Distr.
GENERAL

WP.17
15 May 2012

ENGLISH ONLY

UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE (UNECE)
CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE EUROPEAN UNION (EUROSTAT)

ORGANISATION FOR ECONOMIC COOPERATION AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE

Meeting on the Management of Statistical Information Systems (MSIS 2012)
(Washington, D.C., 21-23 May 2012)

Topic (iii): Innovation

Creating Highly Interactive Websites for the Dissemination of Statistics

Supporting Paper

Prepared by Mark Elbert, U.S. Energy Information Administration, USA

I. Introduction

1. The first browser war was about whether proprietary or open standards would dominate the Internet, and open carried the day. The second and ongoing browser war is about speed.

2. When Google launched its much faster Chrome web browser in 2008, it challenged other browser manufacturers to keep up. And they did. Internet Explorer 10 executes JavaScript, the programming language of the web, 30 times faster than Internet Explorer 8. With each successive version, all the major browser get faster and more capable. It does not matter which vendor emerges from this second browser war with the largest market share, because the competition among manufacturers has endowed the world with an infrastructure of high-performance, standards-based web browsers. As a result, a new class of interactive web applications for publishing, visualizing, and analyzing statistics is possible.

3. As browsers' JavaScript execution engines improved, open-source JavaScript libraries have evolved and matured, too, allowing web programmers to add capabilities by including libraries and only writing custom code to add value.

4. The U.S. Energy Information Administration (EIA) commenced public testing of a new Electricity Data Browser (www.eia.gov/beta/enerdat) in March 2012, a web product for the dissemination US electricity statistics, that takes full advantage of this new web environment. This paper will share the lessons learned from this approach.

