JavaScript (HTML5, CSS3) Toolkits for InfoVis (Graphics)

Group 2

Amir Kanuric, Raoul Rubien, Jörg Schlager

706.057 Information Visualisation SS 2012
Graz University of Technology

2 May 2012

Abstract

Graphical data visualization is used to display data in a comprehensive way in form of various types of graphs. There are a number of technologies used for graph generation. In this survey reader can get an overview over some most popular JavaScript information visualization libraries that are available today. Such libraries simplify the development process of custom visualization tools or provide ready to use functionalities which can be used for rendering of basic types of graphs like: Pie charts, Line charts, Area charts or to render special graphs like: Sunburst (with Highcharts), Gauges (with RGraph), Stepped plots (with Flot) or BoxPlot (with Sparklines). The libraries listed below use JavaScript, HTML5, CSS, CSS3, canvas, SVG, and VML as technique for rendering.

The purpose of information visualization is to amplify cognitive performance, not just to create interesting pictures. Information visualizations should do for the mind what automobiles do for the feet.

Contents

Chapter 1

Motivation

1.1 Motivation

People today are using huge amount of digital data both such data stored on the local user computer and such data that user consumes from the internet. Data in its raw form is hard to browse and user has to spend a lot of time in order to understand or find the information that he wants to know. One solution of this problem is a comprehensive graphical visualization of data. The advantage of this art of data visualization is that user can have fast overview over the values, to notice some extreme values or visually compare different data that is visualized. Visualization software can be used for generation of different types of graphs. There are software solutions that user can install on local computer and use it with local data as input, but we are consuming increasingly data from the internet and such data should also be visualized. Basically there are two approaches for graphics generation. First approach uses server for graphics generation, the generated graphic is transferred to the client and the client just shows it. Second approach is cleverer and uses client for graphics rendering and retrieves only the input data that should be visualized from the server. In this manner developer can build interactive applications that are running in a browser. Different technologies can be used on the client side for rendering and displaying of such graphs. The most popular technologies used nowadays are: Adobe Flash, Silverlight, JavaApplet and JavaScript. To make the development of JavaScript visualization easier, developer can use various commercial or OpenSource JavaScript visualization libraries. In this survey the reader can get an overview over some most popular JavaScript InfoVis libraries available today.

Chapter 2

Low Level Graphics Toolkits

2.1 mxGraph

2.1.1 What is mxGraph?

mxGraph is JavaScript version of more familiar market leading Java diagramming library JGraph. JGraph is an open source Java Swing graph drawing library. First version of JGraph library was released in year 2001 under BSD license and last release is 6, called JGraphX. The JavaScript version (mxGraph) was released in year 2005 under Commercial license. This library is under active development and new features, examples and layouts are regularly provided. It is used to build interactive rich internet diagramming applications that are running in most modern browsers using pure JavaScript and without installation of any plugins. [JGraph Ltd., 2012]

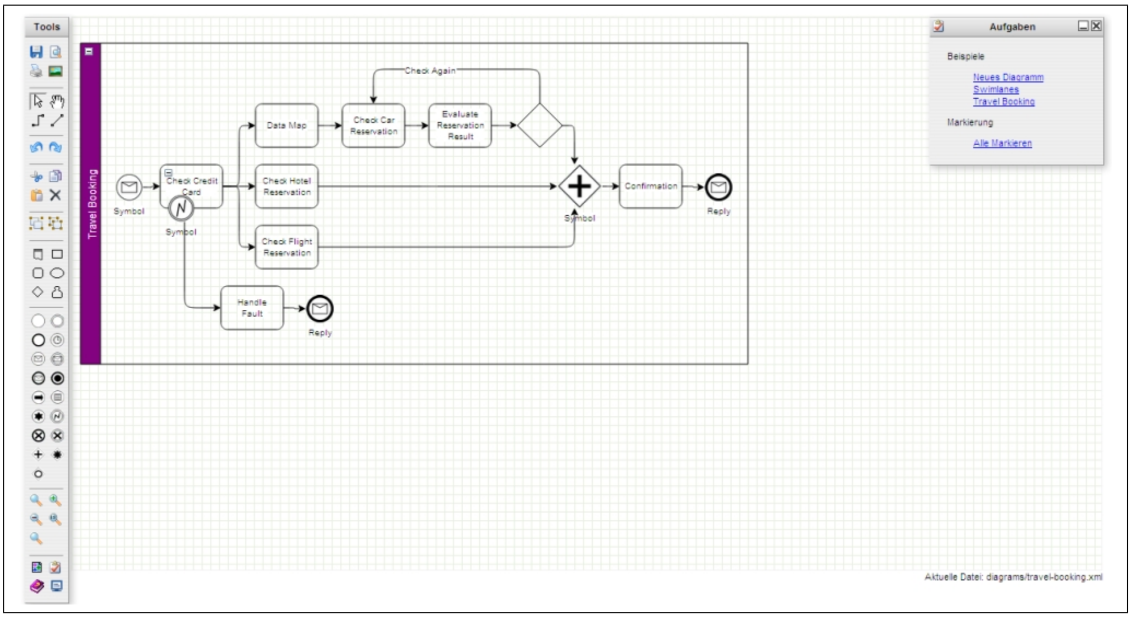


Figure 2.1: mxGraph example application: Workflow editor

Figure 2.1 shows an example application from the homepage. This interactive graphical workflow editor can be used to create or modify different workflows by arranging and connecting particular workflow elements from the toolbox on the left side. One disadvantage is that there is no possibility to export created diagram as image file.

2.1.2 How to begin using mxGraph: Hello World example

1. Create a div element in the HTML page that will be used as container for the graph

2. Use onLoad event in body tag of document to invoke the JavaScript function that creates the graph with created container as a parameter

3. When the JavaScript function is invoked, it should create graph inside the given container, add two cells with labels to the model and connect created cells using built in library functions.

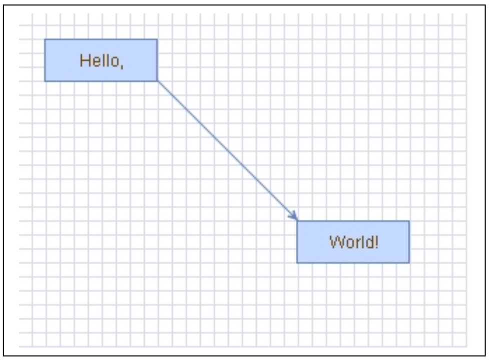


Figure 2.2: Hello world example

2.1.3 Technology

Depending on the client's browser mxGraph uses different vector graphics language for rendering and displaying graphics. VML is used for older versions of Internet Explorer which do not natively support SVG. For most other modern browsers SVG is used for rendering. There is also support for rendering using canvas, the new feature of HTML5. Which technology will be used, depends on client's browser and the mxGraph library uses transparently technology that best fits to the browser that is used, so that developer do not have to care about browser incompatibility. The advantage of using JavaScript and no other technologies like Flash, Silverlight or JavaApplet is that JavaScript requires no plugins and uses DOM elements like canvas for the rendering of graphics. The usage of mxGraph is not limited to use just some static data as input. mxGraph can use XML for communication with a server and dynamically load the data.

2.1.4 Documentation and examples

There is a user manual provided on the project's homepage, but no classical API documentation. This manual contains only basic information about library and its usage. Examples provided on the homepage can be seen as different layouts that user can adept to own wishes depending on the software that is to develop. The best way of using mxGraph library is to take an example provided on the homepage that best fits to the software which user wants to develop and to adapt its source code to the software requirements. On the homepage user can see this examples in action, but without corresponding source code or the ability for editing and live preview. In order to see the source code of an example user has to download the whole library package that contains also source code of all examples. JGraph community has forum where developer can ask their questions, but as usual in forums answers come from community members and those answers are not always realible. There is also Commercial-level support available for library customers with support package.

2.2 Raphaël

Raphaël [Baranovskiy, 2012] is a low-level vector graphics API for Javascript available under the MIT license. It includes built-in support for transitions and animations, but no direct support for generating graphs and charts. This means, that even the most basic charts, for example bar charts, have to be coded by hand and have to be built out of a list of available graphic primitives. As there are not enough built-in functions, the library can be very difficult to use. A lot of code is required to complete even the simplest tasks. In addition, the provided examples are of a rather poor quality and often fail to highlight the single feature the user is interested in. In conclusion, Raphaël has to be considered an abstract framework which may be used as a base to build a new, more application-oriented framework for generating custom charts.

2.2.1 Technology

The vector graphics are rendered using Scalable Vector Graphics (SVG) and the Vector Markup Language (VML) is used as a fallback for older browsers.

2.2.2 Browser Compatibility

Raphaël currently supports Firefox 3.0+, Safari 3.0+, Chrome 5.0+, Opera 9.5+ and Internet Explorer 6.0+.

