ProtoDocs

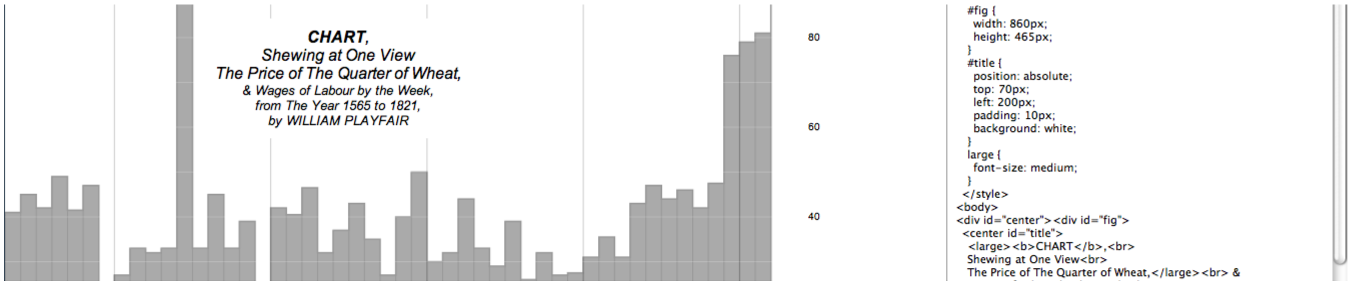


Figure 1: The ProtoDocs Interface

1 Abstract

In this paper we present ProtoDocs, a system for collaborative collection of data and producing information visualizations. Information visualization, being a relatively new field, has a fairly significant number of tools available, but many are often difficult to use and hard to learn. We argue that a simplified interface is needed--one with which the common computer user is familiar with, but also powerful enough for a user to be able to customize the visualizations as they deem necessary. We present various mechanisms for creating visualizations and discuss the advantages and disadvantages of our system within the information visualization community.

2 Keywords

Information visualization, collaboration, data analysis, user interfaces

3 ACM Classification Keywords

H.5.2. UI, H.5.3 Group and Organization Interfaces

4 Introduction

Information visualization plays an important role in understanding data, particularly large quantities of data which can be difficult to navigate through. The power of a good visualization lies in its ability to help solve problems and to make sense out of data that is collected. More importantly, good visualizations tell stories and to give the viewer an awareness of what's really in the data. In the information age, information visualization plays an increasingly greater role as more and more data is collected about the world we live in. There are two main aspects in the process of producing information visualizations: the first being that of creating the visualization itself, and the other being that of obtaining the data.

Figure 2a: The two aspects of producing a visualization

The process of data collection is one which often is tedious and time consuming and usually relegated to assistants. Although it is a fairly basic task and also a very menial one, its role is actually very important and significant in the process of the information visualization because there is a need for accuracy and quality in the data. Inaccurate values can lead to a misrepresentation of the data being presented. Additionally, there is a lack of collaboration in the effort to collect data, nor a means for backtracking the data collection efforts.

In creating information visualizations, there are a number of challenges. The first consideration of the visualization design is which variables to quantify. Another challenge is how to visualize those variables: color, thickness, spatial arrangement, etc. And yet another challenge is the actual production of the visualization, which is what we aim to address through our project. There exists a growing number of information visualization software and toolkits that are available for users, but they are often cumbersome to use and are not meant for the average user.

In summary, we are looking at three major issues that are related to information visualization, which are the following:

- Data collection - we aim to ameliorate the task imposed on those who perform data collection activities

- Collaboration - the process of producing an information visualization is not meant to be an individual endeavor, and collaboration in the process provides a number of benefits on both the side that is involved in creating the visualization as well as those who are viewing the visualization

- Creating information visualizations - these tasks are often designated for technically proficient individuals who have the expertise to develop visualizations, which usually means that it is not feasible for the common user to produce information visualizations

From a UI perspective, spreadsheets have been found to be very useful for interacting with numerical data (Chi, Barry, et al). Spreadsheets are often used in common office tasks, such as with Microsoft Excel. This domain of data analysis is one in which many computer users should be familiar with and it is also a very important area for us to consider. A 2008 study at the University of Hawaii of over 113 spreadsheets showed that 88% of them contained errors (Panko 2008), and that consultants estimate that roughly 20% to 40% of spreadsheets contain errors (Panko 2005). As a result, collaboration is a critical aspect to deal with regard to data collection. Because spreadsheets are a common format for handling data and there is an important need to deal with spreadsheets, we focus our project specifically on spreadsheets and collaboration.