II. CaseStudy: EIA's Electricity Data Browser

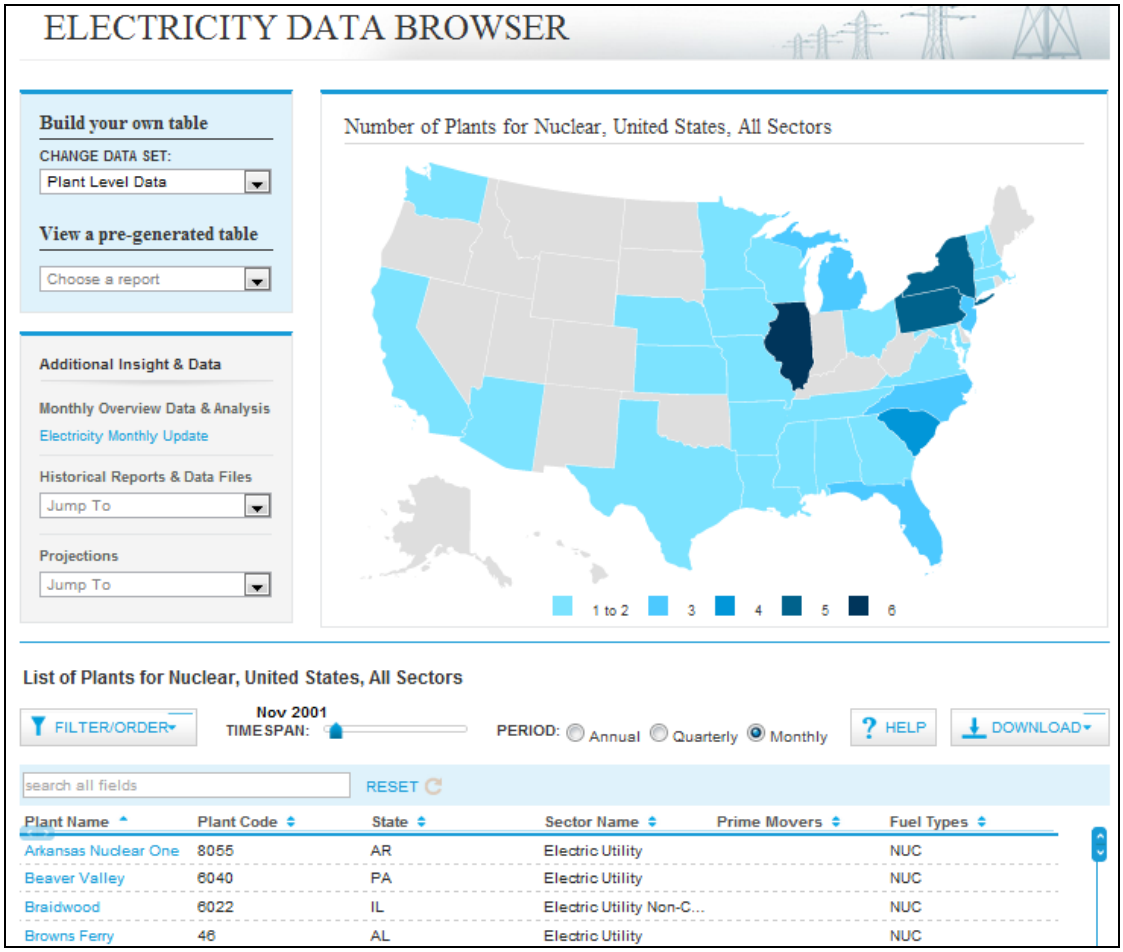

Figure 1: Screenshot of Electricity Data Browser

A. Basic architecture of an interactive data tool on the web

5. The Electricity Data Browser (shown in figure 1 above or visit www.eia.gov/beta/enerdat), written by two skilled web developers in just under 10 weeks, allows a user to view standard or custom reports to find and chart any of approximately 80,000 aggregate time series by month, quarter, or year. From any of the aggregate values, users may drill down to electricity generation plant maps and data underlying aggregate statistics. Exposing another 73,000 time series, these plant-level data show time series as detailed as the fuel consumed by a particular steam turbine at a particular plant.

6. The application architecture has two primary components, a client layer and a data layer. The client layer is composed of HTML and JavaScript code that runs in the user's web browser. It displays data and controls via HTML, and makes requests to the data layer using AJAX (Asynchronous JavaScript and XML) techniques. The data layer is composed of server code running on EIA's web servers that processes requests for data from the client layer running in users' web browsers by fetching the data from EIA's databases and returning datasets to the client layer. In effect, the data layer is an Application Programming Interface (API) that is valuable in and of itself, as it could be used by third-party websites and mobile applications to access EIA's data.

B. Custom programming code vs. third-party code in the Electricity Data Browser

7. The client layer consists of custom JavaScript and HTML code written by EIA programmers and a number of third-party JavaScript libraries. The data layer consists entirely of custom server code. The following table shows the breakdown as measured by size of source code files in kilobytes.

Data layer	
custom server code	12%
Client layer	
existing EIA JavaScript modules	7%
custom JavaScript and HTML code	17%
Cascading Style Sheets	3%
third-party JavaScript libraries	61%

Table 1: Electricity Data Browser source code breakdown

8. The key to the speed with which this complex application was developed lies in the extensive use of third- party libraries and existing modules, comprising 68% of the total source code as judged by the size of source-code files. These concise and well-tested libraries provided complex functionality and would require a programming effort far exceeding 68% of the total project, if written from scratch. Therefore, a more detailed examination of these third-party libraries and their characteristics is given in Table 2 below:

Name	**Description**	**License**	**Version**	**Size**	**First published**	**Dependencies**
jQuery	helper library; simplifies JavaScript coding	open-source	v1.7.1	92kB	Jan 2006	-
jQuery UI	user interface control	open-source	v1.8.7	202kB	Sep 2007	jQuery
Highstocks	charting library	commercial ($80)	v1.1.4	117kB	Oct 2011	jQuery
jVector Map	map infographics lib	open-source	v0.1	23kB	Feb 2012	jQuery
USA map definition	USA map definition	open-source	v0.1	47kB	Feb 2012	jVector Map, jQuery
SlickGrid	fast, scalable grid component	open-source	v2.0	12kB	Mar 2009	jQuery, jQuery.event. drag
JQuery Event Drag	extends jQuery; used by Slick	open-source	v2.0	5.1kB		jQuery
jQuery mousewheel	simplifies mousewheel programming	open-source	v3.0.6	3.5kB	Apr 2007	jQuery
Loadmask	creates wait screen effects	open-source	v0.4	4.1kB	Jun 2009	jQuery
qTips	help tips	open-source	v1.0	22kB	Apr 2010	jQuery
Curl	loads library as needed	open-source	v0.6.1	14.2kB	Jan 2011	-