2.2.3 Usage Examples

1. Path Animation

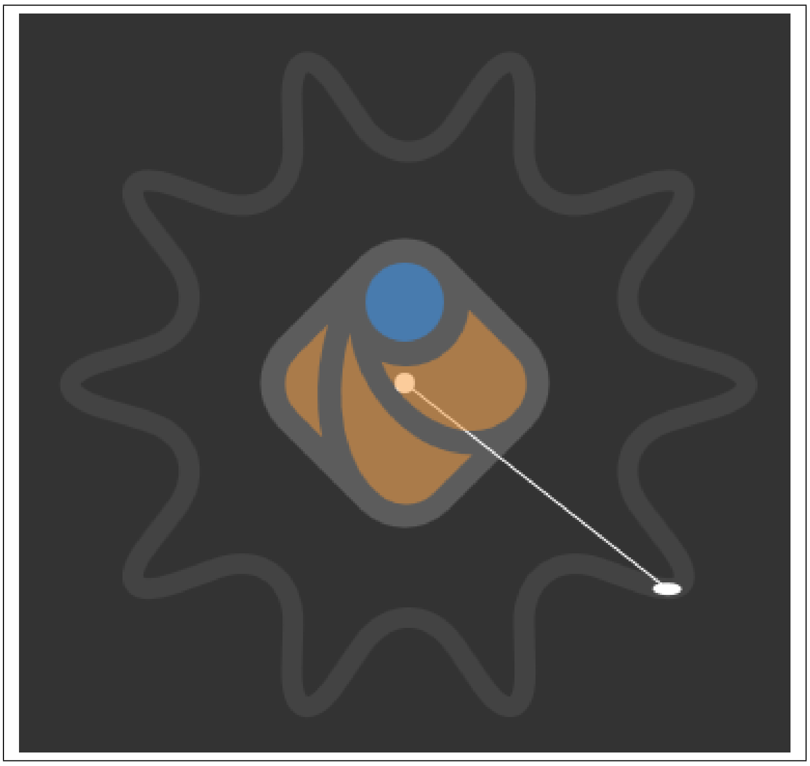


Figure 2.3: Movement of an object along a path

2. Clock

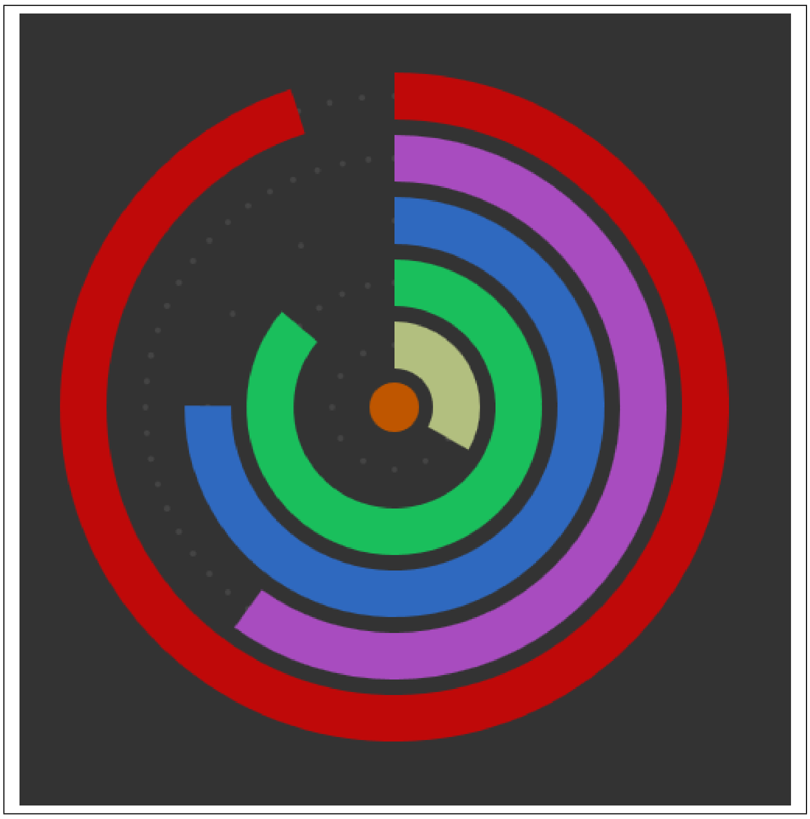


Figure 2.4: Animated clock with nice transition effects

3. Area Chart

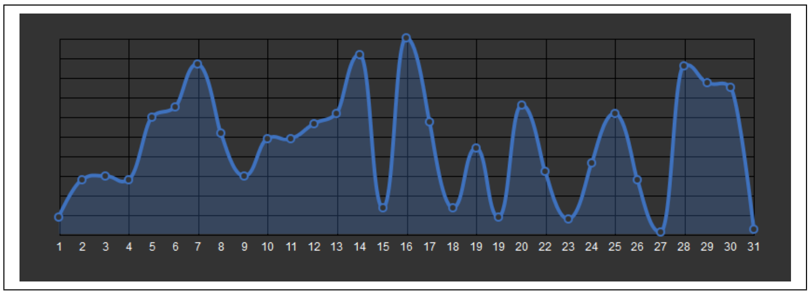


Figure 2.5: Area chart

2.3 Comparison of Low-Level Toolkits

	mxGraph	**Raphaël**
Technology	HTML5, SVG,VML	HTML5, SVG
Maturity	Stable	Stable
License	Commercial (6.000 EUR)	MIT
Documentation	**o**	**o**
Examples	**+**	**-**
Updates	30 / year	Frequently
Popularity	**o**	**o**
Browser Compat.	++	++
Interactivity	**++**	**+**
Transitions	**−**	**+**

Chapter 3

Chart Toolkits

3.1 Highcharts

Highchart is a dual licensed charting toolkit that can visualize many charts. It's goal is to offer intuitive and interactive charts for websites. Beside it supports a palette of conventional charts it comes up with mature technology and well documented online examples. The provided documentation is kept clear and the demonstration gallery provides multi-faceted examples that can be modified online in a sandbox-like tool named jsFiddle. With many releases a year, a frequently used support forum and an online feature request site it leaves a trustworthily impression. [Highsoft Solutions AS, 2012]

3.1.1 Appreciable Features

During investigation some features attracted attention: The export and print functionality is available on nearly any chart in form of two tiny buttons on the top right. Exporting to PDF, JPEG, PNG and even vector graphic SVG is possible. Zooming is done by click-and-select an area and hiding graphs by clicking on its description in the chart's caption. Furthermore on mouse over datasets labels with values are faded in. The combination of this mentioned features makes the use of Highcharts very intuitive.

3.1.2 Chart Types

Although this tool can be called highly sophisticated it supports only usual charts. On the other hand those are well advanced.

- Line and scatter charts
- Area chart
- Dynamic chart
- Column and bar charts
- Pie chart
- Combinations (i.e.: Multiple axes, scatter with regression line)

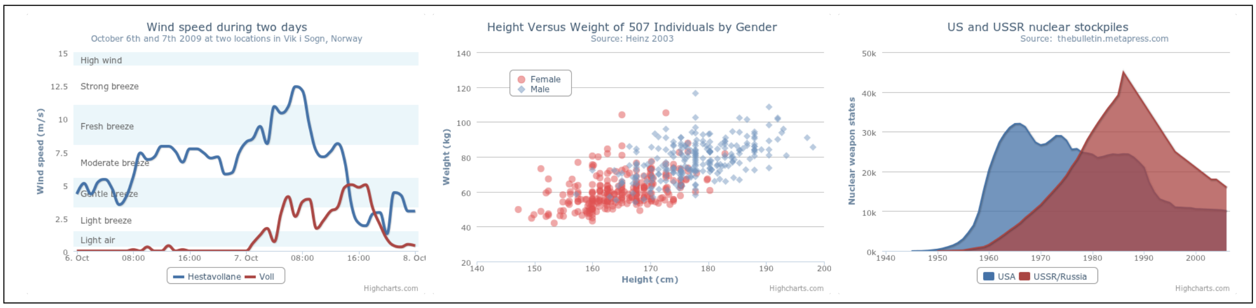


Figure 3.1: Line chart, scatter chart, area chart

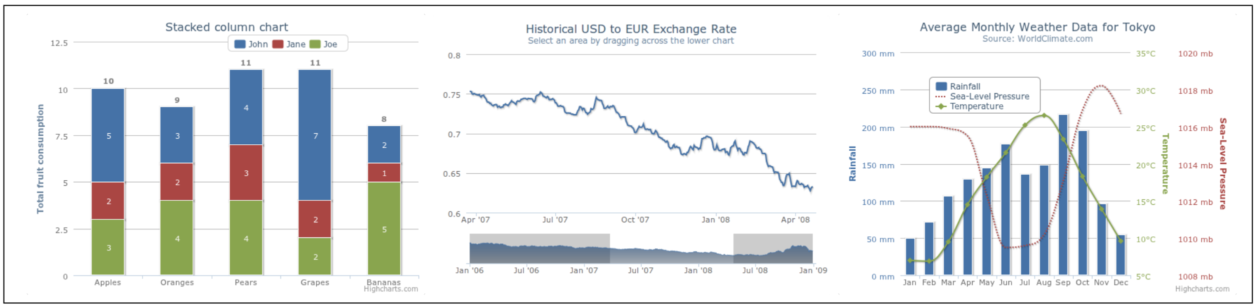


Figure 3.2: Stacked column chart, dynamic chart, multiple axes chart

3.1.3 Browser Compatibility

In comparison with other charting toolkits the browser compatibility is very good. One positive aspect is the dynamically chosen technology to get best performance and compatibility out of the currently used browser. The underlying compatibility list is remarkable.