On the information visualization end, while basic spreadsheet software provides simple visualizations such as charts and graphs, they lack the sophistication and power to further customize the visualization. Schneiderman describes a powerful technique in UI called direct manipulation, which is considered a more natural and intuitive interface for users, which has significant potential for improving the user experience. In our project, we explore different possible mechanisms, including direct manipulation, for improving the user interface for information visualizations.

5 Related Work

Collaboration in information visualization is not entirely new. The sense.us web application, designed by Jeffrey Heer, Fernanda Viegas, and Martin Wattenberg, was implemented as a system that would enable users to be able to comment on visualizations and share their observations about visualizations, enabling collaborative sense-making among the various users of the application. The site allowed users to be able to discuss visualizations, make annotations, and explore data collaboratively. The extent of the collaboration in this project is directed towards the end users who are assessing the visualization.

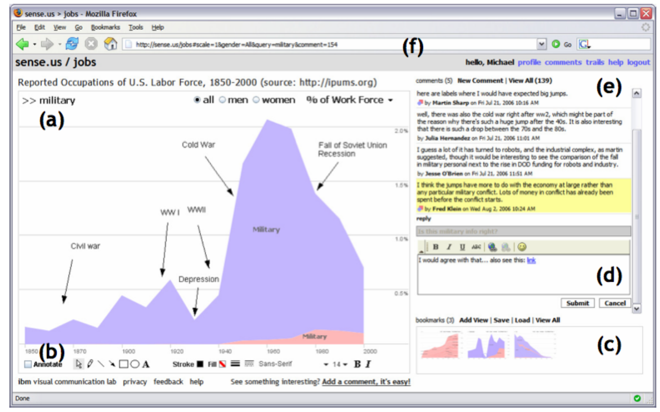


Figure 3: The sense.us web application, which supports asynchronous collaborative analysis of visualizations

Towards the other end of information visualization in the data collection arena, much of the research conducted in this spectrum is classified under Computer Supported Cooperative Work (CSCW) (Dekeyser, Watson). This area of research has been developed upon since it was first coined in 1984. Related to this sphere of work are various software packages for collaboration: online chat, instant messaging, web conferencing, version control systems, online spreadsheets, electronic calendars, and many others.

On both ends of the spectrum in producing information visualization, we see that there exist collaborative methods amongst those, but as far as bridging the gap between the two, there is seemingly little work that has been accomplished. Furthermore, the collaboration that takes place towards the information visualization spectrum has more to deal with the analysis of the visualization rather than creating the actual visualization. In our project, we seek to fill in the gaps that lay in between in order to facilitate the transfer from collecting the data up to the extent of producing the visualization.

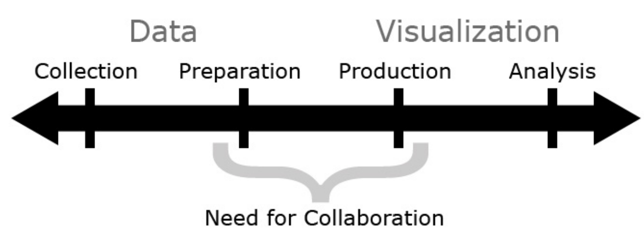


Figure 2b: Gaps of collaboration within the process of creating information visualizations

Another significant issue with the production of visualizations is that there is a significant learning curve for non-programmers and programmers alike. Michael Bostock and Jeffrey Heer created Protovis, a Javascript toolkit for producing visualizations, with the goal of allowing greater control and simplifying the ease of use to create visualizations. While Protovis allows for much control and expressiveness in creating the visualization, its ease of use is still not quite suitable for non-programmers, although its authors have expressed their interest in simplifying that process (Bostock, Heer). As Protovis is easily accessible, we found it worthwhile to develop our project to expand upon the goals of Protovis.