Classy	objectoriented extension for JavaScript	open-source	v1.4	4.99kB	Apr 2010	jQuery
Crossroads	pattern matching framework	open-source	v0.7.1	5kB	Apr 2011	-
Hasher	url hash management	open-source	v1.1	2.8kB	Aug 2011	-
Mustache	logic-less template system	open-source	v0.5	14kB	Oct 2009	-
SimpleModal	creates modal panel (greys rest of screen)	open-source	v1.4.2	23kB	Oct 2007	jQuery
JSON2	provides JSON support for obsolete browsers	open-source	v1.0	5kB	Nov 2010	-
PureMVC	model-view-controller framework	open-source	v1.1	7kB	Nov 2010	-
js-Signals	event framework	open-source	v0.7.4	3.1kB	Nov 2010	-
Kizzy	cross-browser localStorage API	open-source	v1.0	4.6kB	Feb 2011	-

Table 2: Third-party libraries used by the Electricity Data Browser

9. The point of listing the libraries used in this case study in such detail is to highlight three overarching facts of the current state of web application development.

10. First, it is apparent that this new web environment is young and evolving quickly. As a reference point, the Electricity Data Browser project was designed and written in January through March of 2012. Most of the libraries used by the Electricity Data Browser during development were very new, judging by both their first publication dates and by their version numbers.

11. Second, one cannot help but note the variety of lightweight, open-source code libraries that can be easily found and downloaded for free or a very low cost on the Internet. This allows skilled web application developers to quickly create customized application frameworks, with just the capabilities needed, on which to build advanced data dissemination web sites. This is very different from the typical commercial development environment, such as those provided by Microsoft or Adobe, which are more complete out of the box, but lack the flexibility of choosing and adding the exact features your web application needs.

12. The third and final take-away from the above table is the relative maturity of the jQuery JavaScript helper library and its frequency with which it is required by other libraries. While this paper is not meant for web designers, one of its few technical recommendations is that jQuery be considered a standard include across your entire website. jQuery is the emerging standard. It really lives up to its slogan: “The Write Less, Do More, JavaScript Library.”

13. A good example of all three of these observations in this case study is the advanced help system for the Electricity Data Browser (see figure 2 below). The help tips are displayed using the qTips open-source library. Because of the generally improved quality of documentation of open-source projects, the EIA development team was able to include the qTip library in the project, write the help tip texts, and write the code for the help button’s click event in under eight developer hours. The qTips library was first published just two years ago and, like so many open-source libraries, requires that the jQuery library be included as well.

