A Web-based Data Cube Visualization Ecosystem Architecture

Curran Kelleher *
University of Massachusetts Lowell

Abstract

What would the world be like if anyone with an Internet connection could easily use any interactive visualization technique to explore and present any publicly available data sets? Education would be revolutionized, policy making could be mostly data-driven, journalists could present facts to their audiences directly, and the general public could incorporate visualization into their daily decision making processes. This doctoral thesis explores how this goal can be accomplished by harnessing the power of mass-collaboration, the Semantic Web, and modern Web graphics technology.

Index Terms: D.2.3 [Software]: Software Engineering—Coding Tools and Techniques E.1 [Data]: Data Structures—Distributed Data Structures H.2.1 [Information Systems]: Database Management—Logical Design H.5.3 [Information Storage and Retrieval]: Information Interfaces and Presentation—Group and Organization Interfaces

1 Introduction

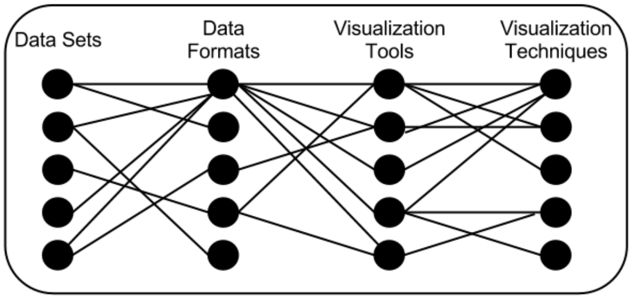
Why aren't the interactive information visualization techniques we see in academic publications all available for use by the general public for exploring publicly available data sets? The fundamental purpose of visualization is "to augment human cognition" [8], but in reality it is only augmenting the cognition of a select few, not humankind. This dissertation aims to change that by establishing an infrastructure for mass-collaborative evolution of a global commons of

- public data sets,
- visualization software, and
- user-generated visualizations.

The general problem addressed is that in practice there are many barriers between data sets and visualization techniques. Each data set is only available in a particular set of formats. Each visualization tool (system) is only able to consume a particular set of data formats. Each visualization tool typically supports a small subset of visualization techniques. Once a visualization is created, dissemination of that visualization can only be accomplished by the means provided by the tool used to create it.

The current landscape of data sets and visualization tools has the property that for any pair of data set and visualization technique selected, the visualization author needs to do significant research to find out

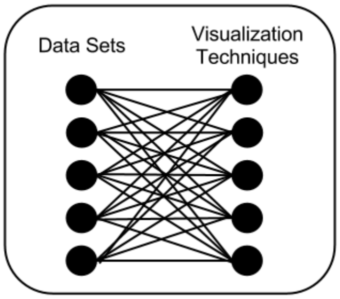
- Which formats are the data set available in?
- Which visualization tools support available formats?
- Which tools support the desired visualization technique?



The Reality

- Which tools support *both* one of the formats the desired data is available in *and* the desired visualization technique?

This makes it extremely difficult for a non-specialist to visualize data. Also, it is impossible to visualize a given data set with a given visualization technique if there are no visualization tools that support one of the available data formats *and* the desired visualization technique, unless you are willing to build your own data conversion or visualization tools.

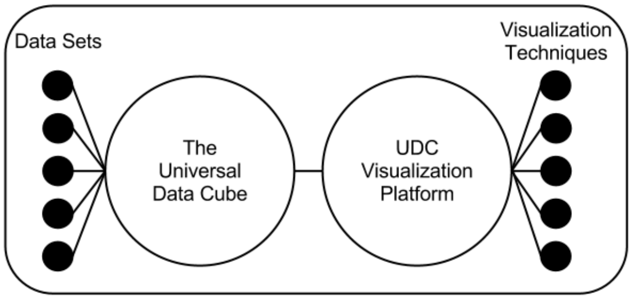


The Ideal

Ideally, users could simply choose a data set and a visualization technique, and an interactive visualization of that data set would magically appear. Ideally, the result could be published in interactive form (rather than as a static image), and viewers of the resulting interactive visualization tool could easily create and publish derivative works that, for example, highlight different aspects of the data, zoom to a different level of detail, tweak the visualization parameters, or even view a different data set. This dissertation investigates the question "How can this ideal be realized?"

