Many Eyes: A Site for Visualization at Internet Scale

Fernanda B. Viégas, Martin Wattenberg, Frank van Ham, Jesse Kriss, Matt McKeon

Abstract— We describe the design and deployment of Many Eyes, a public web site where users may upload data, create interactive visualizations, and carry on discussions. The goal of the site is to support collaboration around visualizations at a large scale by fostering a social style of data analysis in which visualizations not only serve as a discovery tool for individuals but also as a medium to spur discussion among users. To support this goal, the site includes novel mechanisms for end-user creation of visualizations and asynchronous collaboration around those visualizations. In addition to describing these technologies, we provide a preliminary report on the activity of our users.

Index Terms—Visualization, World Wide Web, Social Software, Social Data Analysis, Communication-Minded Visualization.

1 Introduction

When visualization researchers talk about scaling, we usually mean handling large data sets. We seek ways to draw huge graphs, explore high dimensional spaces, and display databases with billions of rows. In this paper, however, we consider an alternate perspective: Instead of scaling the size of the data, what happens when we scale the size of the audience?

This perspective is suggested by the rise of the Web as a visualization platform. Recent years have witnessed internet-based visualizations ranging from political art projects (e.g. Theyrule [25]) to New York Times stories (Faces of the Dead [16]). These displays are seen by thousands and it is natural to ask what new opportunities arise when visualizations move to an environment where vast crowds of people can create, view and discuss them. Not only are interactive visualizations a key medium for communication in a data-rich world, but preliminary reports hint that visualizations potentially have a catalytic effect on storytelling [26] and collective data analysis [28].

Unfortunately there are two main roadblocks to overcome before visualization has a chance to fulfil this potential. First, the creation and publishing of interactive visualizations remains accessible only to specialists. While frameworks such as Flash and Processing ease development, they do not help non-programmers. End user tools for sophisticated visualizations such as Tableau [23] and Spotfire [21] still require expertise, installation and training. Furthermore, none supports easy publishing to the public web. A non-developer blogger who wishes to write about an interesting data set, for instance, is currently limited to static charts.

Even if that hypothetical blogger manages to publish an interactive visualization, a second challenge remains: how can readers discuss it? Web-based visualization has obvious social and collaborative potential, but without special technology it can be hard to have a discussion around a visualization. If one person sees something interesting, for instance, how can they point it out to others? Systems that do explicitly support collaboration, e.g., DEVise [14], Spotfire [21], and the Command Post of the Future [20], have been aimed at scientists and other experts operating in a closed intranet or military environment. The open Web is a very different environment: the designers of the Command Post of the Future likely did not want to encourage public discussion about the discoveries its users made.

This paper describes a public web site, Many Eyes, that addresses the challenges of end user construction and the unique environment of open web-based collaboration.

• *All authors are with IBM Research. E-Mail: viegasf@us.ibm.com, mwatten@us.ibm.com, fvanham@us.ibm.com, jesse.kriss@us.ibm.com, mmmckeon@us.ibm.com respectively.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007. Published 14 September 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

The site launched on January 23, 2007 and allows users to upload data, create visualizations of that data, and leave comments on both visualizations and data sets. In addition to constituting a research platform, the site is an effort to "democratize" visualization technology. By this we mean both providing the technology to the broadest possible audience and fostering a democratic deliberative style of data analysis.

One lesson learned from building this site is that the most difficult issues in scaling the audience are not necessarily related to engineering. Instead, many of the key questions are those of design and user experience. To give one example of the difference between Many Eyes and a system such as Tableau or Spotfire, consider that many of our users arrive directly at our visualizations for the first time via links from external web sites. No one is an accidental user of a commercial system such as Tableau; by contrast, our audience often views the site with little context and absolutely no training. If they do not understand what they see, they can (and will) simply click away from the site, never to return.

This fact of life on the web implies a goal of instant usability that has ramifications for all aspects of the site. The data model we support, features offered to users, and flavors of visualization techniques supported on Many Eyes all reflect the need for broad, immediate accessibility. These implications are reflected in design choices that need to strike the delicate balance between powerful data-analysis capabilities and accessibility to the non-expert visualization user.

Our aim in this paper is to describe the design of the site and the rationale behind the decisions we made. We will focus on decisions that affected our two overarching goals: enabling end-user creation of visualizations and fostering large-scale collaborative usage. The design decisions in Many Eyes fall into three different areas: information visualization, end-user data collection and manipulation, and the social aspects of collaborative analysis. The organization of the paper reflects this division. Section 2 is an overview of the Many Eyes web site and its basic functions. Section 3 presents the data considerations in the site. Sections 4 and 5 cover the different kinds of visualizations and the social/collaborative features available. Finally, Section 6 presents preliminary results on usage. Since the work introduced here spans a variety of disciplines, related work is discussed in multiple sections, each one addressing a specific area of research.

1.1 Related Work—*asynchronous collaborative visualization*

There is a large body of work on collaboration in visualization systems [2][14][20][27]. Long-standing research has explored techniques for collocated collaboration (e.g., large displays and shared workspaces) and synchronous distance work (e.g., real-time shared visual exploration environments). However, less research

1077-2626/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

attention has been devoted to asynchronous collaboration around visualizations or to the questions that arise in mass-audience visualizations.

Sense.us [12] is probably the research project that comes closest in spirit to Many Eyes. It is a web-based prototype that supports commenting and annotations on visualizations of US Census data. The system provides mechanisms for facilitating view sharing, threaded discussion across views, and the construction of tours and presentations. Unlike Many Eyes, Sense.us was pre-populated with six visualizations; users could not contribute new data sets or create new visualizations. The system was deployed on a corporate intranet, as opposed to the public web and was designed to explore issues of discussion and annotation rather than content collection and creation around visualization. With a small number of visualizations produced by the site administrator and its presence in corporate environment, Sense.us felt very different from Many Eyes, which contains thousands of visualizations and data sets contributed by users, spanning a wide array of serious and non-serious themes.

