Tools for Data Stories

Jonathan Harper
UC Berkeley Computer Science
CS 294 – Information Visualization
jharper@eecs.berkeley.edu

ABSTRACT

This paper describes a set of tools developed to explore the space of direct manipulation tools for annotation of web-based interactive graphics which have become increasingly common in journalism and media. Examples of interactive graphics using the D3 library were collected from the web, and a Google Chrome extension was developed to allow end users without programming knowledge to link their own text or other text on a page with interaction events in these visualizations. The core contributions are the interface for direct manipulation and the approach of injecting scripts to monitor activities in and extract data from the underlying data visualization. An additional tool was developed to allow users to extract the data bound to D3 elements using a similar direct manipulation approach.

Author Keywords

Information visualization; Storytelling; Authoring tools

ACM Classification Keywords

Information Visualization: Miscellaneous

INTRODUCTION

One of the core functions of data visualization is to convey information to others. Graphics may emphasize certain aspects of the data in order to convince the viewer relationships within an article or presentation. Interactive visualization is being distributed to broader audiences than ever via news sources such as the New York Times which take advantage of the engagement interaction provides. Segel et. al. [7] note that so called ”data stories” or ”narrative visualizations” can differ in important ways (primarily interactivity) than traditional storytelling.

Though interactive narrative visualizations have become more common, only experts are able to create them using a visualization library like D3 [3]. Many interactive visualizations also exist without an explicit narrative. The goal of this project is to explore tools for linking narratives with, adding narratives to and extracting data from existing interactive graphics.

RELATED WORK

Gershon and Page [5] discuss the effectiveness of storytelling and how the goal of utility in conveying ideas mirrors similar goals in information visualization. Some data visualization platforms such as GeoTime Stories [4], sense.us [6] and Tableau public [2] have integrated stories and annotation. These systems have made it easier than ever to integrate stories and visualization, but don’t yet address certain parts of the design space of narrative visualization nor annotation and storytelling with existing visualizations.

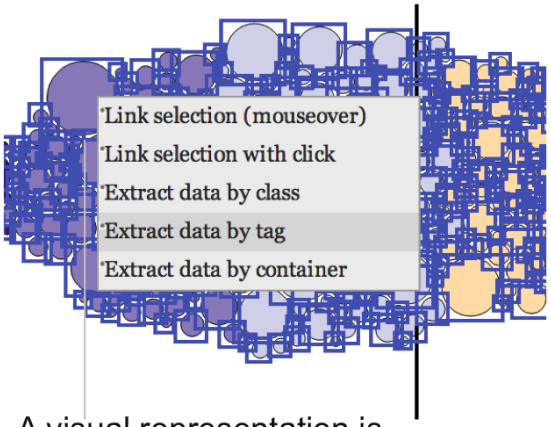
Segel and Heer [7] explore the design space of narrative visualization, classifying them into several high level styles. They note that the ”magazine style” visualizations, in which graphics are placed alongside or within narrative text, were the most common style identified but rarely utilized interaction techniques. This work investigates the challenges behind integrating interaction into these ”magazine style” data stories.

These ”magazine style” interactive narratives are referred to by Victor [9] as ”explorable explanations” or reactive documents. The Tangle library [8] was developed as a tool for creating such reactive documents. However, Tangle requires access to and understanding of the document’s underlying Javascript code. This project seeks to explore options for achieving some amount of interactiong without the user having to change the underlying implementation.

IMPLEMENTATION

A set of examples of narrative visualizations were gathered from the web and inspected for use of interaction and narrative. Visualizations created using the D3 visualization library were chosen because of its use within the visualization community as well as its increased adoption by the journalism community.

As a first step, a prototype of a system to automatically link text with data graphics based on text matches with the dataset was developed. This relied on references to the graphic within the text exactly matching those in the underlying dataset. This linking process required the data model of the visualization to be made explicit by the developer and for code to be changed in the visualization’s implementation. This prototyping process revealed issues with automated linking: narratives rarely explicitly mention data elements from the graphic and the



A visual representation is presented of the user's selection for data extraction.

user doing the linking needed to understand the underlying implementation.

Next, a tool was developed to allow end users to manually link text on a webpage with an interactive visualization without having to change any Javascript code. This was challenging because the DOM doesn't provide an interface for retrieving interaction handlers and browser plugins discourage access to Javascript objects within a page. Event handlers were retrieved by using knowledge about common patterns in libraries like D3 and jQuery to do a heuristic search for registered functions. Then, if arguments are required by the event handler, common values are attempted such as the data bound to the element of interest.