Brand	Versions supported
Internet Explorer	6.0 +
Firefox	2.0 +
Chrome	1.0 +
Safari	4.0 +
Opera	9.0 +
iOS (Safari)	3.0 +
Android Browser	2.0 + (with some limitations)

Table 3.1: Browser compatibility list

3.1.4 Technology

The toolkit makes use of external JavaScript (jQuery, Mootools, Prototype) frameworks to do DOM manipulations in the browser. For the development this brings the advantage to test latest frameworks with less delay. The drawing technique depends on

3.2 RGraph

RGraph [Heyes, 2012] is made available under a dual license, where the free version may only be used for personal and educational projects. The package includes many exclusive, business-oriented charts, for example GANTT, waterfall and funnel diagrams. In addition, radar rose and bipolar charts are available. Many of

Browser version	Rendering technology	Rendering performance
Internet Explorer 9	SVG	Excellent
Internet Explorer 8	VML	Ok
Internet Explorer 7	VML	Slow
Internet Explorer 6	VML	Slow
Firefox	SVG	Excellent
Chrome	SVG	Excellent
Safari	SVG	Excellent
Opera	SVG	Excellent
iOS Safari	SVG	Ok
Android 3+	SVG	Ok
Android 2.x	Canvas	Slow

Table 3.2: Rendering technology and performance

different chart types feature nice transition effects. All charts can be saved as bitmaps with a transparent background using the context menu and the ”Save Image As” option. RGraph accepts data in various formats, is very easy to use and has an excellent documentation and many useful examples. Unfortunately, there is no support forum and no complete contact information, which makes it rather difficult to recommend investing in the commercial (120 EUR) version.

3.2.1 Technology

The charts are rendered using the HTML5 canvas element. For compatibility with older versions of the Microsoft Internet Explorer, it is recommended to use the ExplorerCanvas [Google Inc., 2012a] extension.

3.2.2 Browser Compatibility

Brand	Versions supported
Internet Explorer	7.0, 8.0 (ExplorerCanvas)
Internet Explorer	9.0
Firefox	3.0+
Google Chrome	1.0+
Apple Safari	3.0+
Opera	9.5+
iPhone	4.0+
iPad	4.2+

Table 3.3: Browser compatibility list

3.2.3 Chart Examples

1. GANTT Chart

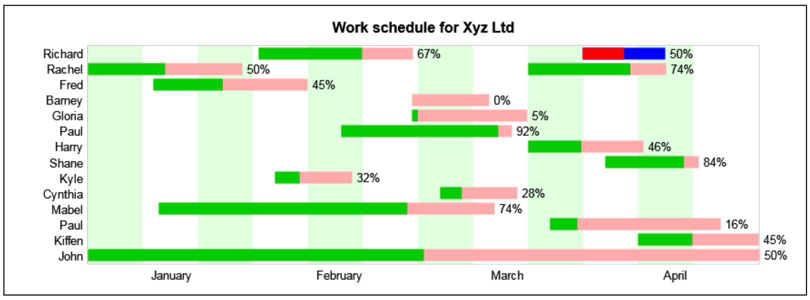


Figure 3.3

2. Rose Chart

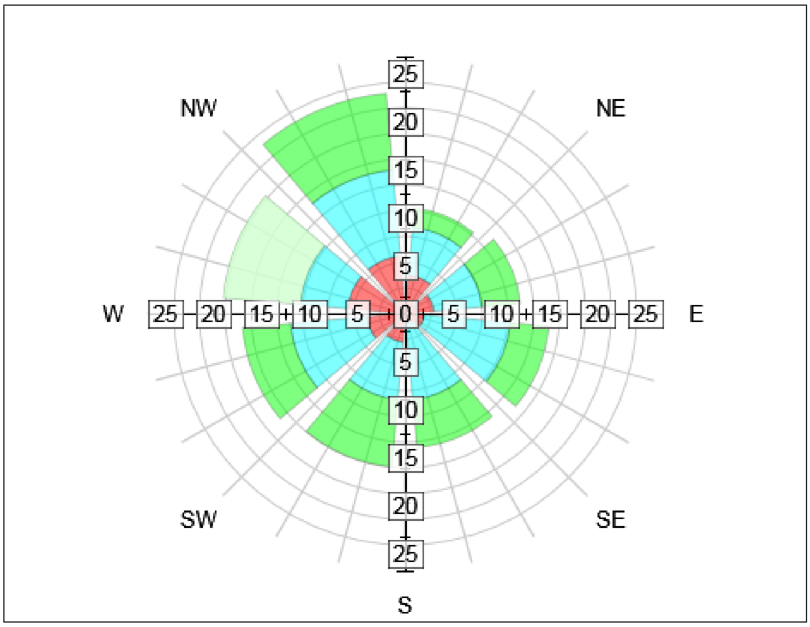


Figure 3.4

3. Bipolar Chart

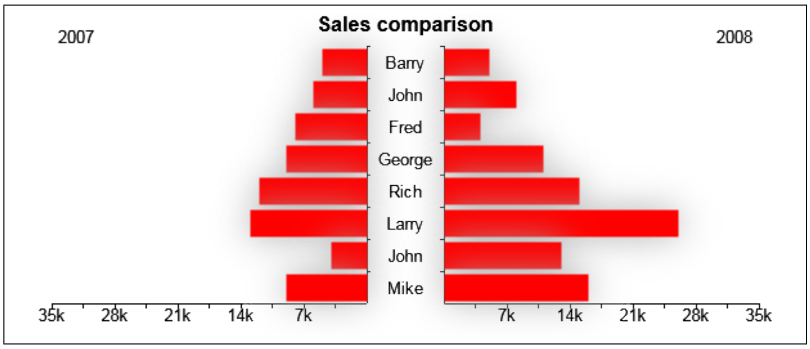


Figure 3.5

3.3 Flot

Flot [Flot, 2012] is a project which aims to provide simple, attractive charts for jQuery. Javascript plugins are supported for enhancing the built-in functionality. Using such plugins, several new chart types - which are not part of the base product - become available. Flot is a very mature project, which unfortunately appears to be no longer actively developed. While the examples are simple and comprehensible, the documentation is hardly useful as there is no index and the manual is just one huge text file.

3.3.1 Technology and Browser Compatibility

The charts are rendered using the HTML5 canvas element. For compatibility with Microsoft Internet Explorer versions 7.0 and 8.0, it is recommended to use the ExplorerCanvas [Google Inc., 2012a] extension. Internet Explorer 6.0 is supported via Vector Markup Language (VML) output.