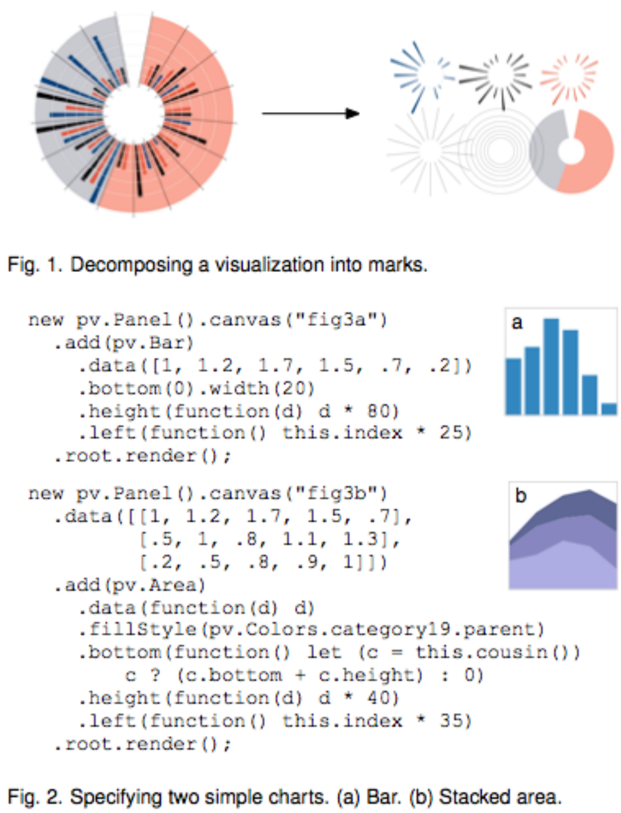


Figure 4: The Protovis JavaScript toolkit

6 METHODS

To approach the issue of collaboration in the space of information visualizations, we evaluated several different methods of improving the visualization system. To approach our problem, we looked first at the common approaches to data collection, which we estimated that the majority of this work was done using spreadsheet software. Judging from our point of view, we concluded that Microsoft Excel and Google Spreadsheets were two of the most widely used spreadsheet applications available to consumers, and from that point we began to assess some of the strengths and weaknesses of both of these software applications.

In our observations of the software, we noted that while Microsoft Excel was sufficient for the purposes that it was designed for (creating spreadsheets), its ability to produce visualizations and to customize the software were relatively limited. Furthermore, by nature of it being a desktop application, there is limited capacity to collaborate using the software. The other application, Google Spreadsheets, provided an API from which we could build extensions to the software, and being designed for collaboration, we found it to be more suitable for the purposes of our project. We also compared other alternatives such as using other open source spreadsheet applications so that we would not have to be locked in to the limitations of Google Docs, but we found that for the purposes of our project we would build upon Google Docs to demonstrate an implementation of our idea.

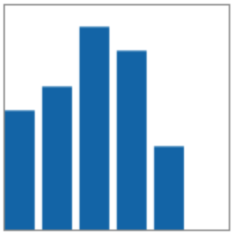
One of the design considerations for us in extending Google Spreadsheets was whether to develop a separate web application that used the Google Spreadsheets API or to develop a Gadget which could be embedded within an actual Google Spreadsheet. In our process, we decided that developing a Gadget would be the best method for the implementation, which would enable the users to be able to edit the data and to see the visualization at the same time. Another important aspect of our project is the integration of Protovis into Google Spreadsheets, which would enable more powerful and customizable visualizations, using Microsoft Excel as a baseline. We find that our approach would allow us to develop a simplified UI, as well as giving users more control over the production of the visualization. We also attempted to develop templates in which a user could produce their data, which could then be plugged into a visualization that has already been pre-made, which can then be modified by a common user.

In summary, for our implementation, we decided that by using Google Spreadsheets and the Gadget API as a central component for our project, we would be able to handle many of the goals that we have set out to accomplish with our project. In our implementation, we sought to demonstrate the usefulness of Protovis, and we hoped that through using both Google Spreadsheets, a popular web application, and Protovis, we would be able to envision an system in which designers could produce libraries of Protovis visualizations which could then be plugged in to spreadsheet data.

