

Aufzeichnungen zum Praktikum Betriebsdatenerfassung

Stefan Härtel

31. Oktober 2011

Inhaltsverzeichnis

1	Allgemeines	3
2	Aufgabe	3
3	Überblick über mehrere Bibliotheken	3
3.1	CanvasXpress	3
3.2	Cube	4
3.3	D3	4
3.4	Dygraphs	5
3.5	Elycharts	5
3.6	Google Chart Tools	5
3.7	Highcharts	6
3.8	InfoVis	6
3.9	Protovis	7
3.10	Andere Bibliotheken	7
4	Entscheidung für eine Bibliothek	7
5	Implementierung	8
6	Treffen	8
6.1	20.10.2011	8
7	Ideen	8
8	Notizen	8

1 Allgemeines

- Praktikumsleiter Steffen Dienst (dienst@informatik.uni-leipzig.de)
- Git: www.github.com/IT-Dude/PVSvis

2 Aufgabe

- Visualisierung einer großen Menge durch Solaranlagen erzeugte Wechselrichterdaten
- Evaluierung mehrerer Javascript-Bibliotheken zur Visualisierung
- interaktive Darstellung im Browser
- Must-haves der Anwendung:
 - Anzeige verschiedener Diagrammtypen (Linien-, Scatter-, Balkendiagramme ...)
 - Verlinkungen zwischen unterschiedlichen Diagrammen
 - Darstellung mehrerer Achsen innerhalb eines Diagramms
 - Tooltips an Messpunkten ...
 - Hilfslinien, Lineale ...
 - Markierung von Werten
 - Ein-/Ausblenden von Messwerten und -reihen

3 Überblick über mehrere Bibliotheken

3.1 CanvasXpress

www.canvasxpress.org

- Grafiken werden im HTML5 Canvas-Element erstellt
- Balken-, Linien-, Kerzen-, Scatter-, Kuchendiagramme, Heatmaps ...
- große Anzahl verschiedener Diagrammtypen
- viele Konfigurationsmöglichkeiten
- Unterstützung von JSON
- verschiedene Diagramme benötigen bestimmte Struktur der Daten (als JSON)
- Erstellung eines Diagramms aus Daten, Konfiguration und Events
- absolute High-Level-Bibliothek
- Zeichnen einfacher Formen möglich
- Dokumentation ist der hohen Abstraktion angemessen
- GNU LGPL

3.2 Cube

www.square.github.com/cube/

- nutzt unter anderem D3, Node.js und MongoDB
- besitzt eine serverseitige Komponente (Collector)
- speziell zur Analyse von Zeitreihen entworfen
- Anforderung von Daten über das WebSocket-Protokoll
- Zusammenstellen von Diagrammansichten über eine interaktive Oberfläche
- eigene kleine Sprache für Anfragen, Filter
- Cube empfängt Events, fertigt Metriken anhand von Anfragen über diese Events an
- Metriken werden zwischengespeichert
- Visualisierung der Anfragen clientseitig
- “Emitters” senden Daten an Cubes “Collectors”
- einige wenige Grundvisualisierungen vorhanden
- Erstellung eigener Visualisierungen möglich (wahrscheinlich mit D3)
- Dokumentation sehr spärlich, fast nicht vorhanden
- Apache License

3.3 D3

www.mbonstock.github.com/d3

- gleiches Entwicklerteam wie Protovis
- basiert auf Konzepten und Erfahrungen von und mit Protovis
- Binden von Daten an Elemente des Dokuments, “Data-Driven Documents” = D3
- Nutzung des Document Object Model, Schnittstelle zum Inhalt eines Dokuments
- direkter Zugriff auf alle Elemente
- doppelte Leistung von Protovis
- Erstellung eigener komplexer, interaktiver Diagramme
- interne Grafikrepräsentation erfolgt direkt über das DOM, eine interne Verwaltung eines Szenengraphen
- “Medium-Level-Bibliothek”: Low-Level-Kernel, High-Level-Hilfsmodule
- Orientierung an Webstandards HTML, SVG, CSS
- eingebauter Parser für CSV-Dateien
- viele Beispiele, gute Dokumentation
- **[TODO: LIZENZ HERAUSFINDEN]**