Just as the goal of Wikipedia was to establish a repository of all human knowledge, the goal of this dissertation is to establish a repository of all public data sets and implementations of all interactive visualization techniques. Just as Wikipedia imposes simplifying assumptions (e.g. a common markup language and article guidelines) in order to realize their seemingly impossible goal by leveraging mass-collaboration [11], the solution introduced in this dissertation imposes the following simplifying assumptions to realize our seemingly impossible goal by leveraging

*e-mail: ckellehe@cs.uml.edu



Our Solution

mass-collaboration:

- Data sets are published in the *Universal Data Cube* (UDC), a subset of the Semantic Web that uses the RDF Data Cube Vocabulary [10] and conforms to interoperability guidelines introduced in the dissertation.
- Visualization software is created within the *UDC Visualization Platform*, a JavaScript software repository featuring browser-based code editing, automated dependency management, support for embedding applications in Web pages, an application state management framework, libraries for accessing UDC data, and libraries supporting interactive visualization applications.

If all public data sets were imported into the UDC and all interactive visualization techniques (with effective configuration interfaces) were implemented in the UDC Visualization Platform, the ideal of publicly usable visualization tools will be realized. Users could simply choose a data set and a visualization technique, and an interactive visualization of that data set would magically appear. The result could be published in interactive form within Web pages, and viewers of the resulting interactive visualization tool could easily create and publish derivative works.

The task of importing all public data sets and implementing all interactive visualization techniques is too large a task for any single person or research institution to accomplish, and will likely never be completely realized. Therefore the focus of this dissertation is to introduce a substrate that facilitates mass collaboration in building a global commons of public data sets, reusable software libraries for interactive information visualization, and interactive visualization applications. Our hope is that over several years of mass-collaborative effort from visualization enthusiasts around the globe, most well known public data sets and interactive visualization techniques will exist within our architecture.

Centralized Web-based hubs of data sets and software implementations are introduced as part of the dissertation, but the design of the system does not constrain the data or software to the centralized systems. Any data published by any server on the Web using the UDC interoperability guidelines can be consumed by tools built to work with UDC data. Software created in our centralized development system can easily be extracted into independent JavaScript projects and retain all data access capabilities. These design aspects are key factors in the longevity and long term effectiveness of our contribution.

2 Related work

Our system architecture covers domains that can be categorized as Web-based collaborative software development, collaboration and history, Web-based information visualization, metadata and distributed data sets, and general purpose visualization systems (particularly for visual OLAP).

2.1 Web-based Collaborative Software Development

Several projects introduce a Web-based software development platform. Goldman et al. introduced *Collabode*, a Web-based collaborative IDE for Java development supporting real-time collaborative text editing [14]. Sites including JSFiddle, JSBin, CSSDesk and IDEOne feature Pastebin-like saving and running of source code. Google's "Code Playground" includes browser-based code editing for trying out their APIs. The OpenProcessing.org and SketchPad.cc projects feature a showcase of Processing sketches and some IDE-like features. Web-based IDEs that emulate desktop IDE features include Cloud9 IDE, Kodingen and CodeRun. GitHub is a commercial service for managing Git repositories, supporting in-browser code editing and saving. None of the Web-based coding tools we surveyed directly support publication and use of public modules or automated dependency management.

Many popular platforms have introduced central Internet-based repositories for automating dependency management for popular libraries. The associated tools often support dependency management for local assets as well. Repositories include the Node Package Manager Registry for Node.js, JSAN and Scripteka for JavaScript, the Maven and SpringSource repositories for Java, RubyGems and Rubyforge for Ruby, CRAN for R, PyPI for Python, CTAN for Latex, and Pear and Pecl for PHP. Most of these handle automatic dependency management, updating of content, and user-contributed content. None of the repositories we surveyed have a Web-based editing environment, and all require locally installed tooling in order to function.

