This article outlines the benefits of an industry-wide notation to express the rules for representing visualizations. It highlights one particular notation called VizJSON, which is based on Leland Wilkinson's *The Grammar of Graphics* approach for visualization.

Introduction

VizJSON is a markup language for describing charts to a rendering engine such that the engine can interpret the associated data and display it in a chart. It is similar in concept to HTML in that HTML is a common markup language that browsers can interpret and use to display web pages. Think of it this way: VizJSON is for visualization renderers as HTML is for browsers. This article describes the advantages of an open language (such as VizJSON) for describing visualization. This language can serve as the foundation of new open-source software that produces and consumes VizJSON as the descriptive language of visualization.

VizJSON is a language-based specification of a visualization data model based on *The Grammar of Graphics* (see Related topics) and expressed in JavaScript Object Notation (JSON). *The Grammar of Graphics* describes the rules that connect data and graphics. The ideas behind this book have influenced statistical software such as R and SPSS. (See Related topics to more about R and SPSS.)

VizJSON defines charts as a set of features that can be mixed and matched, then applied to data sets to help convey specific information about the data. For example, consider a data set on the frequency of patient visits to dentists in the United States. VizJSON could be used to highlight the fact that the frequency of visits to dentists is increasing over time on the east coast of the United States.

Using VizJSON, the features of a particular chart can be predefined to create a template that conveys a specific message for a particular type of data. The resulting template, which displays a bar chart, pie chart, word graph, or other format, can become an agreed way to present a particular type of data within an organization or across organizations. Templates, which can be reused, offer a consistent way for the data to be viewed, interpreted, and analyzed.

The VizJSON approach makes it simple to offer interactive capabilities that enable users to take existing data sets and render them in different ways to tell different stories or to take different data
sets and render them in the same way to easily analyze their differences. Because VizJSON is expressed in JSON, it can be easily integrated into HTML, presentations, or other formats. It can even be shared on the web through social media.

**Charts and visualizations described by VizJSON**

VizJSON is a language-based specification of a chart. Various charts are specified in different ways:

- **Bar chart**— Can be represented by basic 2-D coordinates with categorical \( x \) numerics displayed with intervals dropped from locations.
- **Line chart**— Can be represented by basic 2-D coordinates with any \( x \) numeric displayed with lines connecting locations.
- **Histogram**— Can be represented by basic 2-D coordinates with numeric \( x \) statistic binned counts, displayed with intervals dropped from locations.

VizJSON provides an orthogonal set of features that describes all common charts, almost all uncommon charts, and most cutting-edge research charts.

Figure 1 illustrates that, using the grammar of graphics approach, a user can easily change how data is visualized by simply changing one line in the specification. This one-line change causes the data to be displayed as a specialized pie chart instead of a bar chart. With this capability, tools providers now have a very easy way to write tools to change the specification to enable users to easily view their data in many formats that can help the reader understand what information the data is trying to convey.

**Figure 1. Simple change in the specification transforms a bar chart into a pie chart**

A simple change turns a bar chart into a pie chart
Benefits and problems solved by using the VizJSON approach

One of the main problems with data and the visualization of data is that there are many ways to represent data. The variety poses problems because when data is shared among companies, or even within different divisions of the same company, the data can be rendered differently or may be unable to be rendered at all because the visualization engine is unable to understand the input.

VizJSON addresses these issues by providing a declarative way to express visualizations in much the same way that HTML provides a declarative way to express a web page. By standardizing how the visualization is represented through the VizJSON graphics grammar (serialized into a JSON notation), VizJSON offers a consistent method to represent visualizations that can be used in JSON-based web APIs, in regular HTML, and with various JSON-based software.

JSON templates and complete visualizations (complete visualizations include the template and the data) can even be stored and queried in the increasingly popular JSON databases. Many benefits arise from having a standard visualization notation. It costs less to build visualization software over time using sets of supporting libraries and popular templates, along with building blocks for more varied sets of visualizations. By having an agreed set of VizJSON graphics grammar, multiple client targets and rendering engines can be supported more easily. In addition, it's easier to use visualizations in social media because VizJSON representations can be shared on Facebook and other social media outlets.

With a standard set of VizJSON templates, it becomes easier for regulatory bodies and governments to provide guidelines for how data should be presented to people so it can be compared more easily. Moreover, with the use of VizJSON metadata in the templates, rendering engines can ensure that the visualizations are accessible by implementing the Web Accessibility Initiative Accessible Rich Internet Applications Suite (WAI-ARIA) (see Related topics).

How VizJSON compares to existing technologies

VizJSON enables users to declare how they would like data to be displayed to convey a certain story or to show the data in a way that makes the information easier to understand. VizJSON provides a common way to express a visualization, much in the same way HTML provides a common way to express a web page. Once a common approach for expressing a visualization is adopted, we can have many ways of creating or displaying the specification.
VizJSON supports several formats:

- **SVG**— A lower-level notation used for rendering output intended for a web browser.
- **Dojo**— A JavaScript library that can be used to develop JavaScript applications that work across browsers. Dojo can be used to render VizJSON on a browser by using the Dojo Widget Extensions for VizJSON. (This works the same way any other extension works.) Other alternatives to using the VizJSON DOJO extensions include using JQuery plug-ins for VizJSON or Sencha plug-ins for VizJSON.
- **canvas tag**— An HTML5 tag to identify a drawable region. Dojo widgets can use the canvas tag to display charts including VizJSON.