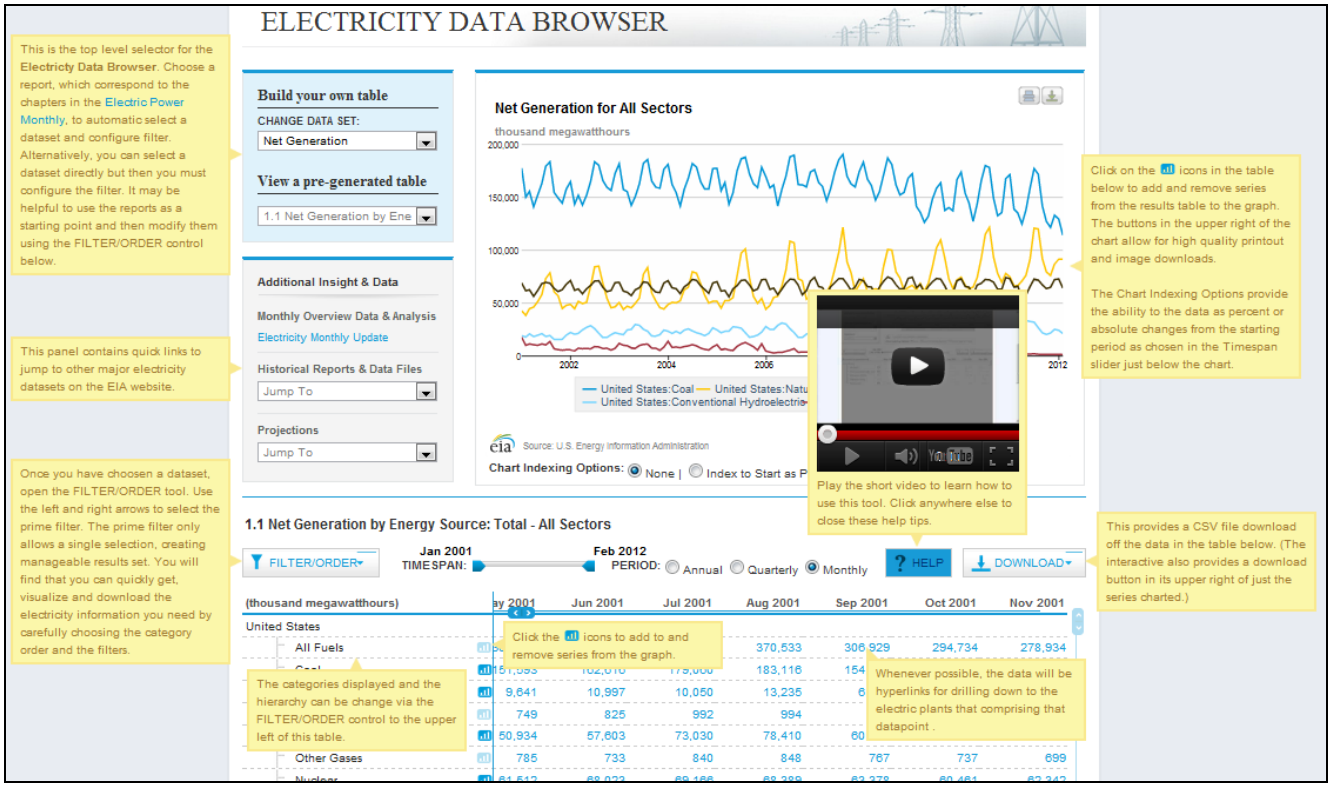

Figure 2: Screenshot of help feature

II. Pros and cons of creating highly interactive web applications to disseminate statistical information

14. There are many advantages (pros) to this development methodology, including:

(a) Faster development cycles.
(b) Data layer is lightweight and can be easily ported to a different server technology if needed.
(c) Data layers may be published as Application Programming Interfaces (APIs), meeting other customer needs and promoting open data and open government efforts.
(d) The variety of third-party libraries allows the development team to focus their skills and creativity on problem-solving instead of building the plumbing.
(e) The third-party libraries are mostly free or very low-cost open-source projects.
(f) May reduce publishing cost if large number of files and web products are replaced with web applications.
(g) Improve the findability of statistics by offering a single interface to browse, filter, and find information that would otherwise be spread out among multiple publications.
(h) Enables organizations to create the kind of highly interactive, rich user experiences web users are coming to expect.

15. The disadvantages (cons) include:

(a) Requires significantly higher skill sets than traditional HTML coders and web server developers typically possess.
(b) Too many third-party libraries can make maintenance a challenge.
(c) Third-party libraries without strong developer communities may be buggy or lack solid documentation.
(d) Versioning and updating multiple third-party libraries can be a challenge.
(e) Cross-browser support, especially for legacy browsers, can be very challenging.