2.2 Collaboration and History

Asynchronous collaboration for Web-based visualizations has been explored in the *sense.us* project, which supports view sharing, discussion, graphical annotation and social navigation [19]. The ManyEyes project allows users to import their own data tables and publish visualizations using a set of pre-packaged visualization tools [31]. Heer et al. surveyed asynchronous collaboration for visual analytics and articulated high level design considerations [16], and also introduced a history model for the Tableau system [18]. Prefuse is a Java visualization toolkit that includes event logging for monitoring and recording of user events, but is not a general solution for history and has not been extended to support collaboration [17]. VisTrails is a system for scientific visualization featuring advanced history management functionality [7].

2.3 Web-based Information Visualization

Google Fusion Tables is a cloud-based data table management service that enables developers to query the data via a Web API [15]. Google Chart Tools introduces a JavaScript visualization library supporting predefined visualization types and composition of dashboards. Protovis [5] and D3 [6] are toolkits that support developers in creating interactive Web-based visualizations, but do not provide support for a higher level system such as the user interface enclosing the visualizations, linked views, or data set selection. Gapminder [25] is a Web-based socioeconomic indicator data visualization tool featuring a scatter plot view, a time line view, and a map view, based on Adobe Flash technology.

The Weave project is a general purpose Web-based interactive visualization environment featuring multiple linked views, brushed selection and probing, synchronous collaboration, asynchronous collaboration in the form of view sharing and history sharing, and history navigation [2][3]. Weave is based on Adobe Flex technology, and uses tables as its data model. Weave supports integration of multiple data sets provided that the keys (record identifiers) match, or a mapping between corresponding keys is configured.

2.4 Metadata and Distributed Data Sets

The Resource Description Framework (RDF) was designed to support representation and query of distributed data sets as well as unambiguous interpretation of data using ontologies [20]. A vocabulary has been introduced by Cyganiak et al. for representing data cubes in RDF called the RDF Data Cube Vocabulary, which is on track for becoming a W3C recommendation [10]. Priebe et al. introduced an approach for building an enterprise-wide knowledge portal supporting integration of multiple OLAP data sources using RDF [24].

2.5 General Purpose Visualization Systems

The problem of designing and implementing general purpose information visualization systems has been faced by many researchers. XmdvTool [32], Spotfire [1], XGobi [29], GGobi [30], Advizor [12], and the Universal Visualization Platform [13] are interactive multidimensional (table-based) visualization systems designed for desktop environments that support multiple visualization methods with interaction techniques including dynamic queries, brushing and linking. DEVise [21] is a visualization system focused on visualizing the contents of relational databases.

Researchers have explicitly considered the case of interactive visualization for OLAP cubes. Mansmann et al. introduced the term *Visual OLAP*, and defined a visual exploration framework that surveyed OLAP operation interactions and visual metaphors for data cubes [23] [9]. Mansmann et al. also introduced novel means of visualizing data cubes including hierarchical heat maps and decomposition trees [22]. Polaris, the predecessor to Tableau, is a visualization system based on the hierarchical data cube model [26] [28] [27].

3 Architecture Overview

Our architecture is divided into four components:

- The Universal Data Cube (UDC) a data representation framework supporting distributed heterogeneous hierarchical data cubes, and
- CodeHub - a Web-based hub of participatory culture for creating and publishing interactive graphics software,
- Application State Historian (ASH) an application state management framework supporting collaboration (both synchronous and asynchronous) and history navigation,
- The UDC Visualization System - an interactive visualization platform with multiple linked views and rich interactions built on top of the first three components.

3.1 Web-based Development Platform for Interactive Graphics Software

The first component of our architecture, called CodeHub, is a Web application that provides an in-browser source code editing and publishing environment for interactive graphics applications built with HTML5 and JavaScript (using CommonJS modules). Applications can be run full-screen by navigating to a URL. Applications can be embedded in Web pages using iFrames. CodeHub stores and publishes all software versions, so an application link will always resolve to the exact same software, and embedded applications will always retain the exact same behavior. CodeHub has the potential to evolve into a global commons of JavaScript and HTML code.

Typically, systems or prototypes for which information visualization research papers are published are not available for readers to try themselves or modify. When research prototypes or systems are built using CodeHub, they will be runnable by all readers of the research papers, and available for modification at the source code level so derivative works may be produced by other researchers, and the techniques may be adapted to work with other data sets.