3.4 Dygraphs

www.dygraphs.com

- auv Visualisierung von Zeitreihen zugeschnitten
- interaktive, zoombare Diagramme
- Visualisierung im Canvas-Element
- Unterstützung von CSV-Dateien
- Darstellung automatisiert (Farbwahl, Auflösung der Skalen ...)
- kompatibel zur Google Visualization API (Datenformat)
- auf eine Liniendiagramm-Art beschränkt
- hoher Abstraktionsgrad
- einige Konfigurationsmöglichkeiten
- sehr kleine Dokumentation
- MIT-Lizenz

3.5 Elycharts

www.elycharts.com

- nutzt jQuery und Raphaël
- mehrere Diagrammtypen
- mehrere Datensätze und Achsen innerhalb eines Diagramms
- viele optische Konfigurationsmöglichkeiten
- Interaktion mit Diagrammen
- Diagramme werden als SVG dargestellt
- optionale Templates für Diagramme
- High-Level-Bibliothek
- MIT-Lizenz

3.6 Google Chart Tools

www.code.google.com/intl/de-DE/apis/chart/

- Grafikdarstellung als SVG
- viele vordefinierte Diagrammtypen
- sehr viele Konfigurationsmöglichkeiten
- interaktive Diagramme, Events

- High-Level-Bibliothek, sehr sehr viele Funktionen
- eigene Sprache für Anfragen, ähnlich SQL
- Unterstützung von JSON, CSV, TSV (tab-separated values) und HTML als Datenformate
- weitreichende Dokumentation
- Google-eigene Lizenz

3.7 Highcharts

www.highcharts.com/products/highcharts

- Grafiken als SVG
- viele Diagrammtypen, vereinbar innerhalb eines Diagramms
- interaktive Diagramme
- Unterstützung mehrerer Skalen und Achsen
- zoombare Diagramme
- Master-Detail-Chart ähnlich Dycharts
- Datenformate: JSON, CSV, XML
- hoher Abstraktionsgrad; Zugriff auf niedere, einfache Zeichenfunktionen
- ausreichende Dokumentation
- kommerzielle Lizenz, Creative Commons Lizenz für nichtkommerzielle Projekt

3.8 InfoVis

www.thejit.org

- viele, komplexere Diagrammtypen (TreeMap, SpaceTree, RGraph)
- erhöhter Grad an Animation und Interaktion, abhängig vom Diagrammtyp
- gute Dokumentation
- JSON nutzbar
- High-Level-Bibliothek
- wenig allgemeine Informationen
- sehr aktive Entwicklung auf GitHub
- Lizenz ???

3.9 Protovis

www.mhstock.github.com/protovis

- wird seit Juni 2011 nicht mehr weiterentwickelt
- Inspiration zu D3, von den gleichen Entwicklern
- nutzt SVG
- Binden von Daten und einfache grafische Elemente, direkte grafische Repräsentation eines Datensatzes
- Bereitstellung von Grundelementen, Erweiterung und Vererbung dieser Elemente
- einige grundlegende Diagrammtypen
- Erstellung eigener, komplexer Visualisierungen
- Erstellung der Visualisierung anhand der Daten
- viel Interaktion
- hohe Komplexität von Visualisierungen und Animationen
- sehr viele Möglichkeiten der Konfiguration
- Medium-Level-Bibliothek
- sehr gute Dokumentation, viele Beispiele, learning by example
- BSD Lizenz

3.10 Andere Bibliotheken

Diese Bibliotheken sind sehr primitiv gehalten beziehungsweise beschränken sich auf Low-Level-Grafikerstellung.