3.3.2 Chart Examples

1. Combined Chart

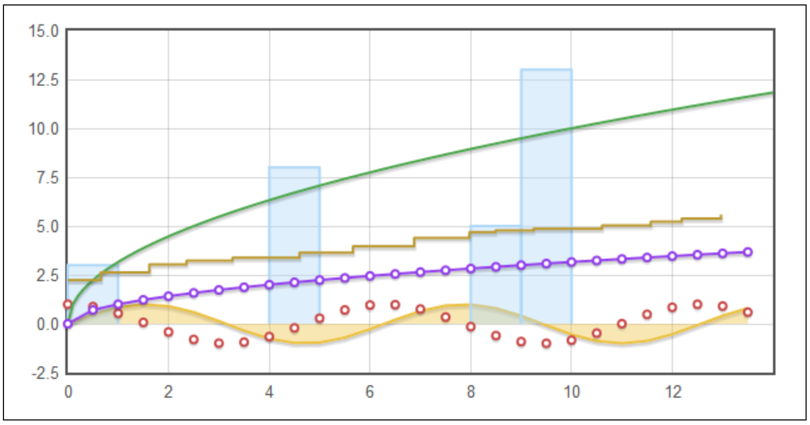


Figure 3.6

2. Filled Area Chart

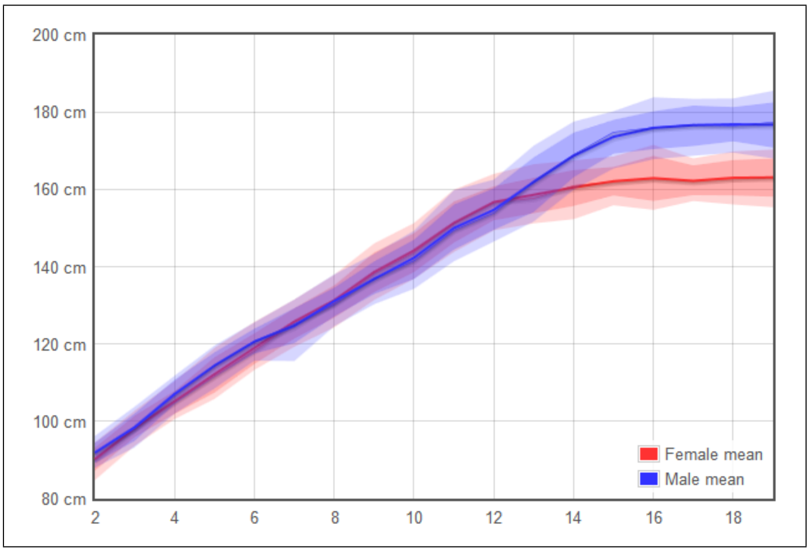


Figure 3.7

3.4 Emprise

The Emprise Javascript Charts [Emprise Corporation, 2012] Toolkit is a commercial product for creating Javascript Charts, which retails for 76 EUR. It supports interactive selection of data series and zooming and panning of charts. There are however only the most common chart types included, and the product is no longer actively developed. This is also reflected in the quality of the documentation and examples, which is only average. Nonetheless, the wide range of compatible browsers and different data sources - with built-in support for XML, CSV and JSON - is impressive and makes Emprise suitable for many different applications.

3.4.1 Technology and Browser Compatibility

The actual chart is rendered using a canvas element; the axes labels and ticks, various captions and other properties are however individual DOM objects, which may be directly modified by Javascript.

Operating System	Brand	Versions supported
Windows	Internet Explorer	6.0+
	Safari	3.1+
	Firefox	1.5+
	Mozilla	5.0+
	Netscape	9.0+
	Opera	9.0+
Macintosh	Safari	2.0+
	Camino	1.0.4
	Firefox	1.5+
iOS	iPhone Browser	1.0+

Table 3.4: Browser compatibility list

3.4.2 Chart Examples

1. Pie Chart

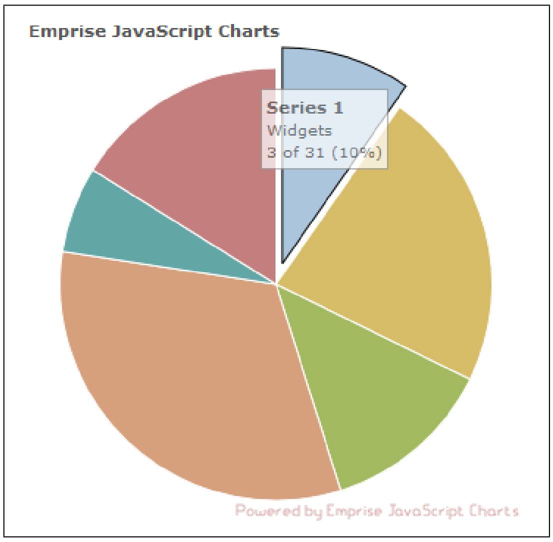


Figure 3.8

2. Candlestick Chart

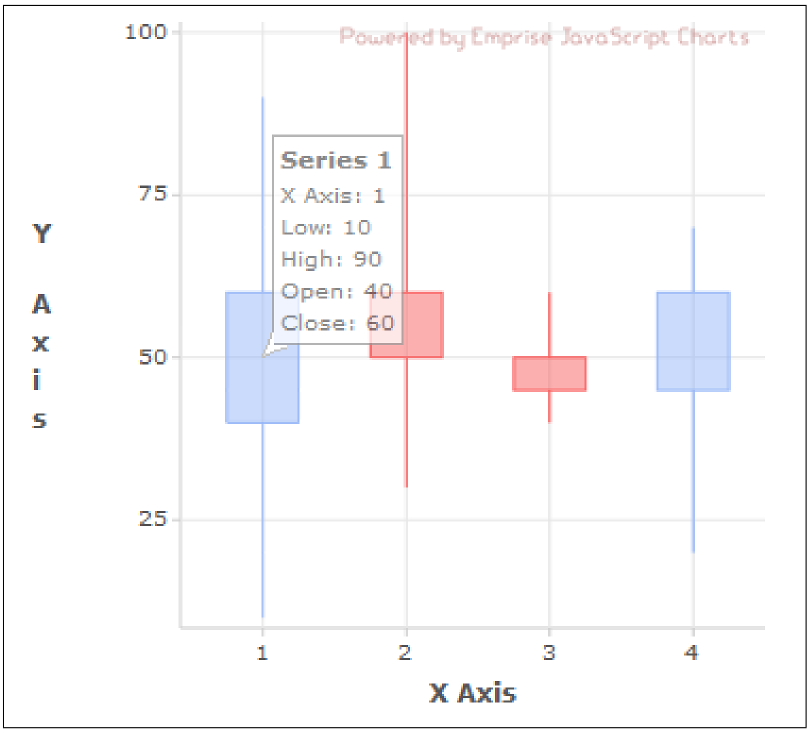


Figure 3.9

3. Combined Chart

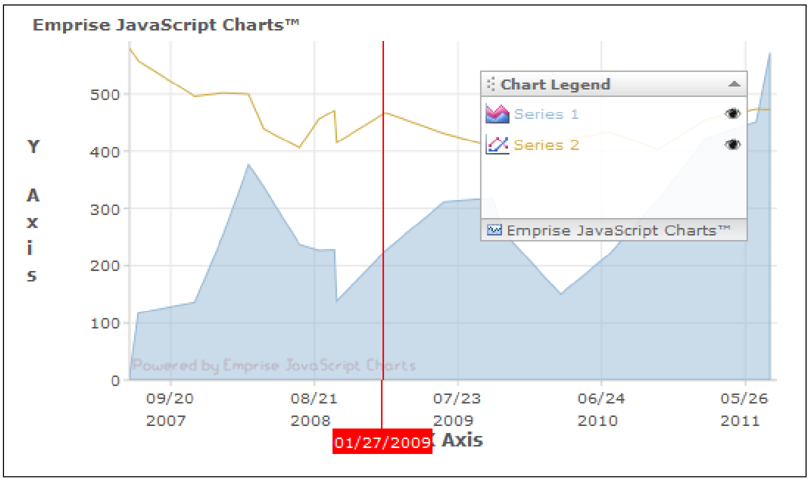


Figure 3.10: The legend text box can be moved by the user; interactive selection of data series

3.5 Sparklines

3.5.1 What are the sparklines?

Sparklines are small graphics typically used to show the shape of some numerical variations of financial or economic data. The idea is to use graphics without coordinates and to use it inline in text just for quick impression of value variations over time. Basically every type of charts can be used as sparkline if it is enough visible in such miniature form. Typical sparkline charts are bar, area and line charts. The concept and the name ”Sparkline” was introduced by Edward Tufte, university professor at University of Yale. [Splunk Inc., 2012]

Tufte [2006] Tufte [2012]