This is inherently an imperfect matching process which could be alleviated by the DOM API including access to registered event handlers, and typifies standards issues that make developing tools for interacting with web-based visualizations difficult. Since there wasn't a standard method for solving issues like this one a significant amount of development time was spent attempting to correctly retrieve event listeners. A simpler approach was eventually developed in which the event generated by the user when selecting the element to link is intercepted and used to send future events to the target element through the DOM.

A script to extract handlers and data is injected into the page of interest at runtime by a Google Chrome plugin to avoid issues with browser sandboxing. The injected script was instrumented to allow direct extraction of data bound to DOM elements by class or tag name using D3 and export it in JSON format. Data may be extracted from elements selected by HTML tag, the container element, or the element's classes. To make the system's interpretation of the user's selection clear, all elements selected for data extraction are surrounded with a bounding box if selected by tag or the background color of the container is changed for container-based extraction.

DISCUSSION

This project contributes a direct interface for linking text (or other) elements on the web to interactive visualizations without access to the underlying implementation. While we demonstrate a proof of concept, another goal is to raise questions about what kind of annotation interfaces are effective for interactive visualizations. We believe that it is possible to learn more about how people interpret visualizations using these kinds of annotation tools as well as afford new types of interaction with data graphics.

FUTURE WORK

This project is early work towards a larger goal of making it easier to annotate existing visualizations on the web, create interactive "magazine style" narrative visualizations and extract and reuse data from web-based visualizations. Immediate future work involves expanding the tool's compatibility with more visualizations and visualization systems. Allowing manipulation of more complex types of interaction by recording events generated by the user could allow the system to reproduce their effect. For example, this could allow a user to associate scrolling through their story with menu selections or animations in an interactive visualization.

Another area of further work involves investigating the potential for direct manipulation of data graphics. Making it simpler to directly manipulate SVG graphics and generate animations could open up a broader range of applications for generating interactive visualizations from existing static visualizations directly. Data extraction could also be used to allow restyling so that users can produce new views to compare and contrast how the data is presented by a story.

This project's timespan and scope didn't involve user testing. Future work could involve a more significant user study to find out how users learn from visualizations and how they use annotation tools when they are provided. This kind of study could provide insight into what kinds of tools would be most interesting and useful to users.

Finally, tying annotations of interactive visualization to social media or emerging open annotation platforms such as hypothesis [1] would allow easier sharing of anecdotes or ideas about the visualizations. However, futher work would need to be done building tools to support this and understanding which tools and interfaces are most appropriate for the environment social media and collaborative annotation provides.

ACKNOWLEDGMENTS

Thanks to Maneesh Agrawala for teaching the course and providing honest guidance and feedback during the project and to Steve Rubin for printing my poster.