6.1 Direct Manipulation

There is this very explicit model in which Protovis visualizations are organized around that can be made much more obvious and explicit through some direct manipulation techniques and other user interface features.

Protovis is a graphical approach to data visualization where the user can employ graphical primitives to create custom visualizations. These graphical primitives are called marks, and each mark is data driven. They are declarative encodings of data. It is not completely free form where the user can create any arbitrary shape or interaction, but it's power comes from it's limitation. Things can be specified in the pixel level, yet there are still established rules of layout and graphics generations (bars, areas, lines) that abstract much of the detail. This gives the user control over what they need to have the most freedom in creating a custom visualization, without any of the unnecessary concerns.



```
new pv.Panel()
    .width(150)
    .height(150)
  .add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7])
    .bottom(0)
    .width(20)
    .height(function(d) d * 80)
    .left(function() this.index * 25)
  .root.render();
```

Figure 5: This bar chart encodes an array of values visually through its height. Marks are made for each item in the array, mapping the value at a specific index in the array to visual properties using anonymous functions (e.g., d * 80) or constants (e.g., 20). Thus, a mark represents a set of visual elements that share data and visual encodings.

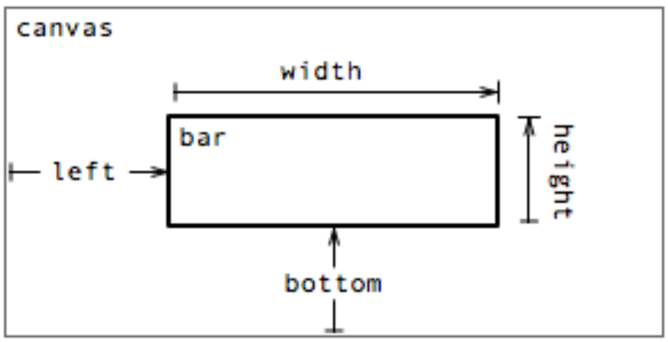
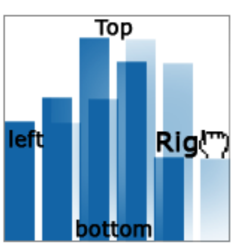
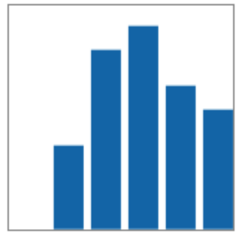


Figure 6: This figure shows the current state of Protovis. The code gives the user access to this underlining data driven model, however this model is never visually represented or directly manipulable by the user. If you look at the diagram above, it elucidates much of the box model structure of layout. We propose to visually break down the declarative code in an interactive way for users to more easily create and modify visualizations.

The user can get the feeling of actually composing the marks by arranging them in different ways by directly manipulating the visualization. Of course, like how Protovis is a graphical toolkit, allowing pixel-level control, but also abstracting away detail when appropriate, we want the interaction with composing the visualization to work in a similar way. We want people to be able to interact by arranging things in a per pixel way, but the interactions to be meaningful with creating a visualization, like dragging the labels around and having it snap to either, left, right, top, or bottom.

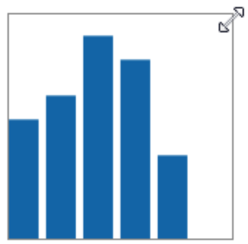



```
new pv.Panel()
    .width(150)
    .height(150)
  .add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7])
    .bottom(0)
    .width(20)
    .height(function(d) d * 80)
    .right(function() this.index * 25)
  .root.render();
```

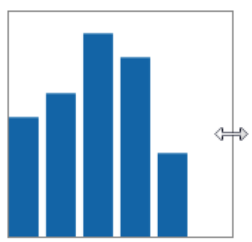


```
new pv.Panel()
    .width(150)
    .height(150)
  .add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7])
    .bottom(0)
    .width(20)
    .height(function(d) d * 80)
    .right(function() this.index * 25)
  .root.render();
```