A number of commercial systems have begun to explore the idea of asynchronous communication around data. Several web sites (e.g., Dataplace [9], Data360 [8], Swivel [22], DabbleDB [7] and Chartall [5]) allow users to upload and graph data. Swivel, Data360, and Chartall allow users to make comments; Swivel especially seems to aim at scaling to a large audience, styling itself as "YouTube for data". All of these rely on static standard business graphics, however, and so do not tackle issues of state bookmarking and end-user construction that arise in the context of sophisticated interactive visualizations.

DEVise [14] is an early exploration into the benefits of sharable visualizations. It offers both customizable visual mappings and sharable views, as well as basic annotation functionality. Although designed to run in a browser it was not designed to be publicly accessible and most of its visualizations were relatively static.

The commercial Spotfire [21] system does feature sophisticated visualizations and includes a product called DecisionSite Posters, a web-based system that enables users to create bookmarks and comments attached to specific views of a visualization. While impressive, the "posters" are essentially an adjunct to the main desktop Spotfire application. Because this product is aimed at intranet usage by expert analysts, it does not explore the design issues involved in scaling to mass audiences on the public web.

2 A Brief Overview of the Many Eyes site

Many Eyes is roughly modeled on well-known participatory sites such as Flickr [10] and YouTube [30]. The central activities on the site are to upload data, construct visualizations, and leave comments on either data sets or visualizations. Each of these activities is described in detail in subsequent sections.

To navigate the continuously growing collections of visualizations, data sets, and comments, the site contains "browsing" pages that display recent contributions. The data set browsing page, for instance, shows a table with the latest data sets to have been uploaded to the site. The table also displays metadata about each data set: keywords ("tags"), source, the contributor's username, size in bytes and number of rows, date of contribution and links to existing visualizations with that data set (Fig 1).

The browse pages are not the only navigational aids. Every day the home page prominently features four visualizations that are chosen by the designers of the site and typically reflect newsworthy events or good models of visualization usage. For users who are interested in a specific topic, we provide a simple text based search function for both data sets and visualizations.

All visualizations and data sets on Many Eyes have an attached discussion forum where users can share textual comments and links to other webpages (Fig. 4). Since all content on Many Eyes resides at a fixed URL, users can also easily link to other visualizations on the site from both inside and outside Many Eyes..

After the site launched in January 2007, it was featured in several prominent blogs as well as some mainstream media outlets. This publicity has provided a steady stream of visitors to the site.

3 Data

Data is at the root of all activity on Many Eyes. Although we seeded the site with a few data sets and visualizations, most of the content is contributed by ordinary users, who upload data sets, discuss them, create visualizations from the data, and then discuss those visualizations.

The fact that users can upload their own data offers a number of potential benefits. The benefits to individuals are clear, since they can visualize their own data. One might also hope for a collective benefit: a user might upload one set of data, and then another user could augment it by uploading additional, related information. Finally, from the point of view of research, this is a unique opportunity to understand user demand: what types of data do people really want to visualize?

Gaining these benefits, however, means balancing complex and conflicting constraints on the design. The fact that data is uploaded to the site by end users means that the data model needs to be easily understandable, with a format that is appropriate for non programmers. At the same time, the format needs to be flexible enough to express the data structures used by visualizations such as treemaps and graph layout algorithms.

In addition to constraints on the data model, we face constraints on changes to data. Data sets and their visualizations are subjects of discussion so that comments and annotations may go stale if the underlying data is edited. A second difficulty is that the kind of data "reshaping" that is often necessary in preparing a data set for visualization can be difficult to explain to lay users without the visual context.

3.1 The data model

The core data model used by Many Eyes is a table: that is, a set of named columns, each of the same length. Each column has a type, which currently can be either text or numeric. The site also supports data that comes in the form of unstructured text, i.e. a sequence of characters, equivalent to a Java String or a CLOB ("character large object") in a relational database.

Each data set, whether a table or unstructured text, is associated with a collection of metadata. Some metadata, such as a (required) title, the source of the data and a paragraph-length description is provided by the user. Other metadata is automatically set by the system, such as creation date and author. Datasets are currently stored on our servers in plain text format, a set-up that has proven to be efficient and simple.

3.2 Uploading data

Users upload data via an HTML form. The form contains a text area where the user can paste in a data set. The data can either be freeform text, in which case it is interpreted as unstructured data, or a tab-delimited grid, in which case it is interpreted as a table. We chose to use tabs as delimiters because it allows users to simply copy and paste from Microsoft Excel or Open Office.

In the case of tabular data, the system makes a guess about whether each column is numeric or textual, using heuristics that account for currency symbols and the difference between US or European punctuation. In the event that the system makes a mistake, the user may override the automatic type choice.

As in the visualization-creation step, described later, user education is an important consideration: after all, many users will not have experience in preparing data so that it is understandable by others. The user is encouraged to provide as much metadata as possible and we include a prominent reminder to label units. Our "help" page describes not just the technical details of the data format, but a section on the "elements of data style" that guides users in creating data that will be comprehensible by others.

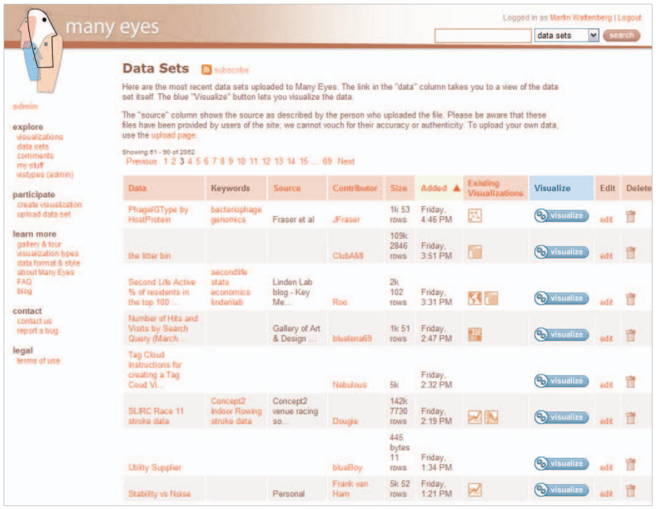

Fig. 1. The data browse page on Many Eyes, giving an overview of all datasets currently uploaded to the site.