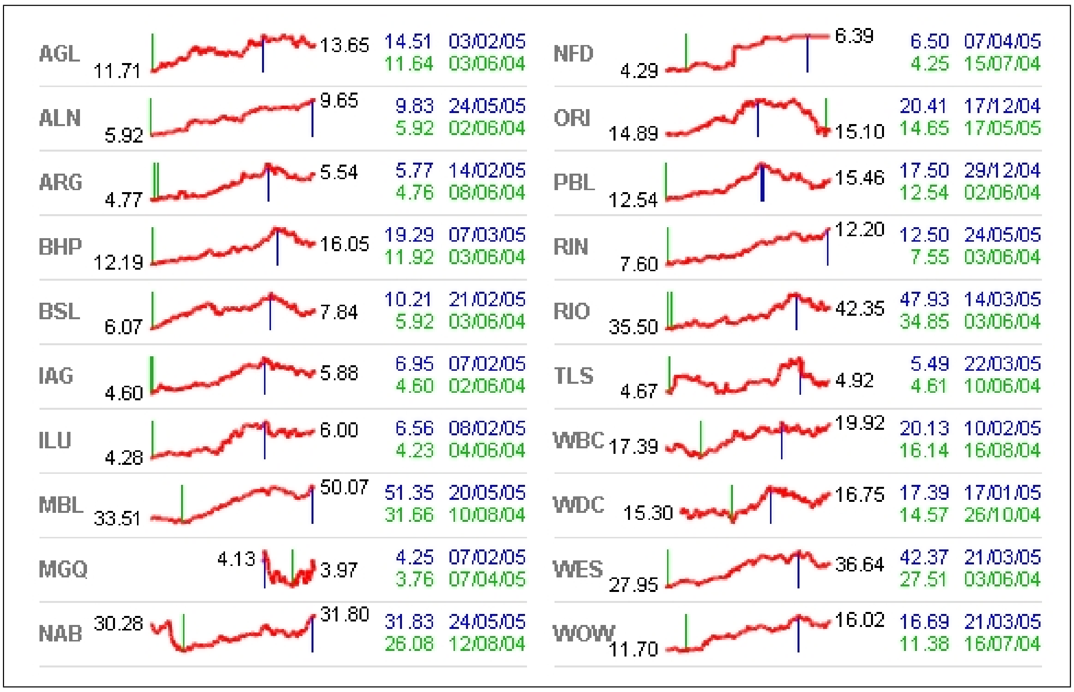


Figure 3.11: Sparklines for financial data

3.5.2 JavaScript library Sparklines

The jQuery plugin with the same name Sparklines is an OpenSource library distributed under NEW BSD license. The library is maintained by Splunk Inc and is not really up to date; the last update was on November 2010. This library provides some extra features for sparkline graphics. The support for mouse over to see underlying date of a sparkline chart is a feature that allows the user to see corresponding values on the mouse position over the graph. Using with and height parameters it is possible to create graphs in full size. The plugin is compatible with most modern browsers and has been tested with Firefox 2+, Safari 3+, Opera 9, Google Chrome and Internet Explorer 6, 7 and 8. The technology used for rendering and displaying graphics is JavaScript in combination with HTML5 element canvas. The library has no classical API-documentation, but it provides an example based documentation. For each example there is a live preview, so that user can see it in action. Additionally there is an interface on the homepage which can be used for custom configuration and preview of all sparklines that this library supports.

3.5.3 Supported charts

The library supports following types of charts:

- Line charts
- Bar charts
- Tristate charts
- Descrete charts
- Bullet charts
- Pie chart
- BoxPlot

The composition of two or more chart types is also possible. Very popular composition is a combination of line and bar charts.

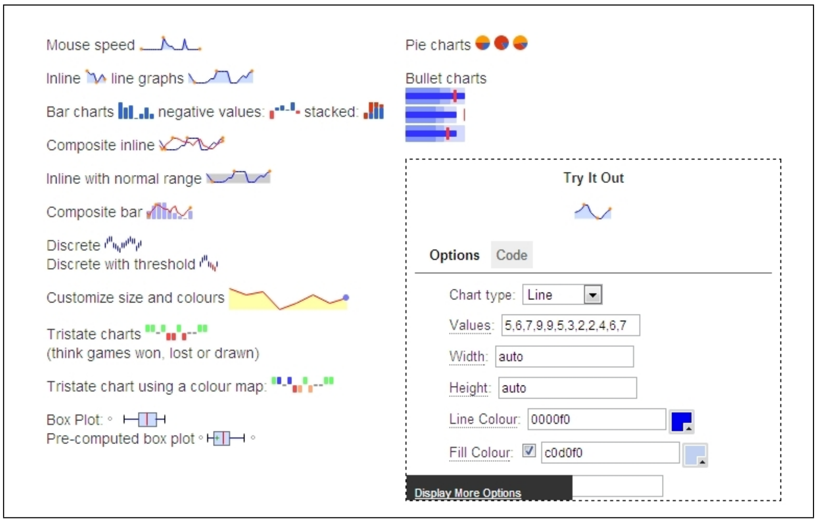


Chart type: Line

Values: 5,6,7,9,9,5,3,2,2,4,6,7

Width: auto

Height: auto

Line Colour: 0000f0

Fill Colour: ☑ c0d0f0

Figure 3.12: The example configurator with preview and some examples of charts

3.5.4 How to use the sparklines library

In order to use sparkline developer has to do following four steps:

1. Use at least version 1.4.3 of jQuery library
2. Load the plugin jquery.sparkline.js into the page
3. Inline tag which will be used to display the sparkline
4. Invoke proper sparkline function to display a graph (sparkline)

If for instance a developer wants to create a sparkline for following sequence of numbers used as input: 2,2,5,6,7,4,3,10,8,9,12, he needs only one inline tag that invokes a function from the sparkline library. It can be as simple as this:
<span class='inlinesparkline' >2,2,5,6,7,4,3,10,8,9,12 </span >
and the figure 3.2 shows generated sparkline.

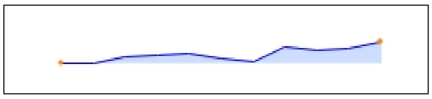


Figure 3.13: Generated Line sparkline

3.6 Comparison of Chart Toolkits

	Highcharts	**RGraph**	**Flot**	**Emprise**	**Sparklines**
Technology	SVG, VML, Canvas	Canvas	Canvas, VML	DOM + Canvas	Canvas
Maturity	Stable	Stable	Stable	Stable	Stable
License	Dual (60 EUR+)	Dual (120 EUR)	MIT	Comm. (76 EUR+)	New BSD
Documentation	++	+	-	o	+
Examples	++	+	+	o	++
Updates	Freq.	Freq.	Aband.	Inaktive	Aband.
Popularity	o	o	+	-	+
Browser Compat.	++	+ (no IE6)	++	++	++ (IE6)
Interactivity	++	o	+ (Sel.)	++	–
Transitions	++	+	- (Plugin)	–	–
Extensions	–	–	++	+	–
Data Sources JSON + ?	XML, jQuery	XML	jQuery	XML	HTML
Configurability	+	+	+	++	o
Visual Appeal	+	o	o	o	-
Graphic Export	++	+	-	-	-
Bar, Area	+	+	+	+	-
Pie, Donut	+	+	o	o	o
Special Plots	Sunburst	Gantt, Radar, Gauges	Stepped Plots	-	Box Plot

Chapter 4

Advanced Chart Toolkits

4.1 JIT - JavaScript InfoVis Toolkit

4.1.1 What is JIT?

JIT is an OpenSource JavaScript library used for creation of some special kinds of visualisation graphics. All examples of graphs rendered with JIT look very imressive and powerful, but if somebody wants to customize some of this examples and use it for own projects, in shortest thime he will notice that the customization and changes on the examples are not easy to do. The reason for this is that the JIT uses very complex data set and local settings coded in JavaScript. [Sanchalabs, 2012]

4.1.2 Layouts

JIT comes with a collection of layouts that can be used as basis for development with JIT. The library with layouts is still growing. Here are some Layouts listed that developer can use for their projects:

- Layout with basic charts: Area, Bar and Pie charts
- Sunburst
- Icicle
- ForceDirected
- TreeMap
- RGraph

Area, Bar and Pie charts

With JIT user can create some basic types of charts like area, bar or pie chart with some extra features. Figure 4.1 shows a static pie chart example with gradients that display toolkits when hoovering the stacks. On this way pie charts created with JIT can give the user some additional information about particular part of chart.

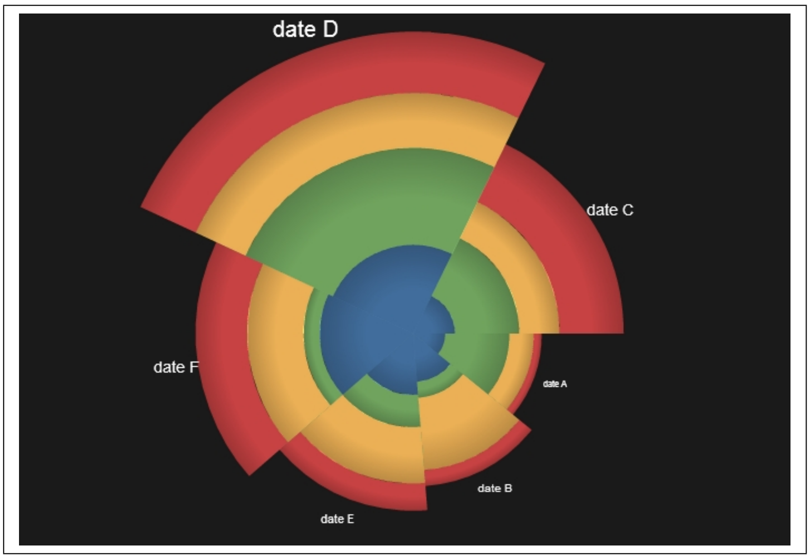
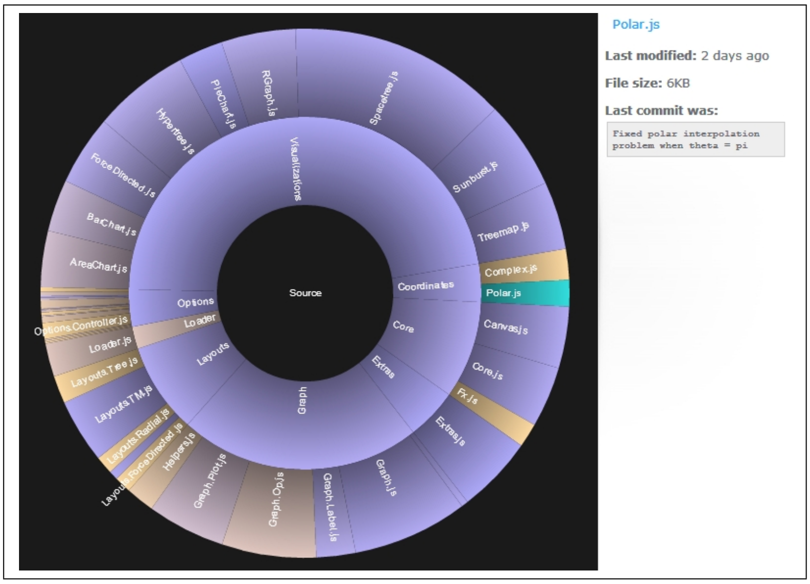