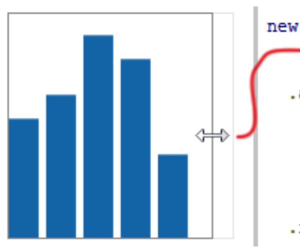
Figure 7,a,b: For the example above, one such way of changing the position of the bars within the canvas is to be able to click the appropriate part of the bar graph that will bring up four tool-tip like labels which indicate the four positions which the bars can be arranged. The user can then drag the graph onto one of those labels to get the new arrangement.



```
new pv.Panel()
    .width(150)
    .height(150)
  .add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7])
    .bottom(0)
    .width(20)
    .height(function(d) d * 80)
    .left(function() this.index * 25)
  .root.render();
```



```
new pv.Panel()
    .width(150)
    .height(150)
  .add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7])
    .bottom(0)
    .width(20)
    .height(function(d) d * 80)
    .left(function() this.index * 25)
  .root.render();
```



```
new pv.Panel()
    .width(143)
    .height(150)
  .add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7])
    .bottom(0)
    .width(20)
    .height(function(d) d * 80)
    .left(function() this.index * 25)
  .root.render();
```

Figure 8a,b,c: The label inherits the data and position property, which makes anchoring useful. The anchor—another convenience—is an invisible mark that inherits from the bar, while the label inherits from the anchor. However, sometimes it is not obvious how the anchor works since it is an invisible mark. The new user interface will make the anchor, which is an invisible mark, visible and directly manipulable through clicking and dragging. The below shows the anchor as a gray dot so that the user knows what the label is anchored to even.

```
new pv.Panel()
    .width(150)
    .height(150)
  .add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7, .1])
    .bottom(0)
    .width(19)
    .height(function(d) d * 70)
    .left(function() this.index * 24 + 5)
  .anchor("top").add(pv.Label)
    .textBaseline(function(d) (d < .2) ? "bottom" : "top")
    .textMargin(20)
  .root.render();
```

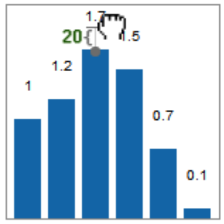


Figure 9: Using direct manipulation to control the placement of text. We may also consider implementation of algorithms for text placement in the system.

7 Results

After implementing a basic prototype of our project, we found that it was possible for users to be able to collaboratively input data through the Google Spreadsheets application. Much of the collaboration was supported through the Google Spreadsheets framework. In the current implementation, our prototype did not add much to the Protovis toolkit, but it did enable users to have more control over both the data and the source code at the same time.

8 Discussion & Future Work

Through our research, we recognize that collaboration in information visualization still has much room for further exploration. Within the past decade, web applications have been coming to similar levels of quality relative to that of desktop applications, and there will undoubtedly be a growth in web applications for information visualization, especially with the preeminence of web-based toolkits such as Protovis.

We observe that direct manipulation is not necessarily the most viable UI paradigm to attain to as there are limitations to what can be accomplished, as well as the difficulty of applying direct manipulation to the computing metaphors that would be involved in information visualization. We think that one of the important next steps to enhancing information visualization steps is to develop a more appropriate metaphor for the various components in an information visualization, as there needs to be a way for the visualization and the data to be modified in separate abstractions.

Acknowledgements

We would like to thank Jeffrey Heer and Michael Bostock for their feedback as we approached this project. We also want to thank Maneesh Agrawala for the information visualization class where we gained many insights into how to understand data and represent it.

References

Michael Bostock and Jeffrey Heer. Protovis: A Graphical Toolkit for Visualization, *IEEE Transactions on Visualization and Computer Graphics*, 2009.
J. Heer, F.B. Viegas, and M. Wattenberg. Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. *In Proc. ACM CHI*, pages 1029–1038, 2007.

Dekeyser, S., and Watson, R. Extending google docs to collaborate on research papers. Tech. rep., The University of Southern Queensland, Australia, 2006.

Ed Huai-hsin Chi, Phillip Barry, John Riedl, and Joseph Konstan. A Spreadsheet Approach to Information Visualization, *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, 1997.

Raymond R. Panko. What We Know About Spreadsheet Errors, *Journal of End User Computing, Volume 10, No 2. Spring 1998*, pp. 15-21.