3.2 Application State Historian

The second component of our architecture, called Application State Historian (ASH), provides solutions for synchronous collaboration, asynchronous collaboration, and session history navigation. The core of ASH is a managed application state model based on a versioned data structure and corresponding history graph. Applications that use ASH for application state management will automatically gain collaboration and history functionality. ASH is partitioned between a JavaScript client library providing an application state management API, and a server component that stores application history graphs and manages real-time communication between clients.

Whenever insights are generated by using visualization tools, it is essential that users of the system be able to re-create the insight generation process [18]. This is enabled by the history navigation and replay functionality of ASH. Imagine someone has created a visualization and published it to the Web, what if readers were able to tweak the visualization and re-publish the result? This is made possible by the asynchronous collaboration feature of ASH.

Interactive visualizations can be used in a classroom setting, for example viewing historical demographic data in a history or economics class, where the teacher and students all share the same view of the visualization. The synchronous collaboration feature of ASH gives students the ability to “take control” of the input such that the student’s actions were immediately replicated in the systems of all other students and the teacher. This feature functions across the Internet, so could also be used in settings such as online courses or international corporate meetings.

3.3 The Universal Data Cube

The third component of our architecture, called the Universal Data Cube (UDC), is a data representation and publication methodology based on the RDF Data Cube Vocabulary [10] and additional interoperability guidelines enabling precise semantics and distributed data sets. The UDC enables many data sets from multiple sources to be integrated together and visualized as a coherent whole. The UDC data model provides a powerful representational foundation upon which interactive Web-based visualizations can be built.

The UDC is designed to be a distributed system allowing many parties to publish their data. However, it is important that an initial set of example data sets is provided. Eventually an import wizard user interface may be introduced, but initially the fastest way to import data is to write small scripts to do so. Our initial targets for import via scripts include

- population data for World Countries over Years from the United Nations,
- US Census population and demographic data for US States and Counties over Years,
- the Millennium Development Goals Indicators data set from the United Nations,
- the US Bureau of Labor Statistics Employment data set, and
- the World Development Indicators from the World Bank.

These particular data sets were chosen because public visualizations of them have many applications including education, journalism, and public policy.

Since the UDC includes metadata allowing unambiguous interpretation of the data, that metadata can be utilized in visualizations. For example, the UDC metadata can be used to automatically generate axis labels for scatter plots (including units of measurement), labels for bars in bar charts, labels for regions in maps, labels for color legends, footnotes noting the source of the data (with links to the original source), and even visualization titles. This kind of labeling is essential for compelling and understandable visualizations [4], and is easy to “get wrong” for inexperienced visualization authors.

3.4 The UDC Visualization System

The fourth component, called the Universal Data Cube Visualization System (UDCViS) is an interactive visualization platform featuring multiple linked views that is built upon the first three components. The UDCViS contains standard visualization features such as color ramps, normalizations, well known visualization techniques, multiple visualizations arranged on the page (dashboards), brushed selection, subset selection, interactive OLAP operations (drill-down, roll-up, slice, dice, pivot), and brushed probing. The aim of this system is to serve as a testbed for research, experimentation and rapid prototyping, as well as a platform for developing information visualization products usable by the public. In addition, it can serve as a commons for the global information visualization research community in terms of data sets (in the UDC), visualization algorithm implementations (in CodeHub), and concrete visualization configurations and dashboards (using ASH).

4 Implementation Status

The essential features of CodeHub have been implemented using Node.js, MongoDB and Git (for storing script revision histories). The site is live at `code-hub.org`. A prototype of ASH that supports only synchronous collaboration has been implemented using Node.js and WebSockets. The UDC publication guidelines have been partially defined, and a subset of the UN Population data set has been imported into the UDC format. The dissertation is expected to be completed within the next two years.

References

[1] C. Ahlberg. Spotfire: an information exploration environment. *ACM SIGMOD Record*, 25(4):25–29, 1996.

[2] A. Baumann. *The design and implementation of Weave: A session state driven, web-based visualization framework*. PhD thesis, University of Massachusetts Lowell, 2012.