A key decision was whether to require registration for users to upload data to the site. Registration is a significant barrier to entry, so on its face it would seem to work against our desire to reach a broad audience. For two reasons, however, we decided to require registration. First, we wanted to provide a slight barrier to entry to prevent frivolous or malicious data uploads. Second, during the registration process we could require explicit agreement to certain legal terms.

Finally, users can navigate the collection of data sets in a variety of ways: a standard search box; use of keywords or "tags"; and a list that can be ordered by upload data, contributor, and other meta data.

3.3 Working with data

Each data set that is uploaded to the site is given its own page. This page contains metadata, a snippet of the data set, and a discussion forum on which users can talk about the information. We also provide a link to the original version of the data in plain text form. At the bottom of the data page is a button that starts the process of matching the data with a visualization (see section 4.2) as well as a set of small icons that indicate which methods have been used to visualize the data already.

As mentioned above, Many Eyes currently does not allow any direct editing of the data sets: each is immutable. The reason is that editing data could invalidate any visualizations created from the data set as well as comments made on those visualizations. This was a difficult decision to make, however, since users clearly would benefit from being able to edit data: not only have we received numerous requests for this feature, but we have seen users upload many separate versions of a file as they correct errors. We do currently allow users to delete their own data sets if no visualizations have been made from them.

3.4 Related work—*data representation*

Several online data storage services aimed at end users exist already. Two examples are Intuit's Quickbase [18] and DabbleDB [7]. Quickbase uses simple tables, much like Many Eyes, but with a somewhat richer set of types. DabbleDB is more akin to a database and also offers basic visualization features. As mentioned in the introduction, several other social data analysis sites have appeared contemporaneously with Many Eyes, including Data360, Swivel, and Chartall. All using similar tabular formats, although none allow the kind of data reshaping described in Section 4.

Why have these sites converged on this solution? One answer is that all want to exploit the vast amount of data stored in Microsoft Excel spreadsheets. Furthermore, tables are simple and well-understood by end users.

Two alternatives that we considered were to expose a standard relational database model or an OLAP data cube. While this would make it easy to extend Many Eyes to interface with standard data stores (as with Spotfire or Tableau) it seemed unlikely to appeal to end users. Unfortunately, even the ability to use simplified systems such as Microsoft Access is quite rare. Similarly, we judged data cubes to be beyond the knowledge of a typical user: The closest thing to a data cube that most people have encountered is a Microsoft Excel pivot table, and informal discussions with potential users uncovered a lack of understanding of (even outright hostility toward) this tool.

A similar problem exists with semi-structured data models based on XML. Although flexible and friendly to developers, an XML-based model seemed likely to be a barrier for nonprogrammers. For example we asked a professor who regularly teaches introductory statistics if she thought importing XML-formatted data would be important to Many Eyes; her answer was, "What's XML?"

4 VISUALIZATIONS

Many Eyes relies on a pure web-based model in order to reach the largest possible audience. While visualizations have existed on the web for more than a decade, these have been constructed offline and then separately published to the web. On Many Eyes, however, visualizations can be constructed and published by users without ever leaving the browser. Although a browser-based environment imposes some constraints, it is critical to the goal of scaling our audience.

The visualization technology in Many Eyes consists of a set of individual components that represent different display techniques. To create a visualization, a user combines a component with a data set, in a manner described below. Once created, the visualization is given its own home page, which also contains metadata, a link to the original data set, and an area for discussion. Each page carries a prominent link to a compact description that explains the visualization technique being used, what its data requirements are and in what cases it can best be applied. Explanation pages may also contain "expert notes," which usually address some of the subtleties of using a specific technique over another—for example, when one should use a bar chart over a line graph.

4.1 Visualization components

Many Eyes provides more than a dozen different types of basic visualization components (see Fig. 2 for a small taste). The menu includes a mix of standard business graphics (e.g., bar charts), well-established academic techniques (e.g., treemaps), and a few experimental components (bubblecharts and tag clouds). Each of these visualization components was implemented as a separate java applet to keep download sizes to a minimum. Since we lack the space to discuss every single applet separately, we will suffice with a short overview of some of our more experimental visualizations.

4.1.1 Bubble charts

One of the most common types of real world data is simply a list of labeled numbers. Movie titles and box office grosses, for instance, or basketball players and salaries. Such lists are often easily displayed via standard bar charts or histograms, but these methods can run into trouble for highly skewed power-law distributions. To address this difficulty, we included a "bubble chart" which represents items by labelled circles, whose areas are proportional to the displayed quantity. These charts effectively perform a visual square-root transform.

4.1.2 Stack graph for categories

Many time series have a hierarchical structure: the history of the United States federal budget, for example, can be recursively subdivided into various departmental levels: defense overall, atomic weapons, and so forth. To display this type of data, we included a "stack graph for categories" visualization, based on the