Figure 4.1: A static Pie Chart example with gradients that displays tooltips when hovering the stacks

Sunburst

The sunburst can be used, as in example shown in Figure 4.2, for the circular visualisation of a directory tree. User can navigate through the folders by clicking on one of them. The sunburst highlights selected folder and rotates so that selected folder and its files are on the right side. A click on a file causes smooth rotation of sunburst that moves selected file to the right side and displays some details about the file that has been selected.



Polar.js

Last modified: 2 days ago

File size: 6KB

Last commit was:

```
Fixed polar interpolation
problem when theta = pi
```

Figure 4.2: Sunburst of a Directory Tree

Icicle

One possible usage of Icicle can be the file system visualization (Figure 4.3). The root folder is always located on the left side and the subfolders are on the right side of corresponding parent folder. The user interaction is possible using the left and right mouse click. With the left click user can set a folder as root and it shifts selected folder to the left side. Right click sets the parent folder as root and updates the visualization.

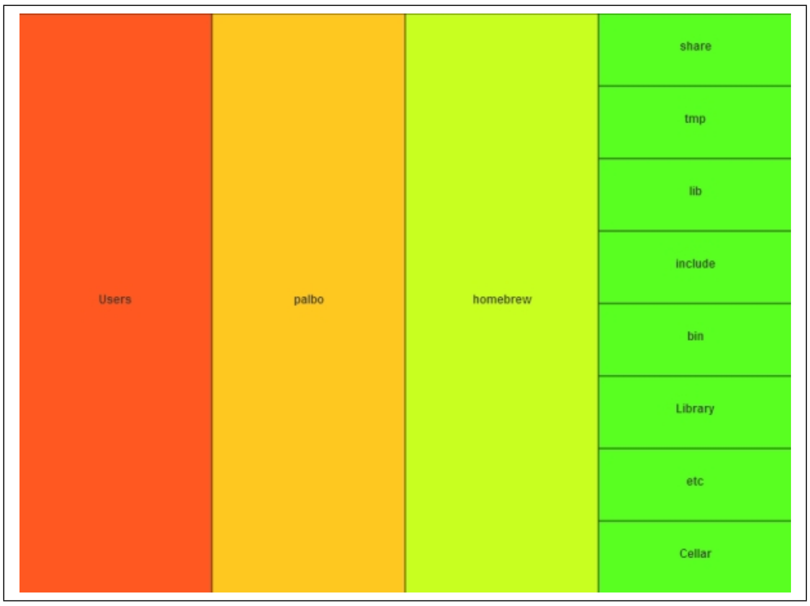


Figure 4.3: Icicle tree with limited levels

4.2 Protovis

Protovis is not a simple charting tool but rather a graphical approach to visualization. It allows to compose custom views from simple mark types (i.e. dots, bars). The project's target is to be highly customizable and allow the user to build his own visualization by using this library. For beginners the documentation is cleanly, structured and the examples are quite impressive. Never the less the provided examples have not as much interactivity or animations that other toolkits. Protovis was first released in April, 2009 and maintained till June, 2011 when the final version 3.3.1 was released. The development begun working on a new visualization library named d3.js. Switching to the new library is recommended.

[Bostock and Heer, 2012]

4.2.1 Chart Types

An great advantage of Protovis is the high number of examples that cover many unconventional chart types. To name but a few appreciable charts:

- Conventional

  Donut charts

- Hierarchies

  Circle packing

  Sunbursts

  Icicles

  Dendrograms

  Node-Link trees

- Networks

  Arc diagram

  Force-Directed layouts

  Matrix Diagrams

- Maps

  Map projections

  Dymaxion maps

  Dorling cartograms

  Symbol maps

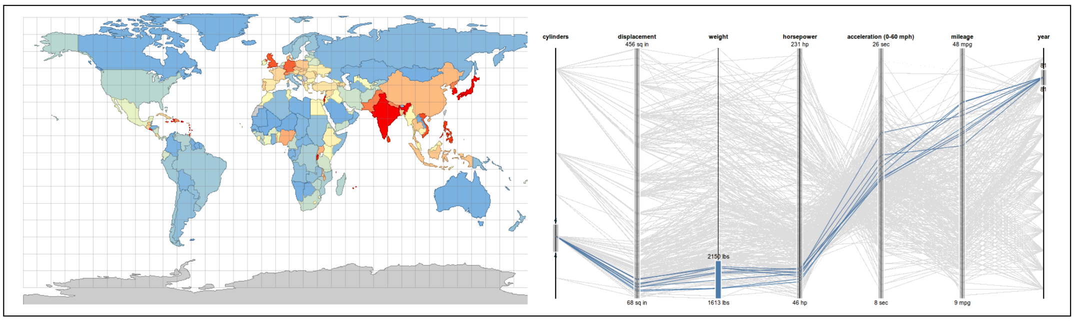


Figure 4.4: Map projection, parallel coordinates

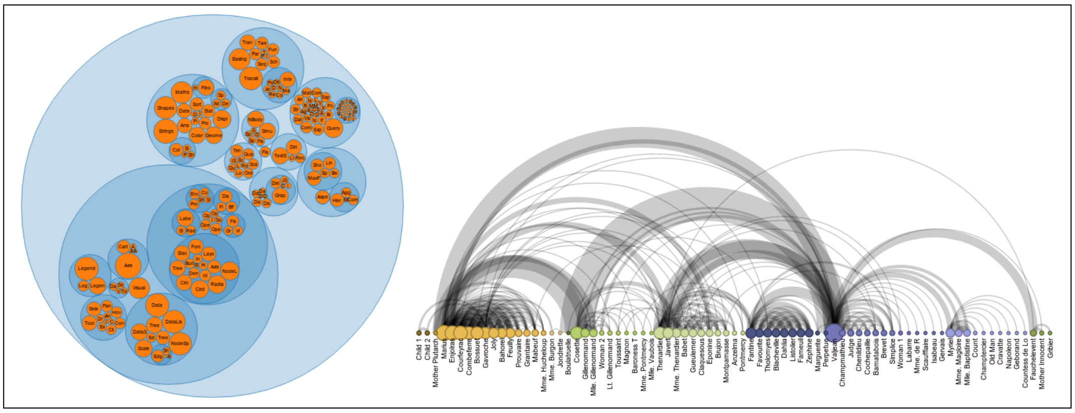


Figure 4.5: Circle packing, arc diagram

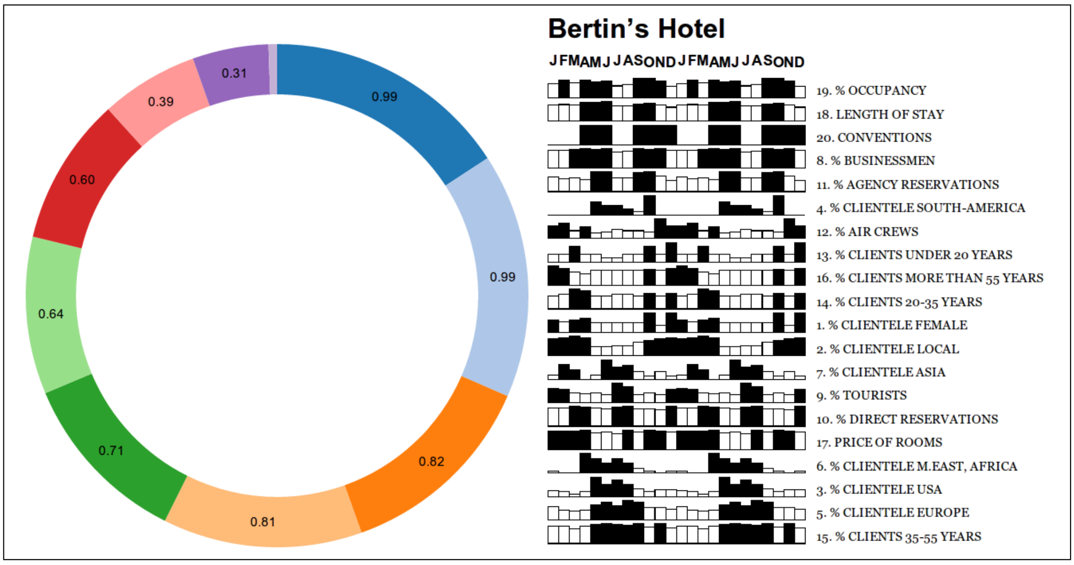


Figure 4.6: Donut chart, Bertin's hotel

4.2.2 Browser Compatibility

In comparison to Highcharts, whose compatibility is well documented, an overview is missing. The browser compatibility is not well documented though you will need a modern web browser. Problems will be encountered when using Internet Explorer 8 where the library definitely does not visualize anything except of causing an error message to pop up.