[3] A. Baumann, A. Dufilie, S. Kolman, S. Kota, G. Grinstein, and W. Mass. Exploratory to presentation visualization, and everything in-between: providing flexibility in aesthetics, interactions and visual layering. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 200–204. IEEE, 2011.

[4] J. Bertin. Semiology of graphics: diagrams, networks, maps. 1983.

[5] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1121–1128, 2009.

[6] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[7] S. Callahan, J. Freire, E. Santos, C. Scheidegger, C. Silva, and H. Vo. Vistrails: visualization meets data management. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 745–747. ACM, 2006.

[8] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[9] A. Cuzzocrea and S. Mansmann. Olap visualization: Models, issues, and techniques. *Encyclopedia of Data Warehousing and Mining*, pages 1439–1446, 2009.

[10] R. Cyganiak and D. Reynolds. Rdf data cube vocabulary specification. *W3C working draft*, 2012.

[11] A. Doan, R. Ramakrishnan, and A. Halevy. Mass collaboration systems on the world-wide web. *Communications of the ACM*, 2010.

[12] S. Eick. Visual discovery and analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):44–58, 2000.

[13] A. Gee, H. Li, M. Yu, M. Smrtic, U. Cvek, H. Goodell, V. Gupta, C. Lawrence, J. Zhou, C. Chiang, et al. Universal visualization platform. In *Proc. SPIE*, volume 5669, pages 274–283, 2005.

[14] M. Goldman, G. Little, and R. Miller. Collabode: collaborative coding in the browser. In *Proceeding of the 4th international workshop on Cooperative and human aspects of software engineering*, pages 65–68. ACM, 2011.

[15] H. Gonzalez, A. Halevy, C. Jensen, A. Langen, J. Madhavan, R. Shapley, W. Shen, and J. Goldberg-Kidon. Google fusion tables: web-centered data management and collaboration. In *Proceedings of the 2010 international conference on Management of data*, pages 1061–1066. ACM, 2010.

[16] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7(1):49–62, 2008.

[17] J. Heer, S. Card, and J. Landay. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 421–430. ACM, 2005.

[18] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1189–1196, 2008.

[19] J. Heer, F. Viégas, and M. Wattenberg. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1029–1038. ACM, 2007.

[20] O. Lassila, R. Swick, et al. Resource description framework (rdf) model and syntax specification. 1998.

[21] M. Livny, R. Ramakrishnan, K. Beyer, G. Chen, D. Donjerkovic, S. Lawande, J. Myllymaki, and K. Wenger. Devise: integrated querying and visual exploration of large datasets. In *ACM SIGMOD Record*, volume 26, pages 301–312. ACM, 1997.

[22] S. Mansmann and M. Scholl. Exploring olap aggregates with hierarchical visualization techniques. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1067–1073. ACM, 2007.

[23] S. Mansmann and M. Scholl. Visual olap: A new paradigm for exploring multidimensional aggregates. In *Proc. of IADIS Intl Conf. on Computer Graphics and Visualization (CGV)*, pages 59–66, 2008.

[24] T. Priebe and G. Pernul. Ontology-based integration of olap and information retrieval. In *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pages 610–614. IEEE, 2003.

[25] H. Rosling, A. Rosling-Ronnlund, and O. Rosling. New software brings statistics beyond the eye. In *Proceedings of OECD World Forum on Indicators: Statistic, Knowledge and Policy*, 2004.

[26] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):52–65, 2002.

[27] C. Stolte, D. Tang, and P. Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 112–122. Citeseer, 2002.

[28] C. Stolte, D. Tang, and P. Hanrahan. Multiscale visualization using data cubes. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):176–187, 2003.

[29] D. Swayne, D. Cook, and A. Buja. Xgobi: Interactive dynamic data visualization in the x window system. *Journal of Computational and Graphical Statistics*, pages 113–130, 1998.

[30] D. Swayne, D. Lang, A. Buja, and D. Cook. Ggobi: Evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43(4):423–444, 2003.

[31] F. Viegas, M. Wattenberg, F. Van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1121–1128, 2007.

[32] M. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization'94*, pages 326–333. IEEE Computer Society Press, 1994.