to the raw data. In the future, this may allow us to adjust our methods to compute messages about both the visualization and the data, possibly allowing the viewer to make comparisons between the two and to gain insight into the truthfulness and accuracy of the visualization being presented. [19]. We also intend to study the effect that the availability of a dynamically changing high-level message has on a visualization designer, by presenting designers with a basic goal and observing the design process to see if it is affected by an accurately changing message.

Works Cited

"Accessibility." *Apple*, http://www.apple.com/accessibility/voiceover/

Baumann, A., The design and implementation of Weave: A Session State Driven, Web-Based Visualization Framework. Doctoral Dissertation, University of Massachusetts Lowell, 2011, 116 pages, AAT 3459174

Bjork, E: Many Become Losers When the Universal Design Perspective is Neglected: Exploring the True Cost of Ignoring Universal Design Principles, Technology and Disability, IOS Press 2013, vol 21, no. 4, pp 117-125 (2009)

Burns, R., Carberry, S., Elzer, S., Chester, D.: Automatically Recognizing Intended Messages in Grouped Bar Charts. In: 7th International Conference, Diagrams 2012, pp 8-22, Canterbury, UK, July 2-6, (2012)

Demir, S., Oliver, D., Schwartz, E., Elzer, S., Carberry, S., McCoy, K. F.: Interactive SIGHT into information graphics. In: *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A),* Article 16, ACM, New York, NY, USA (2010)

Frank, J.: Web Accessibility for the Blind: Corporate Social Responsibility or Litigation Avoidance? In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, pp. 284, 7-10 (2008)

Ferres, L., Verkhogliad, P., Lindgaard, G., Boucher, L., Chretien, A., Lachance, M.: Improving accessibility to statistical graphs: the iGraph-Lite system. In: Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS, pp. 67–74 (2007)

Greenbacker, C. F., Carberry, S., McCoy, K. F.: A Corpus of Human Written Summaries of Line Graphs. In: Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop, pp. 23–27, ACM, Edinburgh, Scotland, UK (2011)

Greenbacker, C. F., Wu, P., Carberry, S., McCoy, K. F., Elzer, S., McDonald, D. D., Chester, D. Demir, S.: Improving the Accessibility of Line Graphs in Multimodal Documents. In: Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies, pp 52-62, Edinburgh, Scotland, UK (2011)

Grinstein, G., Tuccar, M., Kamayou, F., Luo, Y.: Measuring Truth in Visualization (2013) [Unpublished Data]

Hermann, T., Hunt, A. and Neuhoff, J.G. (Eds): The Sonification Handbook. Berlin, Germany: COST/Logos Verlag GmbH (2011). W3C: HTML5: Techniques for providing useful text alternatives. http://www.w3.org/TR/html-alt-techniques/

"JAWS Screen Reading Software by Freedom Scientific." *JAWS Screen Reading Software by Freedom Scientific*., http://www.freedomscientific.com/products/fs/jaws-product-page.asp

Kramer, G. (Ed): Auditory Display: Sonification, Audification, and Auditory Interfaces. Reading, MA: Addison-Wesley (1994).

Kurze, M.: Giving blind people access to graphics (example: Business graphics). In: Proceedings of the Software-Ergonomie '95 Workshop on Nicht-visuelle graphische Benutzungsoberflachen (Non-visual Graphical User Interfaces), Darmstadt, Germany (1995)

Mittal, V.O., Carenini, G., Moore, J.D., Roth, S.: Describing complex charts in natural language: A caption generation system. Computational Linguistics 24(3), 431–467 (September 1998)

Open Indicators Consortium Information, http://www.oicweave.org

Shneiderman B.: The Visual Information-Seeking Mantra, Journal of Information Visualization (1996)

Sonification Sandbox: http://sonify.psych.gatech.edu/research/sonification_sandbox/

W3C, G94: Providing short text alternative for no-text content that serves the same purpose and presents the same information as the non-text content, http://www.w3.org/TR/WCAG20-TECHS/G94.html

Weave Javascript API Documentation, http://ivpr.github.com/WeaveBinaries/asdoc/weave/core/ExternalSessionStateInterface.html

Web Accessibility Initiative. Web Accessibility Guidelines WCAG 2.0, http://www.w3.org/WAI/

Zhao, H., Plaisant, C., Sniederman, B.: iSonic: Interactive Sonification for Non-visual Data Exploration. In: Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility, pp. 194-195 (2005)