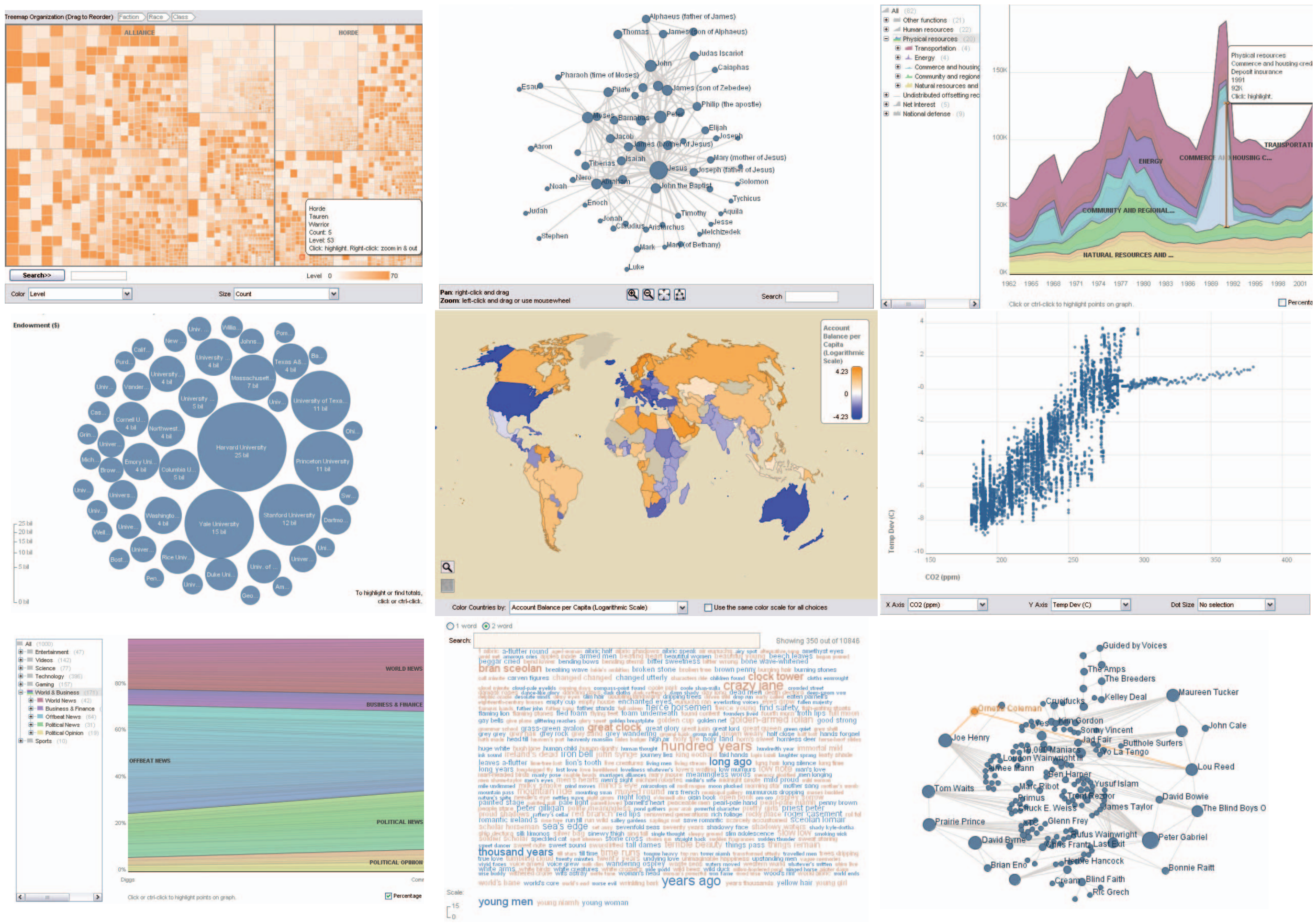

Fig. 2. A tiny cross-section of the visualizations users created by matching their own data with our visualization components. From left to right and top to bottom: A treemap on character data in a World of Warcraft realm, a network diagram of social relationships in the bible, a stack graph showing the categorized spending of the US government over time, a bubble chart showing the size of the endowments of different US universities, a world map showing the average account balance per capita, a scatterplot comparing CO2 levels against temperature, a cleverly used stacked graph that shows relative differences in the type of postings on two online link aggregators, a tag cloud showing two word occurrences in the work of Yeats and a network diagram of musical ties.

BookVoyager design described in [29]. To the left of the graph is a tree control that follows the hierarchical structure of the time series. Clicking on elements in the tree control filters the stack graph to show only time series at that level of hierarchy. The icon representing each hierarchy level is a tiny "sparkline" graph that provides a preview of what the user will see upon clicking. This helps new users understand how the control works, as well as providing an overview of the individual series.

4.1.3 Graph drawing

Although network data can be found in many different application areas, there are surprisingly few online tools that can provide a graph visualization of a user uploaded file. The Graph Drawing Server [3] was one of these services, but seems to be inactive now. The Large Graph Layout [1] server is another but only returns a list of coordinates by e-mail, which can hardly be considered interactive. We have implemented a force directed graph drawing algorithm in a zoomable user interface. The layout algorithms (although implemented in java) are able to provide layouts for graphs up to a thousand nodes in a couple of seconds, and only have to run when the visualization is created. Although many different 'standardized' dataformats for graphs exist, we settled on a simple edge table. Although not the most efficient storage format it proved both highly understandable to the end users, and fitted nicely with our table oriented design.

4.1.4 Tag clouds

Since one of the types of data that users frequently uploaded were word counts, we decided to include a visualization that would take unstructured text as input. Tag clouds can quickly give the user an overview of the most salient terms in a large corpus of text. We implemented a number of improvements over the standard tag cloud applets. One of these is the ability to measure the frequency of two word tags in the text. The other includes the ability to dynamically filter the tag cloud by entering query strings in a text box. We found tag clouds attracting a whole new set of users, whose interest is primarily in textual data instead of numbers.

4.2 Visual mapping

A visualization is created by matching a dataset with a visualization component. Of course not all visualizations display the same type of input data. A treemap, for example, requires a number of textual columns to define its hierarchy and two numerical columns that map to size and color. On the other hand, a basic scatterplot requires two numerical columns (one for each axis), an optional numerical column that specifies the size of each dot and a textual column for the labels. A single datafile might be used to drive both visualizations, offering different perspectives on the data (Figure 3). To set up this mapping, the visualization components need to be able to express its data needs in a precise manner. In Many Eyes, a component's data needs are expressed in a schema specified by the

