Democratizing Data Visualization

Larry Hardesty, MIT News Office

Study examines use of 'Exhibit' tools in creating interactive data visualizations.

In 2007, members of the Haystack Group in MIT's [1] Computer Science and Artificial Intelligence Laboratory released a set of Web development tools called "Exhibit." Exhibit lets novices quickly put together interactive data visualizations, such as maps with sortable data embedded in them; sortable tables that automatically pull in updated data from other sites; and sortable displays of linked thumbnail images.

In April, at the Association for Computing Machinery's Conference on Human Factors in Computing Systems, Haystack members will present an in-depth study of the ways in which Exhibit has been used — with ramifications for the design of data-visualization tools; data-management software, such as spreadsheets; and Web-authoring software, such as content management systems.

The study also indicates ways in which websites could better gauge the effectiveness of the visualizations they publish. "Imagine if The New York Times was able to track how well you understood a visualization, or how you used it, rather than simply how much time you spent on it," says Ted Benson, a graduate student in electrical engineering and computer science and co-author of the new paper, along with professor of computer science and engineering David Karger. "That could help them design more engaging data displays and maybe even help uncover new stories in the data you didn't know were there."

In their study, Benson and Karger performed a series of successively more tightly focused analyses. First, they examined the design decisions that characterize 1,897 pages built using Exhibit — "Exhibits," in the application's parlance. Then they studied the automatically generated access logs of the 100 most popular Exhibit sites. The authors of 24 of those sites also allowed the researchers to install software that tracked the individual mouse clicks executed by site visitors —

200,000 interactions in all. Finally, Benson and Karger interviewed the developers of 12 Exhibit sites about their experiences with the tool.

Untapped Market

Karger believes the fact that so many people — scientists posting research findings, administrators of commercial websites, journalists — have gravitated to Exhibit is telling in itself.

“There are 1,900 websites that have chosen to build an Exhibit,” Karger says, “which is actually a pretty remarkable stretch given that this is a research project with no technical support and no decent documentation. In my mind, what that says is that there is a need out there that is not being met. I believe the need centers on achieving full authorial control over the design of your interactive visualizations without having to become a programmer.”

The new paper, Karger adds, is an attempt to investigate Exhibit’s utility more rigorously. Exhibit is a “declarative” language, like HTML, not an “imperative” language, like Java, Karger explains. That means that programs written in Exhibit simply describe how existing classes of graphical elements will be deployed on screen and which data sets they’ll draw from. Exhibit doesn’t enable the programmer to create new functions from scratch.

That limits its versatility but, Karger argues, makes it much easier to use. The same goes for another aspect of Exhibit’s design: An Exhibit page, or multiple pages on the same site, can feature different visualizations of the same data. But the data must be stored in a single location, which each of the visualizations accesses independently. Visualizations can’t refer to each other.

In combination, these design decisions mean that novices can quickly build their own pages simply by cutting and pasting other people’s code. They just need to change the names of the data files the code refers to — and they don’t need to worry about broken links to other visualizations.

The Numbers Speak

The new study offers some strong evidence that this is exactly what Exhibit users do. The data that Exhibit pages display can be stored in a variety of formats, including Excel spreadsheets and comma-separated text. But 69 percent of Exhibit sites instead use the more obscure JavaScript Object Notation format, or JSON.

Several interview subjects explained that JSON was the format in which data were stored in most of the examples on the Exhibit website — and to produce their sites, they had simply cut and pasted code from existing Exhibits. The prevalence of JSON suggests that many other Exhibit users are doing the same thing.

Exhibit’s declarative design also made it easy to analyze users’ interactions with Exhibit visualizations. Since every mouse click invokes an existing computational module, rather than executing a new computation from scratch, describing usage

patterns is simply a matter of logging which modules are invoked when.

One characteristic of Exhibit sites that surprised Benson and Karger: While most developers used spreadsheets to create their data, their visualizations often exploited more complex relationships among data than spreadsheets are intended to handle. Some 32 percent of Exhibits used “multivalued tables,” in which a single slot — the equivalent of a cell in an Excel file — contained more than one value.

Twenty-seven percent used “graphs,” which capture relationships among data elements, such as which members of a user’s social network are also linked to each other.

The researchers conclude that, since it seems natural even to novice Web developers to organize their data in these more sophisticated ways, spreadsheet designers should offer tools that make it easier for them. Exhibit users found ad hoc techniques for representing more complex data structures in spreadsheets, but in the process, they gave up some of the spreadsheets’ core functionality. For instance, an Excel user can represent a multivalued table by entering comma-separated lists in a single cell, but those lists aren’t sortable, as spreadsheet data is intended to be.

Case Study

Jon Crump, a medieval historian and educational consultant, is one of the 12 Exhibit users interviewed for the new study. He has used Exhibit in his own work, but he also taught history classes at Seattle Central Community College and Pacific Lutheran University in which he asked his students — humanities students who didn’t necessarily have any programming experience — to build their own Exhibits.

“The majority of the class didn’t have any trouble doing it,” Crump says. “Some folks just don’t have the patience or the attention to detail to bother with carriage returns or commas. They didn’t get that right and didn’t have sufficient interest to pursue it. But there’s nothing particularly hard about it.”

In his own work, Crump is interested in Exhibit because it could give historians a convenient, intuitive way to access the wealth of research conducted by their peers that never makes it into print. “Exhibit intrigues me not just because it’s a convenient way to visualize and display information, but because it can serve as a way to visualize and store information in a form that is interoperable with other people’s systems around the world,” Crump says. “So my data can very conveniently be merged with your data, and the result can be displayed, too, using Exhibit.”

For more information visit http://web.mit.edu [1].

Source URL (retrieved on *04/25/2015 - 3:48am*):
http://www.wirelessdesignmag.com/news/2014/03/democratizing-data-visualization

Links:
[1] http://web.mit.edu