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: \{\text{hierarchy : T+}, \text{size : N}, \text{color : N}\} and \{\text{Xaxis : N}, \text{Yaxis : N}, \text{label : T}, \text{[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 \text{N} and \text{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.

#### 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. \text{N} or \text{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—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 reuse 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 visualizations 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.
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 Table 2. 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.

<table>
<thead>
<tr>
<th>Visualization Component</th>
<th>Percentage of use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pie Chart</td>
<td>15%</td>
</tr>
<tr>
<td>Network Diagram</td>
<td>12</td>
</tr>
<tr>
<td>Tag Cloud</td>
<td>11 (on site for only one month)</td>
</tr>
<tr>
<td>Treemaps</td>
<td>10</td>
</tr>
<tr>
<td>Bar Chart</td>
<td>9</td>
</tr>
<tr>
<td>Line Graph</td>
<td>9</td>
</tr>
<tr>
<td>World Map</td>
<td>8</td>
</tr>
<tr>
<td>Scatterplot</td>
<td>7</td>
</tr>
<tr>
<td>US State Map</td>
<td>7</td>
</tr>
<tr>
<td>Stack Graph for Categories</td>
<td>4</td>
</tr>
<tr>
<td>Block Histogram</td>
<td>4</td>
</tr>
<tr>
<td>Stack Graph</td>
<td>3</td>
</tr>
<tr>
<td>Pie Chart</td>
<td>1</td>
</tr>
</tbody>
</table>

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-occurrences 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 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-generated 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