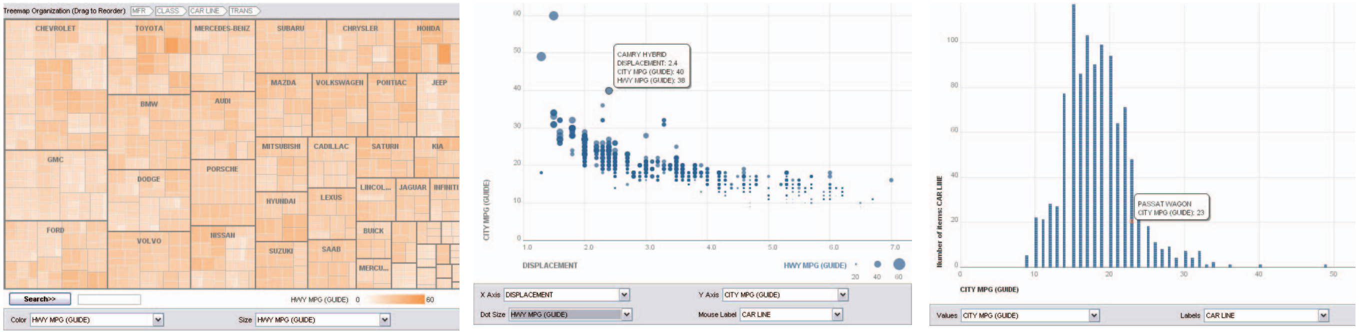

Fig. 3. Three user generated visualizations offering different perspectives on the same dataset on car fuel economy. The grey areas on the top and bottom are automatically generated by the application and allow the user to browse through different dimensions in the data.

visualization programmer. A schema is a set of named, typed slots. Currently we support 5 different slot types:

- **Unstructured slots (*U*):** flat unstructured text
- **Numeric slots (*N*)**: a single column of numeric data.
- **Textual slots (*T*)**: a single column of textual data.
- **Multiple numeric slots (N^+):** one or more columns of numeric data.
- **Multiple textual slots (T^+):** one or more columns of textual data.

Using these 5 basic slot types we can express the data needs for each visualization component. Hierarchies on the data can be expressed as an ordered set of textual columns, where each row in the set describes the path from the top of the hierarchy to the leaf item. We can then express the schemas of a treemap and a scatterplot in the following manner, respectively : {hierarchy : *T*+, size : *N*, color : *N* } and {Xaxis : *N*, Yaxis : *N*, label : *T*, [Dotsize : *N*]}.

Slots can be made optional, so that no mapping is necessary for a visualization to be complete. For example, our scatterplot includes an optional slot for dot size, while the slots for x- and y-axes are required. The challenge in matching up a visualization technique with a dataset is to map the correct columns in the data table to a slot. Since both slots and columns are typed we can make some inference on potential matches. For Many Eyes, we decided on a mix of automated and user generated mapping, depending on the circumstances. For slots that can accept multiple columns (i.e. slot types *N*+ and *T*+) we feed the visualization all columns in the table that have a matching type. For slots that only take a single column, we decided to partly leave this mapping to the user and allow him or her to choose from a number of type compatible columns.

4.3 End User Data Manipulation

One option is to allow the actual viewers of the final visualization to change the mapping, making it possible to change the data shown in the visualization on the fly. This offers a fast way to browse through different dimensions on a dataset. For example, matching a single numerical table with a scatterplot allows the end user to select two columns to display on the x and y axes from the collection of all possible columns. The user interface for these selections is generated automatically from the mapping. For each slot where multiple columns might be a valid match we generate a drop down box below the visualization (see Fig. 3). However, depending on the visualization technique it might be very well possible that not every selection in the drop down box will produce meaningful visualization results. Take the world map as an example: it requires a text column as input for the location slot (typically this contains the country name), but if we have multiple text columns in the data, generally only one will produce a meaningful result. In this case the selection has to be done by the person creating the initial visualization, after which it cannot be changed by viewers. The choice between end user selection and creator selection is specified by the visualization programmer for each slot.

Table 1: Available Visualization Types in Many Eyes

Technique	Data schema
Bubblechart Histogram Pie Chart Maps Tag Cloud	{Labels / item names : T, Values : N}
Bar chart Line graph Stack graph	{Axis labels : T, Values : N+}
Network diagram	{From : T, To : T}
Scatterplot	{Xaxis : N, Yaxis : N, Label: T, [Dotsize : N] }
Stack graph/ categories	{Hierarchy : T+, Values : N+}
Treemaps	{Hierarchy : T+, Size : N , Color : N}
Tag Cloud	{U}

4.3.1 Contextual Data Transformation

The ability to change mapping of columns to visual attributes is not the only data manipulation option we included. As described in Section 3, there are many types of data transformation or reshaping that users may need to perform. In many data acquisition pipelines these transformations occur in a separate stage. Due to our end-user audience we have opted for *contextual data transformation*, that is, we let users perform all such transformation in the context of creating a visualization, so that they may easily see and understand the results of their actions.

For example, in several visualization components it may make sense to transpose the rows and columns of the input data table. Rather than asking the user to perform this operation before starting to visualize their data, a “Flip rows/columns” button is made available whenever the column types permit the operation.

In some cases it might also be useful for the end user to be able to reorder the columns that were fed into a multiple column slot (i.e. *N*+ or *T*+). For example, multiple textual columns can define a hierarchy on items, but the user might want to reorder them to get different orders of aggregation. We designed a widget that allows users to reorder column names by drag and drop, again, changing the visualization on the fly (see leftmost sample in Figure 3).

A more subtle type of manipulation relates to rolling up data sets. In early trials it became clear that users expected a certain kind of automatic aggregation. For example, imagine a bar chart of basketball salaries, where the columns in the underlying data set are player name, position, and salary. When the label slot was set to player, the bar chart yields—as expected—a chart of individual player salaries. However, when the label slot was set to position test users indicated that they expected the bars for each position to be aggregated, preferably by averaging. Interestingly, users did not always expect averaging. In some bar charts they wanted summation rather than averaging. In pie charts—which are designed to show relative totals—it seemed that aggregation should always occur by summation. And in a scatterplot, users did not expect any aggregation! To handle these expectations, we created a set of aggregation widgets with customizable default actions for the different visualization components.

One last example of data normalization relates to the maps. While U.S. state names are fairly standardized, names of countries can appear in many different ways: e.g., “Democratic People's Republic of Korea” “Korea, People’s Republic,” or simply “North Korea.” When one of our map components does not recognize a name, it uses a simple distance measure to suggest a likely match, while allowing users to override this match as necessary.