16. The disadvantages listed above c be mitigated through effective information systems management. For example, training and hiring efforts can build development teams with the needed skills. Management involvement in the selection of a standard set of libraries can ease maintenance challenges by carefully considering each new library the developers propose using, taking into account its maturity and the business needs (i.e. do not let your developer add third-party library after third-party library because it is "cool"). Developing and implementing a clear versioning strategy can prevent library version upgrades from breaking seemingly random parts of your website. Determining which browsers your web applications will support and informing website visitors using antiquated browsers that they need to upgrade to any of a list of free browsers that your site does support will greatly simplify the cross-browser support issue. This is because current browsers are not only faster, they are also more standards-compliant.

17. With the disadvantages properly managed and mitigated, the advantages of this development methodology are too great to ignore. In the case of EIA's Electricity Data Browser, in 10 weeks a very small team was able to create and publicly test a web application capable of replacing and significantly improving 10 years' of EIA's Electricity Power Monthly, its Electric Power Annual, and its electricity generation plant data publications. It added user controller grouping and filtering, quarterly series, and interactive graphing of all times series in nearly any combination of aggregate data. All aggregate can be drilled down to the underlying electricity generation plant level, complete with maps and graphing of individual plant time series, such as the amount of coal consumed in Boiler #2 or the electricity generated by Turbine #1 at a particular plant. This is an impressive amount of functionality to be designed, coded, and debugged in just 10 weeks by two developers with the occasional help of a web graphics designer.

18. Most importantly, building highly interactive web applications for the dissemination of statistics is no longer a choice. Just as Gmail eclipsed Hotmail, users will gravitate to websites offering the most engaging and interactive user experience,. If our respective organizations' websites are to remain relevant in the face of increasingly sophisticated websites that repackage and republish public data, we must build highly interactive web applications with sophisticated visualization tools. I believe that our websites—not just our data--are important and irreplaceable because as statistical organizations, we care about and understand our statistics and the underlying data's accuracy and applicability in ways the republishing websites do not.

III. Creating highly interactive web applications: A how-to

19. Organization may not currently have the capabilities to tackle such challenging projects. In such cases, it is critical to build organization capacity by choosing small projects at first that add some basic yet useful interactive features. Strategic planning of initial projects will not only build organizational capacity; successful first projects will build support within your organization for further efforts.

20. Adding basic interactive charting to existing web pages is a good place to start, since it will create important capabilities that meet important needs for statistical organizations. Even the most basic interactive charting will require an organization to choose a charting library and a helper library. It will require an organization to settle on the style and theming of the interactive charts it will use. Gaining experience configuring and using these libraries are important first steps.

21. Many statistical organizations, EIA included, publish a great deal of data in the form of HTML tables. Usually, each table contains a number of related data series, which many users would like to visualize. EIA has created a library called Autochart which does just that: It adds selection controls to HTML tables allowing users to select series to visualize and creates high quality interactive charts and downloads. This is done using JavaScript running in the client's browser without the need for additional back-end systems. The Autochart library is available at no cost and with no copyright restrictions from www.eia.gov/developer. Using the Autochart library as-is, or adapting it to your organizations needs will give an organization's developers experience using Highcharts, which is the most advanced JavaScript charting library available. Developers will also get experience using jQuery, which has become the standard

web helper library. (Sharing code, as EIA is doing with Autocharts, is also an important part of an organization's commitment to this new and open development methodology.)

22. Regardless of which projects an organization initially selects, the crucial first step in creating highly interactive web applications is a commitment to building staff expertise and raising organisational awareness of the benefits of this approach to web dessimination.

III. Conclusions

23. The new web environment of fast JavaScript engines and other improvements in the current generation of web browsers and loads of free or low-cost JavaScript libraries enable the rapid design and development of rich, highly interactive web application as vehicles for the dissemination of statistical information. Like all web users, users of statistical websites are coming to expect the rich user experience such web applications provide. For the websites of statistical organizations to remain relevant, our organizations must develop the expertise to create interactive web applications with rich user experiences.