4.2.3 Technology

For native visualization JavaScript and SVG is used.

4.3 Data Driven Documents - D3

D3 is the successor of Protovis. It is closely related and shares the same target, being flexible. The library achieves more chart interactivity, animations and transitions as its ancestor. With this features the online examples are visually visually very attractive. Comparing to other toolkits, the projects web site provides the most impressive examples.
D3 helps you bring data to life using HTML, SVG and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation. [D3, 2012]

4.3.1 Appreciable Features

D3 comes with good documentation, great presentation of visualizations on the projects web site and eye-catching examples. It is interactive, animated and provides transitions for changed datasets.

4.3.2 Chart Types

The same advantage of diversity of Protovis applies to D3 also. To name just a few sensational charts is really hard due to a ton of unique visualizations:

1. Hierarchical edge bundling

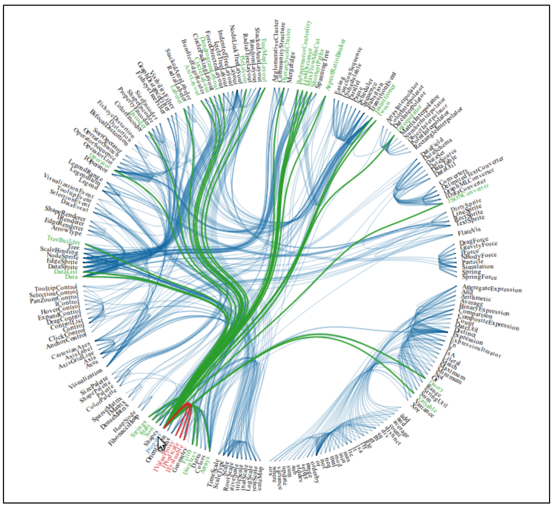


Figure 4.7: Animation: Highlighting paths on mouse over nodes

2. Layout cluster

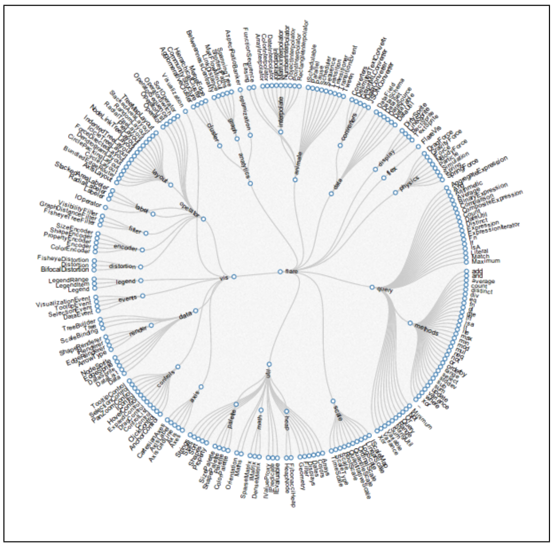


Figure 4.8: Interaction: On drag the disk can be turned

3. Cubic Hamiltonian graph

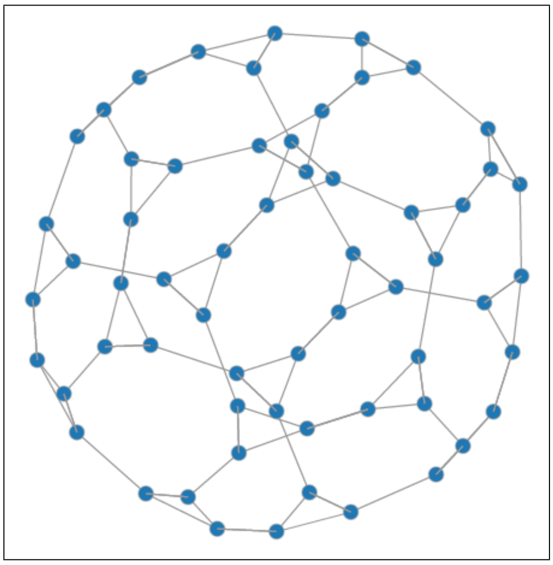


Figure 4.9: Interaction: On drag nodes react like force directed

4. Hive plots

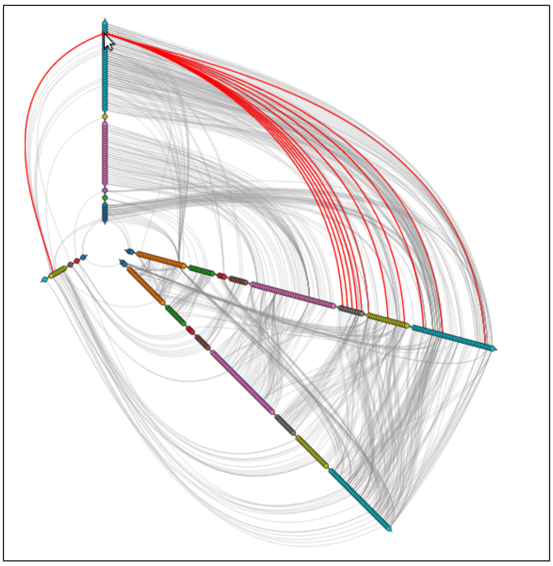


Figure 4.10: Animation: Highlighting paths on mouse over nodes

5. Voronoi diagram with force directed nodes and Delaunay links

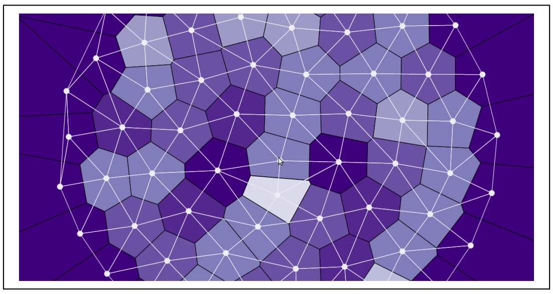


Figure 4.11: Interaction: Composition of force directed nodes and Voronoi cells (cells interact with nodes)

4.3.3 Browser Compatibility

It supports modern browsers, that means every browser above Internet Explorer 8. It is tested against most important browsers but it is no compatibility layer. If a browser does not support the standards W3C DOM, SVG and CSS (or CSS3 Transitions) D3 won't compensate the missing support. D3 uses W3C Selectors API Level 1, which are widely used in CSS, but it can preload Sizzle [2012] for compatibility.
Due to lack of minimum browser versions D3 requires the following compatibility table shows the actual versions of May, 2012.

Brand	Versions supported
Internet Explorer	9 +
Firefox	12
Chrome	19
Safari	5.1.5
Opera (for PC)	11.62

Table 4.1: Browser compatibility list

4.3.4 Technology

D3 binds data to DOM using JavaSctript, CSS3, HTML5 and SVG. Operations are done onto DOM attributes which brings positive aspects. User do not have to learn a completely new vocabulary and while learning D3 any user can take gained knowledge about DOM along. D3 works on native objects, is fast and extensible with optional modules.

4.4 Google Chart Tools

The Google Chart Tools [Google Inc., 2012c] are a Javascript library developed by Google and made available to the general public via the Google Visualization API and under the terms of the Google API Terms of Service (ToS) [Google Inc., 2012b]. Unfortunately, these terms explicitly deny the right to download or mirror the library code, which means the the end user is required to have an active internet connection when viewing a chart create by the Google Chart Tools. Furthermore, Google reserves the right to modify the terms of use at any given time, so it is possible that this service will eventually be no longer available for free.

While the Google Chart Tools do not contain as much eye candy as other applications and no fancy transition effects and all, the basic chart functionality is implemented very well. There is, for example, a dashboard system for combining individual charts to full featured dashboards enabling the end user to choose the displayed data series with custom controls such as combo boxes and buttons.