4.4 Related work—*end user construction of visualizations*

Visualization component models that include a notion of mapping table columns to visual attributes are not new—in fact they may be better characterized as a known best practice in the field. At the simplest level, Microsoft Excel generates graphs by letting users select columns that feed into business charts. The well-known work of [15], parameterized the different visual encodings in visualizations, using them to automatically choose a meaningful visual representation for a given dataset. Some frameworks [19][14] completely parameterize the visualizations and map directly between a data tuple and the shape and position of its representation on screen. More flexible end user desktop visualization applications include Spotfire [21] and Tableau [23], which both offer advanced data mapping paradigms. Many Eyes may be most similar to the systems of [17] and [24] that provide a number of commonly used visualizations with predefined slots and map data tuples to slots.

Where Many Eyes differs from existing end-user visualization systems is the pure web-based interaction and publishing model, which makes visualization construction tools immediately available to millions of people with internet access. While Spotfire allows the publishing of “posters,” these must be created with the desktop application and represent a subset of the desktop functionality. In addition, the contextual data transformation approach distinguishes Many Eyes from systems that include a separate data reshaping stage. Finally, the palette of visualization components and their design reflects the need to provide instant utility to users on a broad range of data sets.

5 Social Features

So far we have concentrated on the constraints of an open web platform, and described the tradeoffs necessary to meet them. In this section we discuss some of collaborative features that exploit the opportunities of an open web deployment. In particular, we describe how we allow users to engage their collective intelligence, by pointing to items of interest, sharing insights, asking questions, and monitoring activity on items of interest.

Previous systems have explored such capabilities, but as we discuss below the web provides a unique social environment. One important distinction is that communication around visualizations can potentially occur both on and off site. Thus, users should not be restricted to discussing Many Eyes visualization only on the site. For this reason, it became important to provide points of entrance to discussions that were external to the site itself—for instance on blogs or in forums.

5.1 On-site communication

The main communication feature in Many Eyes is the textual comment. Comments exist in the context of specific visualizations and data sets. As users interact with a visualization, they can enter comments very much in the same way comments are entered into a blog. The same is true of data sets; each data set has a page where comments may be entered. The other communication features, described next in this section, are anchored in textual comments. In a sense, comments are the medium for all communication that happens on the site.

5.1.1 User Identity

Another crucial aspect of community oriented web sites is user identity. In order for a community to evolve over time, people need to be able to interact with each other with a minimum of persistent identity so that they may recognize each other and build up on previous interactions. On Many Eyes, a user’s identity is directly related to their activity history. By registering to enter the site, users create persistent handles that become part of their identity. Each registered user has a page that lists all of their contributions to the site: uploaded data sets, created visualizations, and comments. The page serves two purposes: it allows users to keep track of their activity in a single place and, at the same time, the accumulated history functions as an identity marker on Many Eyes. By looking at another user’s page, one can quickly get a sense of their interests.

One of the challenging aspects of sharing insights in the context of asynchronous, interactive visualizations is establishing common ground [6]. Different users need to be able to point out specific items of interest. Many Eyes supports common ground creation with two features: visualization bookmarks and visualization annotation. Whereas bookmarks allow users to capture the state of a visualization, annotation enables users to highlight specific items within a particular state of a visualization.

5.1.2 Visualization annotations and bookmarks

An interactive visualization may have hundreds of states and, a lot of times, when users wish to talk about points of interest, they may want to refer to a specific view of that visualization—defined by the settings of filtering, navigation, and parameters of visual encoding. Thus, capturing state information is essential for communication in an environment like Many Eyes. To this end we utilize a simple URL bookmarking mechanism that points back to particular views of the visualization. This approach to state sharing is common in other systems as well [12].

Every time a user creates a visualization on Many Eyes, its default view becomes a new bookmark in the system. Additionally, users have the option of including a “snapshot” of the visualization state every time they contribute a comment. Each snapshot is a new, unique URL that captures the state of the visualization at the time the comment was made. This allows users to both easily link to different views on a visualization from their comments as well as easy outside linking to visualizations on Many Eyes.

A lot of times, however, users may also need to highlight specific items within a given state of a visualization—i.e. within a given bookmark. Many Eyes supports this activity by allowing users to include graphical annotations in the comments they make.

Annotations take various forms in different visualizations types—for instance, a selection in a scatterplot looks different from a selection in a stacked graph. At the same time, it is important for visualizations across the site to share, as much as possible, a consistent visual language. In building this shared visual language, we have carefully controlled elements such as color—all visualizations share the same color palette—typography, and animated transitions. Item selection is another area where Many Eyes keeps consistency through color and active resuse of simple highlighting mechanisms across visualizations. Users are allowed to make multiple item selections (using either the “shift” or the “control” keys on the keyboard) in every visualization. We used a common highlighting color in all visualiations with the exception of the piechart, in which case selected slices are detached from the chart.

In some cases, the highlighting capabilities on Many Eyes serve additional purposes. For example, in the pie and bubble charts, selecting multiple items enables users to find the total sum of values of all selected items as well as the percentage represented by this group of points. On the network diagram, in addition to highlighting items for discussion, selection helps clarify structural details of the graph. Because highlighting a node also highlights its edges, it becomes easier to grasp the neighborhood structure of a node that otherwise might be obscured by other elements in the graph.

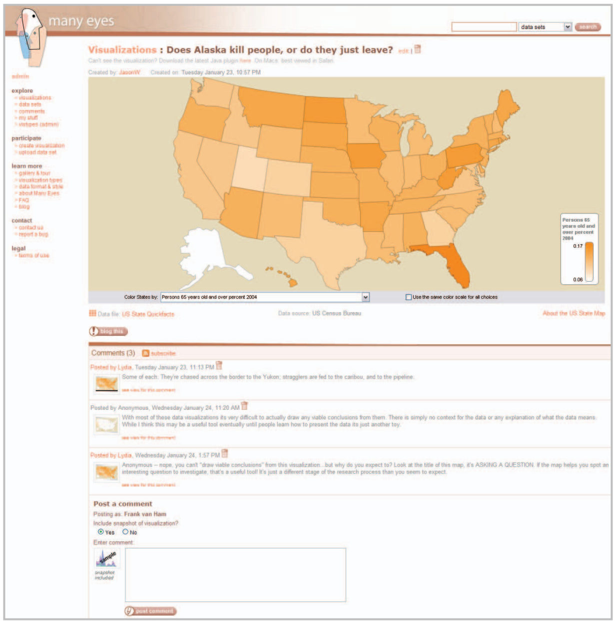

Fig. 4. The visualization and commenting interface on Many Eyes: A user has created a visualization, on which 3 other people have commented. Left of each comment is a small bookmark thumbnail that indicates the state of the user's visualization when the comment was posted. Clicking a thumbnail modifies the state of the 'live' visualization to match the bookmark.