- Flot (www.code.google.com/p/flot)
- Raphaël (www.dmitrybaranovskiy.github.io/raphael/)
- Processing.js (www.processingjs.org)

4 Entscheidung für eine Bibliothek

Grundsätzlich zerfallen die ausgewählten Bibliotheken in zwei Kategorien. Auf der einen Seite sind zusammen D3 und Protovis zu betrachten, auf der anderen Seite CanvasXpress, Cube, Dygraphs, Elycharts, Google Chart Tools, Highcharts und InfoVis. Letztere haben als gemeinsame Eigenschaft, dass sie im Grunde vorgefertigte Visualisierungen bereitstellen, mit denen sich in relativ kurzer Zeit und mit wenig Aufwand gegebene Daten darstellen lassen. Diese Visualisierungen sind meist interaktiv und komplett konfigurierbar. Der Konfigurierbarkeit sind jedoch gewisse Grenzen gesetzt, wodurch eigene Visualisierungen erstellt werden müssen, was aber ein tiefgründiges Wissen über ein Framework oder sogar Eingriffe in solches erfordert. Hervorzuheben sind Cube und Dygraphs, da sie auf die Darstellung von Zeitreihen spezialisiert sind

und somit einen Teil der Projektanforderungen erfüllen. Cube kann effektiv mit der serverseitigen Komponente Daten erfassen und Anfragen beantworten; die graphischen Möglichkeiten des Client sind limitiert, können aber anscheinend auf der Basis von D3 erweitert werden. Abschreckend ist hier besonders die nicht vorhandene Dokumentation. Die von Dygraphs angebotene interaktive Darstellung kann eine Datenreihe in verschiedenen Auflösungen anzeigen, was besonders bei langen Zeitreihen von Vorteil ist. Der Nachteil der Bibliothek liegt darin, dass es nur diese eine Visualisierungsform gibt; für das Erstellen weiterer Diagramme müssen andere Frameworks herhalten. Für das Projekt stehen somit D3 und Protovis in der engeren Auswahl. Beide Bibliotheken verfolgen einen Low-Level-Ansatz, bei dem Daten direkt an deren grafische Repräsentation gebunden wird. Protovis wird nicht mehr aktiv weiterentwickelt; D3 greift jedoch die Konzepte von Protovis auf, erweitert diese und bringt viele Verbesserungen ein. So kann D3 ohne Umweg ein Dokument über das DOM ändern, skaliert besser und legt eine gesteigerte Performance an den Tag; dabei ist D3 allerdings nicht als Protovis 2 zu verstehen. Die für das Projekt verwendete Bibliothek wird D3 sein. Es ist eine aktuelle Technologie und ermöglicht auf einer eigenen Ebene die Erstellung interaktiver und attraktiver Visualisierungen, welche speziell auf die Anforderungen des Projekts zurechtgeschnitten werden können. Das Fehlen konkreter vorgefertigter Diagramme ist hierbei nicht als Nachteil zu sehen, sondern deutet auf die große Flexibilität der Bibliothek hin. Die Lernkurve von D3 ist gegenüber den anderen Frameworks sehr viel steiler, es wird aber eine gute Zugänglichkeit zur grundlegenden Technik geboten, was die Entwicklung und die Qualität des Projekts sicherlich gut beeinflusst.

5 Implementierung

- interessantes Konzept bei Dycharts

6 Treffen

6.1 20.10.2011

- Besprechung der Aufgabenstellung
- Festlegung der Must-haves

7 Ideen

- dunkler Hintergrund, helle Visualisierung → mehr Energie
- Sonne nachahmen ???

8 Notizen

- Visualisierung großer Zeiträume, geeignete Methoden?
- `bl.ocks.org`

- jsfiddle.net
- Github Pages
- Processing
- www.sixrevisions.com/javascript/20-fresh-javascript-data-visualization-libraries
- www.splashnology.com/article/15-awesome-free-javascript-charts/325/
- www.flowingdata.com/2011/03/09/data-driven-documents-for-visualization-in-the-browser/
- www.flowingdata.com/2010/01/07/11-ways-to-visualize-changes-over-time-a-guide
- www.dekstop.de/weblog/2011/09/lastfm_heatmap_calendars
- Heatmap Kalender
- www.datavis.dekstop.de/last.fm_heatmap_calendars