REFERENCES

1. Hypothesis. `http://hypothes.is/`, 2013.

2. Tableau public. `http://www.tableausoftware.com/public`, 2013.

3. Bostock, M., Ogievetsky, V., and Heer, J. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2301–2309.

4. Eccles, R., Kapler, T., Harper, R., and Wright, W. Stories in geotime. *Information Visualization 7*, 1 (2008), 3–17.

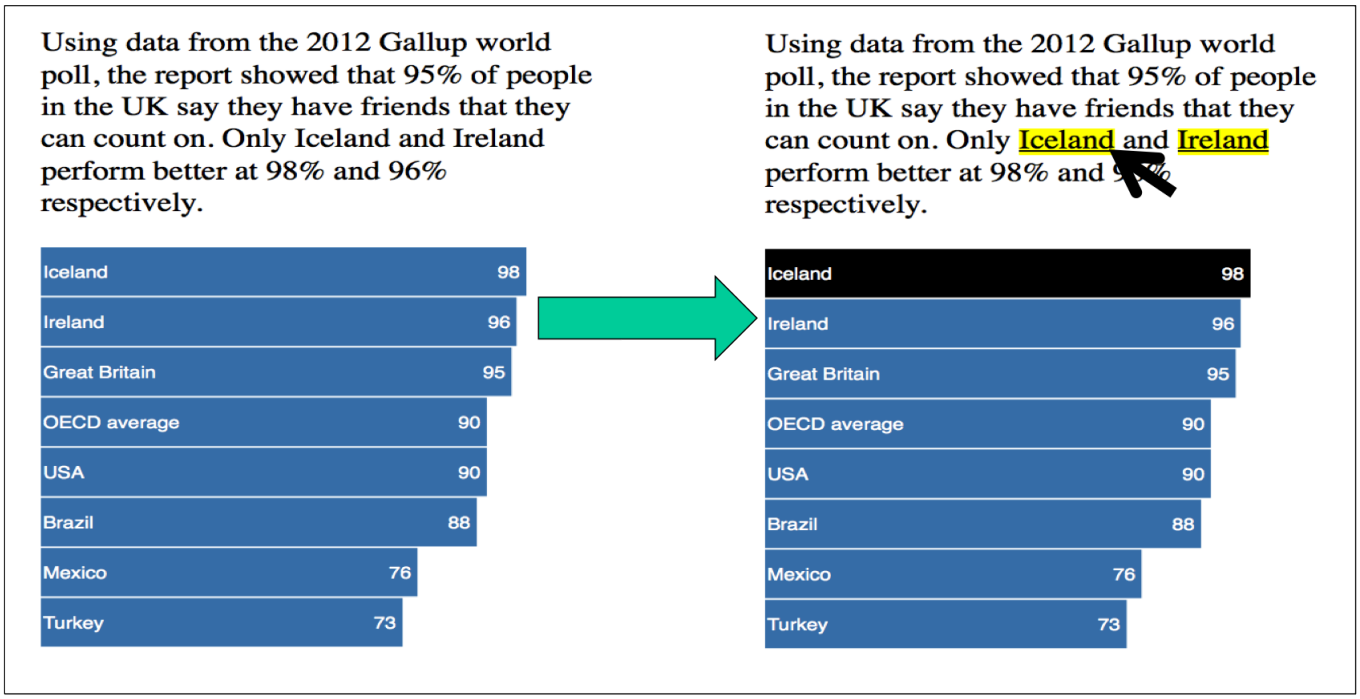
5. Gershon, N., and Page, W. What storytelling can do for information visualization. *Communications of the ACM 44*, 8 (2001), 31–37.

6. Heer, J., Viégas, F. B., and Wattenberg, M. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2007), 1029–1038.

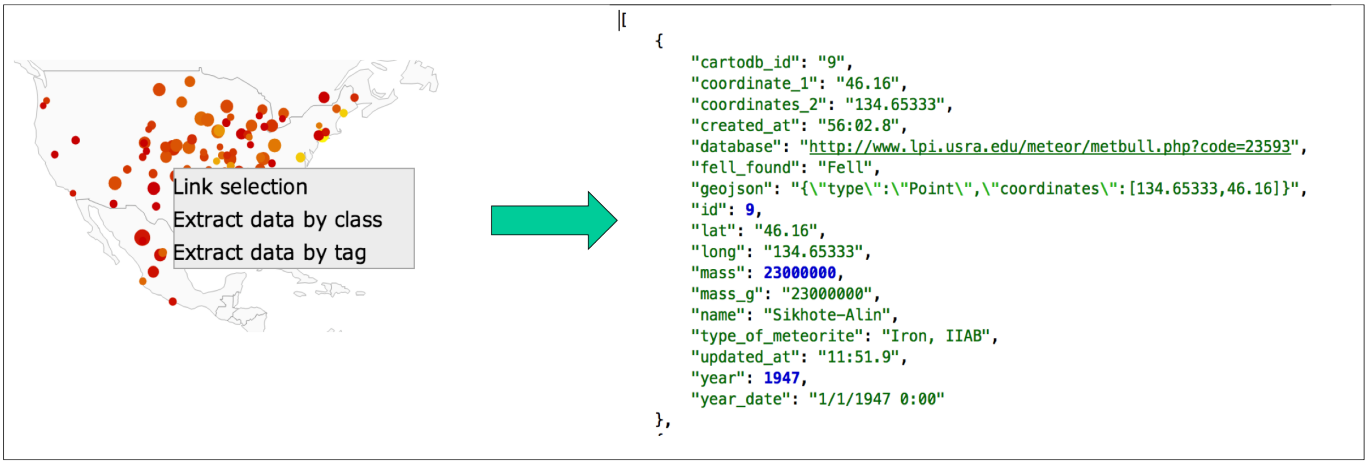
7. Segel, E., and Heer, J. Narrative visualization: Telling stories with data. *Visualization and Computer Graphics, IEEE Transactions on 16*, 6 (2010), 1139–1148.

8. Victor, B. Tangle. **`http://worrydream.com/Tangle/`**, 2010.

9. Victor, B. Explorable explanations. **`http://worrydream.com/ExplorableExplanations`**, 2011.



Hovering over elements in the text highlights related elements in the visualization.



An end user can choose to extract data from elements of a certain class or HTML tag. If appropriate data is found, it it stored in JSON format for further analysis.