The best-in-class documentation deserves special praise. All the examples may be changed on a live demo site, the Google Playground. Changes to the sources Javascript code are immediately visible on the same page, and the examples themselves appear to be very complete and easily comprehensible.

4.4.1 Data Sources

As a special feature, the Google Chart Tools contain not only libraries for displaying charts, but also a custom data table and data source implementation. This also includes a custom SQL-like query language for data sources. A developer may choose to provide the data in a simple JSON format or to implement the full data source API, in which case Google provides advanced code for retrieving and filtering it.

4.4.2 Technology

Charts are rendered using a combination of HTML5 and SVG, if this is supported by the browser. For backwards compatibility, the Vector Markup Language (VML) is used as a fallback for older browsers.

4.4.3 Browser Compatibility

Due to the technology mix used, browser compatibility is excellent. Google states that the charts are available not only on all major browsers starting from Internet Explorer 6, but also on recent hanhdheld devices running Android or iOS.

4.4.4 Chart Types

The Google Chart Tools contain an implementation of the Motion Chart and - as you would expect from Google - excellent implementations of geographic visualizations. The almost total absence of tree is however an obvious disadvantage when comparing this product to its competitors. The org chart ist the only from of a tree visualization in this product, but it is designed specifically for one problem and can not be used for generic visualization tasks.

1. Intensity Chart

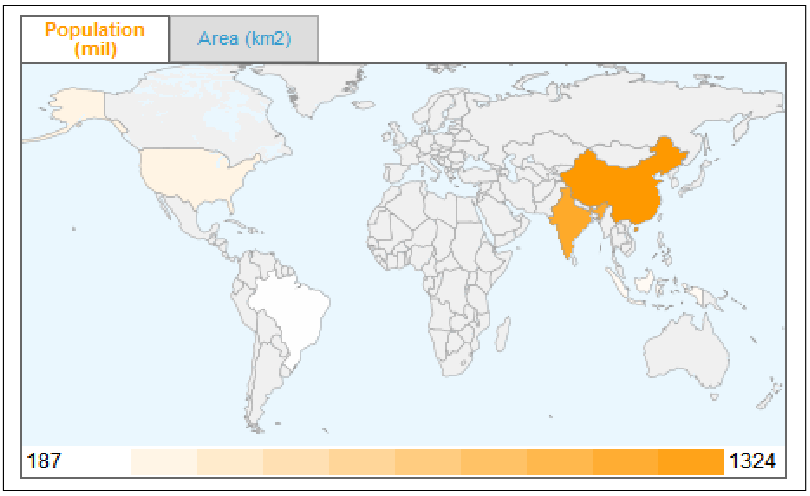
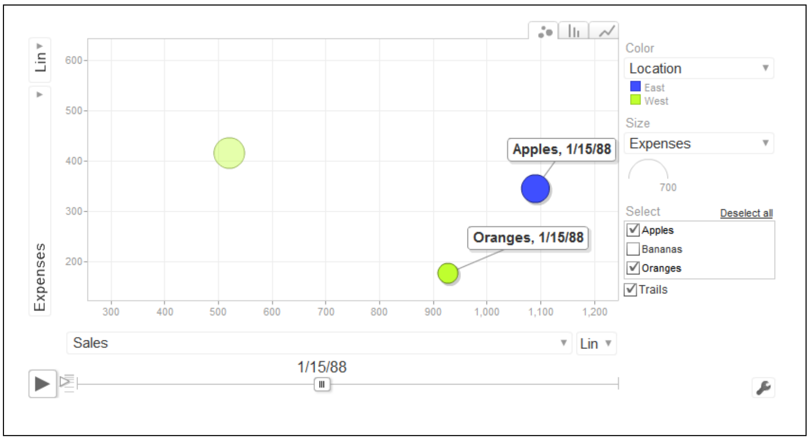


Figure 4.12

2. Motion Chart



Color: Location

Size: Expenses

Select

☑ Apples

□ Bananas

☑ Oranges

☑ Trails

Figure 4.13: Animated visualization of changes to the data over time

4.5 Comparison of Advanced Chart Toolkits

	JIT	**Protovis**	**d3.js**	**Google**
Technology	Canvas	SVG	DOM+SVG	SVG
Maturity	Stable	Stable	In Dev.	Stable
License	New BSD	BSD	BSD	Google
Documentation	++	++	++	++
Examples	+	++	++	++
Updates	Freq.	None	Freq.	Freq.
Popularity	++	**o**	+	+
Browser Compat.	+	- (no IE8)	- (no IE8)	+
Interactivity	++	**o**	+	+ Dashboard
Transitions	++	–	++	**o**
Visual Appeal	++	++	++	**o**
Chord Diagram	++	++	++	–
Arc Diagram	–	++	++	–
Sunburst	++	++	++	–
Tree	++	++	++	–
Motion Chart	–	–	–	++
Tree Map	++	+	–	+
Geo Chart	–	++	++	++
Stream Graph	–	++	++	–
Org Chart	+	-	-	+

Chapter 5

Conclusion

As has been shown, there are many different toolkits to choose from and there is a broad range of technologies being used for visualizing information using Javascript. These technologies include HTML5 (Canvas elements and DOM manipulation), CSS3 and vector graphic formats (SVG, VML). Some of the more advanced toolkits choose the best technology based on the browser used. For example, there may be a backwards compatibility mode which enables support for Internet Explorer 6 via VML, while current generation browser may retrieve the same chart as a HTML5 canvas element and styled by CSS3. During testing it however became evident that the performance of every toolkit varies greatly with the browser used. While some products are more prone to this issue than others, as a general rule, older versions of the Microsoft Internet Explorer (up to version 8) will not be able to handle complex charts regardless of the toolkit chosen. Furthermore, the hardware of the client system apparently has little to no influence on the performance, as long as the browser is able to handle recent standards.

It has to be noted that not all toolkits are suitable for all purposes. Surprisingly, the available commercial solutions appear to be inferior to their free counterparts in most aspects, especially when putting the focus on documentation and support. There is however no single toolkit which offers a solution to all possible problems, because even the most advanced toolkits do not contain all the chart type available in other and simpler products.

The Google Chart Tools are probably the toolkit which is the easiest to learn and use, because the documentation is excellent and the benefits of a simple JSON data format cannot be overestimated. The license may however render them unacceptable for commercial applications. It is therefore recommended, that new projects should be started in either JIT or d3.js, because both of them are not only stable and visually stunning but also open-source and free to use, actively developed, used by a huge community and well documented and maintained.

References

Baranovskiy, Dmitry [2012]. *Raphaël*. `http://raphaeljs.com/`. (Cited on page 4.)

Bostock, Mike [2012]. *D3*. `http://d3js.org`. (Not cited.)

Bostock, Mike and Jeff Heer [2012]. *Protovis*. `http://mbostock.github.com/protovis`. (Cited on page 21.)

D3 [2012]. *D3 Github Wiki*. `https://github.com/mbostock/d3/wiki`. (Cited on page 23.)

Emprise Corporation [2012]. *Emprise Javascript Charts*. `http://www.ejschart.com/`. (Cited on page 13.)

Flot [2012]. *Flot*. `http://code.google.com/p/flot/`. (Cited on page 12.)

Google Inc. [2012a]. *explorercanvas*. `http://code.google.com/p/explorercanvas/`. (Cited on pages 11 and 13.)

Google Inc. [2012b]. *Google API Terms of Service (ToS)*. `https://developers.google.com/terms/`. (Cited on page 26.)

Google Inc. [2012c]. *Google Chart Toolkit*. `https://developers.google.com/chart/`. (Cited on page 26.)

Heyes, Richard [2012]. *RGraph*. `http://www.rgraph.net/`. (Cited on page 10.)

Highsoft Solutions AS [2012]. *Highcharts*. `http://www.highcharts.com`. (Cited on page 9.)

Jacko, Julie A. and Andrew Sears [2003]. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. Routledge. ISBN ISBN 0805844686, 9780805844689. `http://www.amazon.de/The-Human-Computer-Interaction-Handbook-Fundamentals/dp/0805844686`. (Not cited.)

JGraph Ltd. [2012]. *mxGraph*. `http://jgraph.com/mxgraph.html`. (Cited on page 3.)

Sanchalabs [2012]. *JIT Homepage*. `http://thejit.org/`. (Cited on page 19.)

Sizzle [2012]. *Sizzle*. `http://sizzlejs.com`. (Cited on page 25.)

Splunk Inc. [2012]. *Sparklines*. `http://omnipotent.net/jquery.sparkline`. (Cited on page 15.)

Tufte, Edward [2006]. *Beautiful Evidence*. Graphics Pr. ISBN 0299090604. `http://www.amazon.com/Beautiful-Evidence-Edward-R-Tufte/dp/0961392177`. (Cited on page 16.)

Tufte, Edward [2012]. *Inventor of Sparklines*. `http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001OR&topic_id=1`. (Cited on page 16.)