Rapidly Adaptive Visualization Engine (RAVE) is the underlying technology that understands how to render a VizJSON specification. The following open source products and technologies offer similar functions to using VizJSON and RAVE:
• **Tableau Desktop**— A desktop application that enables users to quickly and easily connect to a data source, analyze the data, create a visualization, and create a dashboard with the results. It is easy to use, but is limited in the number of visualizations it can create. RAVE together with VizJSON can create an unbounded number of visualizations on multiple platforms.

• **D3.js**— A JavaScript library for manipulating documents based on data. The library enables users to bind data to a Document Object Model (DOM) and apply transforms to that object model to get the desired results. D3.js uses the web standards CSS3, HTML5, and SVG to achieve flexibility and to avoid the limitations of being bound to proprietary technologies. D3.js is the closest technology to RAVE/VizJSON and provides the most flexibility and freedom to its users.

• **Google Charts**— A JavaScript library wholly owned by Google. The current release is focused solely on the web. Therefore, the charts that are created with Google Charts JavaScript APIs are only available through a browser at this point in time. The data for the charts that the user will create can be embedded in the document itself or can come from an external source, such as Google Spreadsheets and Google Fusion Tables. The user can also take advantage of the Chart Tools DataSource protocol, but this method is limited because the datasource must implement the Chart Tools DataSource protocol. The number of chart types supported by Google Charts is also currently limited to 14. These charts can be considered templates, and the user can customize the type of chart, making changes to colors and fonts, line widths, and other visuals. However, the user is still limited to the base charts provided.

• **R + ggplot2**— A data visualization package for the statistical programming language R. Ggplot2 is an implementation of the *Grammar of Graphics* model, which is a scheme that breaks up charts into their components such as axis, scales, type of chart, etc. This system allows for modularity of the graphics created in a chart that represents a particular data set. However, this approach requires extensive knowledge of the underlying packages to have the flexibility to create multiple types of charts or to enhance a specific chart.

**IBM products that include VizJSON support**

Many IBM products and technologies use VizJSON:

• **CFO Performance Insight**— Improves visibility, insight, and control over financial performance and key financial performance indicators. It enables CFOs to measure, monitor, and anticipate business performance with a pre-configured solution.

• **Next Best Action**— Helps maximize every client interaction to build long-term customer relationships.

• **SPSS Decision Management 7**— Automates and optimizes transactional decisions before deployment to consistently maximize outcomes. It combines predictive analytics, local rules, and scoring to deliver recommended actions in real time. This software can help guide employees and systems to make the right business decisions every time.

• **Cognos® Insight**— Personal analytics solution that empowers users to independently explore, analyze, visualize, and share data without relying on IT for assistance. In minutes, analysts and line-of-business managers can create and share custom applications, dashboards and visualizations to help solve individual and workgroup challenges.
• **Cognos BI 10.2**— Provides reports, analysis, dashboards and scoreboards to help support the way people think and work when they are trying to understand business performance. You can freely explore information, analyze key facts, and quickly collaborate to align decisions with key stakeholders.

• **InfoSphere® BigInsights™ with BigSheets technology**— Brings the power of Hadoop to the enterprise. Apache Hadoop is the open source software framework used to reliably managing large volumes of structured and unstructured data.

**Summary**

VizJSON provides a JSON notation to represent visualizations by defining a set of features that can be applied to a data set. Standardizing VizJSON and using the agreed-upon notation across diverse environments provides many benefits:

• **Shared services**— Common services and libraries can be built to process VizJSON.

• **Shared toolsets**— Common toolsets can be created to help people build visualization quickly and easily.

• **Reusable building blocks**— Common building blocks (VizJSON templates, for example) that can be created and used for unique and specialized visualizations.

• **Reduced cost**— Established products can take advantage of VizJSON libraries and patterns to reduce costs.

• **New clients**— Established products can take advantage of new VizJSON clients created for new environments. New clients benefit from the established portfolio of products that support VizJSON.

• **Heterogeneous solutions**— Diverse products and solutions from multiple suppliers can consume and produce visualizations that can be shared.

• **Visualization templates**— A rich sets of VizJSON visualization templates can be created for reuse, which can be augmented by third parties easily.

• **Web APIs**— VizJSON representations can be used in conjunction with increasingly popular JSON-based web APIs.

**Acknowledgement**

The authors would like to thank Sukumar Beri for helping with this article.
Related topics

- Learn more about VizJSON in this *WorkingVis blog entry* on *The Grammar of Graphics*, including VizJSON and Vega examples.
- Check out *Visualizing Time: Designing Graphical Representations for Statistical Data (Statistics and Computing)*.
- Find out more about ggplot in "An introduction to ggplot: An implementation of the grammar of graphics in R," by Hadley Wickham.
- Get some *SPSS GPL examples*.
- Learn more about the *WAI-ARIA Accessible Rich Internet Applications Suite*.
- Find resources to help you *get started with InfoSphere BigInsights*, IBM's Hadoop-based offering that extends the value of open source Hadoop with features like Big SQL, text analytics, and BigSheets.
- **Download InfoSphere BigInsights Quick Start Edition**, available as a native software installation or as a VMware image.
- Find resources to help you *get started with InfoSphere Streams*, IBM's high-performance computing platform that enables user-developed applications to rapidly ingest, analyze, and correlate information as it arrives from thousands of real-time sources.
- **Download InfoSphere Streams**, available as a native software installation or as a VMware image.

© Copyright IBM Corporation 2013

**Trademarks**
(www.ibm.com/developerworks/ibm/trademarks/)

VizJSON: The grammar of graphics in JSON