5.2 Off-site communication features:

One of the essential differences between Many Eyes and other collaborative visualization systems is that it exists in the context of the open web. To foster collaboration around visualizations we wanted to tap into the often lively discussions that are carried on in various blogs and other online communities. We took special care in providing easy access from outside sources to ManyEyes. Each bookmark in a visualization is published at its own unique URL, making it extremely simple for an outside source to directly link to a view on a visualization. Another feature that was targeted specifically at bloggers is a "blog this" button that appears underneath each visualization on the site. This button generates a snippet of html code that users can copy and paste onto their blogs or into the comments section. The code contains both a static snapshot of the visualization and a link back to the URL of the visualization on the Many Eyes site (see Fig. 4). This feature enables bloggers to easily embed images of visualizations into their sites and thus carry on discussions about items of interest outside of Many Eyes.

Many Eyes also features RSS feeds for data sets, visualizations, and comments. Users can sign up for feeds of the latest visualizations or data sets contributed to the site overall or they can sign up to watch for new comments to specific visualizations and data sets.

5.3 Related Work—*visual collaboration*

The concept of making visual data analysis more of a social effort is gaining popularity. Applications such as Google Earth [11] have enabled not only the creation of a wealth of user-generated geographic visualizations, but also the emergence of large numbers of discussions around visual data. An official Google Earth community has sprung up, where users discuss topics ranging from the satellite images found in the application to the availability of cartographic information about places around the world.

Although not a tool for data analysis per se, Flickr, an online service for photo sharing [10], has become the embodiment of social annotation and discussion around visual information. The site has been highly successful in enabling users to easily create discussion groups around sets of photographs and to collectively annotate the images.

As discussed above, Swivel and Data360 allow users to upload, share and discuss their own datasets with other users. Dataplace [9] is a similar site allowing users to obtain and visualize basic populations statistics on different areas in the US, but does not allow users to upload their own data. The main difference between Many Eyes and these products is that, instead of providing static business graphics, Many Eyes offers a number of interactive visualizations of user's data. This interactivity allows users to drill down into details, view the data from different perspectives and generally makes the visualizations fun to use. The importance of the latter should not be overlooked by a site that targets the average internet user.

In terms of interactivity, Spotfire [21] and, to a lesser extent, Devise [14] also offer interactive visualizations that can be shared among users of the same application. However, Many Eyes lives on the web, where the potential audience for a visualization is greater by multiple orders of magnitude and visualizations can be linked into any online document using hyperlinks. We think that the combination of the enormous amount of collaboration infrastructure available on the web (think of blogs, forums, wiki's and RSS feeds for example) and a webservice like Many Eyes where users can upload, visualize and share their own data brings opens new doors for communication centered visualization.

6 Evaluation & Early Usage

How have our design decisions worked out in practice? This section provides a short overview of activity on the site, although it is hard to capture the full range of activity.

In the first two months of the site's life it has received about 400,000 non-robot page views, divided into 100,000 user sessions, and has gathered 1463 registered users. Users have uploaded roughly 2,100 data sets, created 1,700 visualizations, and made about 450 comments. Of the comments, about 90% have occurred on visualizations rather than data sets. This latter fact may indicate that the visualizations do have a catalyzing effect on conversation, especially given that there are more data sets than visualizations.

All of the visualization techniques have been used at least 25 times. The relative proportions (excluding visualizations created by members of our lab) are shown in . It is interesting to note that the top four visualization types are the non-standard ones. It is unclear whether this indicates an appetite for complex, experimental visualizations, or simply that people who wish to make bar charts have other options.

Table 2 Usage Statistics for the Different Visualization Types.

Visualization Component	Percentage of use
Bubble chart	15%
Network Diagram	12
Tag Cloud	11 (on site for only one month)
Treemaps	10
Bar Chart	9
Line Graph	9
World Map	8
Scatterplot	7
US State Map	7
Stack Graph for Categories	4
Block Histogram	4
Stack Graph	3
Pie Chart	1

Over 42% of registered Many Eyes users have uploaded at least one data set and 29% have created at least one visualization. Of those that uploaded data sets, 63% provided a source for the data and 40% also provided an URL for the data source. This level of data referencing is shockingly high considering that users are not required to provide sources for the data they contribute to the site.

One of the most distinctive aspects of Many Eyes is that it exists as part of the web ecosystem. In fact, we think the Internet has two distinct characteristics that make it uniquely suited as a platform for discussion and discovery. Firstly, its massive scale means that there is always another person out there that shares your interests. This makes it easier to attain the critical mass needed for a discussion site. Secondly, the ability to easily link different information together avoids this mass being fragmented over disconnected sites and allows users to relatively easily adapt different types of tools for their personal use. As an example exemplifying both of these properties, one particular user created a network visualization that showed different textual co-occurences of names in the bible (see Fig. 2) and linked to it on their Bible Sociometrics blog. This blogpost was subsequently picked up by different feed aggregators and received a highly ranked position. This prompted many more users to visit the original visualization on our site. A number of these users also interested in bible metrics then started uploading their own bible related datasets, and used these to create new visualizations, many of which were posted on different bible related blogs.

Our registered users range from scientists to mid level managers and self-proclaimed data geeks to sports fans. 625 of these have personally uploaded data, 425 have created a visualization and 113 have left comments on Many Eyes. Some of these visualizations quickly identified incorrect data points, even in datasets that came from respected government institutions.

Users have been in touch with us with a series of requests for new features. Visualization creators, for instance, would like to have a wider variety of maps on the site while bloggers would like to be able to embed interactive visualizations on their blogs. Visualization builders would like to add new visualization techniques to the site. Overall, feedback about Many Eyes has been positive and the variety of visualization applications—from playful gaming to serious data analysis—seems to attest to the value of the site to different users.

7 Conclusion and Future Work

We have described the Many Eyes web site, which provides a set of visualization creation and publishing tools to a large potential audience. The architecture of a site that aims to be useable by millions is nontrivial, and we have discussed the many choices and tradeoffs in the current design. In some cases these design choices involve simplifications to or reordering of the standard visualization pipeline—allowing data transformation in the context of creating a visualization, for instance. We also have described how flexible collaborative capabilities are woven into all of the visualization components, as in our selection and bookmarking model. Finally, we have described how the site exists as part of a social ecosystem, with discussions on blogs providing a significant amount of attention and activity surrounding Many Eyes.

Future work will focus on three main areas. First, the site could benefit from a stronger set of community tools. In particular, as the collection of data sets grows, organization and quality control will become increasingly important. It would be beneficial to have mechanisms for the user community to do some of the work in organizing and filtering.

There are also natural extensions to the data model of Many Eyes. Some elements are logistical: the site would clearly gain from tools that allowed easy export from other site. More broadly, the capability of reading data from external sources would open up the possibility of "live" visualizations and the construction of composite visualization applications. All of these possibilities raise interesting questions around versioning and collaboration.

A third natural future direction is to augment the visualizations themselves. The current annotation/pointing scheme is extremely simple, and could be extended in a number of directions. For example, the site could exploit the fact that the annotations are tied to the data to allow inter-visualization brushing and selection. More generally, it would be natural to experiment with other types of visual metadata. For instance, as in [29] one could provide visual indications to show which elements of a data set of have been examined closely.

Finally, it may make sense to use Many Eyes as a platform for rapid user testing of new visualization techniques. Conceivably the site could offer an API so that third-party developers and researchers could test their own offerings. Tests could consist of simply putting new components up and watching whether they are used, or could use more sophisticated methods. For instance, in the graph visualization component users may rearrange the graph layout—and they frequently do. Would it make sense to look at these human-created layouts to deduce implicit aesthetic criteria? Such approaches seem promising, and point to the research benefits of a broadly available visualization site with a large user base.

References

[1] Adai, A. et al., "LGL: creating a map of protein function with an algorithm for visualizing very large biological networks", *J Mol Biol.* Vol 340(1), pp. 179-90, 2004.

[2] Anupam, V., Bajaj, V., Schikore,D., Schikore, M. Distributed and collaborative visualization. Computer, vol. 27, no. 7, pp. 37-43 1994.

[3] Bridgman, S and Tamassia R. "The Graph Drawing Server", Graph Drawing 2001, Springer LNCS 2265, pp 44, 2002.

[4] Brodlie, K.W., Duce, D.A., Gallop, J.R., Walton, J.P., & Wood, J.D. "Distributed and Collaborative Visualization", Blackwell Pub., 1981.

[5] Chartall, http://www.chartall.com, retrieved 03-30-2007.

[6] Clark, H.H. and Brennan, S.E. "Grounding in Communication". In: Perspectives on socially shared cognition, American Psychological Association, 1991.

[7] DabbleDB, http://www.dabbledb.com, retrieved 03-30-2007.

[8] Data360, http://www.data360.org, retrieved 03-30-2007

[9] DataPlace, http://www.dataplace.org, retrieved 03-30-2007

[10] Flickr, http://www.flickr.com, retrieved 03-30-2007.

[11] Google Earth, http://earth.google.com, retrieved 03-30-2007.

[12] Heer, J., Viégas, F.B., and Wattenberg, M. "Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization", In *Proc. of CHI*, 2007.

[13] Hill, W.C. and Hollan, J.D. "Deixis and the Future of Visualization Excellence", In *Proc. of IEEE Visualization*, 1991.

[14] Livny M. et.al, "DEVise: integrated querying and visual exploration of large datasets", In *Proc. ACM SIGMOD'97*, pp 301-312, 1997.

[15] Mackinlay, J. "Automating the Design of Graphical Presentations of Relational Information", *ACM Transactions on Graphics*, Vol. 5 ,No. 2 pp 110 – 141, 1986.

[16] New York Times: Faces of the Dead. http://www.nytimes.com/ref/us/20061228_3000FACES_TAB1.html, retrieved 03-30-2007.

[17] North, C. et al, "Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table datasets", *Information Visualization*, Vol 1, pp 211-229, 2002.

[18] Quickbase, http://www.quickbase.com, retrieved 03-30-2007.

[19] Roth S. et al., "Towards an Information Visualization Workspace : Combining Multiple Means of Expression", *Human-Computer Interaction Journal*, Vol. 12 (1&2), pp 131 – 185, 1997.

[20] Roth,S. ''Visualization as a Medium for Capturing and Sharing Thoughts,'' Capstone in *Proc IEEE InfoVis 2004*, p. xiii, 2004.

[21] Spotfire, http://www.spotfire.com, retrieved 03-30-2007.

[22] Swivel, http://www.swivel.com, retrieved 03-30-2007.

[23] Tableau, http://www.tableausoftware.com, retrieved 03-30-2007.

[24] Tang, D et al., "Design Choices when Architecting Visualizations", *Proc IEEE InfoVis'03*, pp 41-48, 2003.

[25] TheyRule, http://www.theyrule.net, retrieved 03-30-2007.

[26] Viégas, F., boyd, d., Nguyen, D., Potter, J. & Donath, J. Digital *Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments*. In Proc. of HICSS-37, 2004.

[27] Viégas, F. and Wattenberg, M. "Communication-Minded Visualization: A Call to Action", *IBM Systems Journal*. 45(4), 2006.

[28] Wattenberg, M. "Baby Names, Visualization, and Social Data Analysis", In *Proc .IEEE InfoVis'05*, pp 1-7, 2005.

[29] Wattenberg, M. and Kriss J., "Designing for Social Data Analysis", *IEEE TVCG*. 12(4), pp 549–557, 2006.

[30] YouTube, http://www.youtube.com, retrieved 03-30-2007.