

Serverseitige Generierung von dynamischen Scalable Vector Graphics (SVG) zur Darstellung von Businessgrafiken

- Seminararbeit -

Alexander Lindhorst

Situation

- Steigende Verbreitung des Internet sorgte für erhöhte Nachfrage nach dynamischen Informationen
- Grafiken als Informationsdarstellung mit besonders hoher Aussagekraft
- Entwicklung von Komponenten zur Bilderzeugung
- Heute: Meistens Rastergrafiken

Rastergrafiken

- Rastergrafiken haben einige Nachteile:
 - Speicherung von Pixelwerten
 - Unlesbar
 - Manipulation/Transformation schwierig
 - Statisch
 - Nicht skalierbar
- Namhafte Hersteller entwickeln XML-basiertes Grafikformat zur Vermeidung dieser Nachteile: SVG

Motivation

- Aufzeigen von Vorteilen bei der Verwendung von SVG
- Beispiel: Businessgrafiken
- Implementierung eines Toolkits
- Implementierung serverseitiger Komponenten, die das Toolkit verwenden

Grundlagen von SVG: Aufbau einer Datei

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  'http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd'>
<svg width="190" height="80" version="1.0"
  xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" font-
  family="&apos;sansserif&apos;">
<g>
<rect x="10" y="10" width="50" height="50" rx="2" ry="2"
  style="stroke:black;stroke-width:1mm;fill:blue;"/>
<circle cx="95" cy="35" r="25" style="stroke:black;stroke-
  width:1mm;fill:red;"/>
<path d="M130 60 150 0 1-25 -50 1-25 50 z"
  style="stroke:black;stroke-width:1mm;fill:green;"/>
</g>
</svg>
```



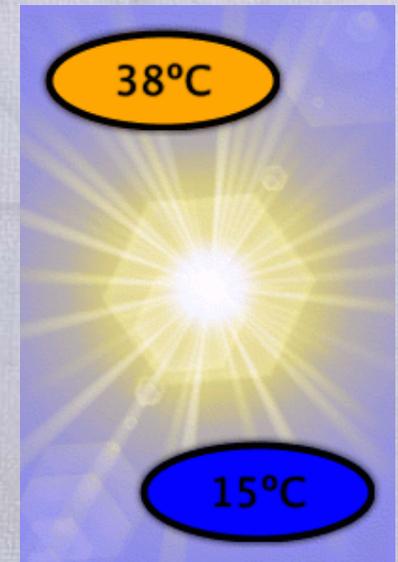
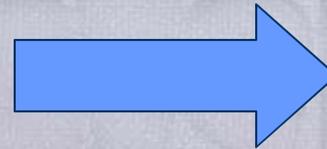
SVG: Besondere Eigenschaften

- Textbasiert
 - Kann von Menschen gelesen werden
 - Kann von „Laien“ manipuliert werden
 - Informationen extrahierbar
- Wohlgeformt
 - Maschinelle Verarbeitung
 - „Interpretation“ unnötig
- Validierbar
 - Verlässliche Verarbeitung
 - Ablehnung möglich, wenn nicht alle Informationen darstellbar
- Gruppen-Tag: Semantik umsetzbar

SVG: Besondere Eigenschaften (II)

- Transformierbar
 - Regelbasierte Manipulation
 - XML -> SVG
 - SVG -> „beliebiges“ Zielformat

```
<?xml version="1.0" encoding="ISO8859_15"?>  
<weather>  
  <sky appearance="sunny" />  
  <mintemp>15&deg;</mintemp>  
  <maxtemp>38&deg;</maxtemp>  
</weather>
```



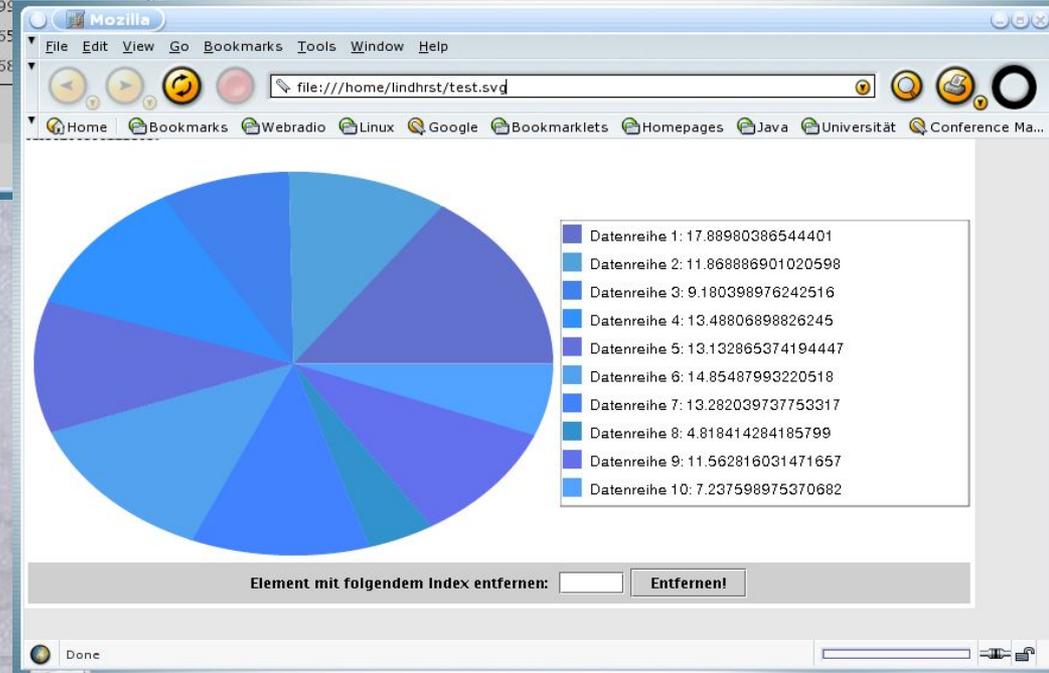
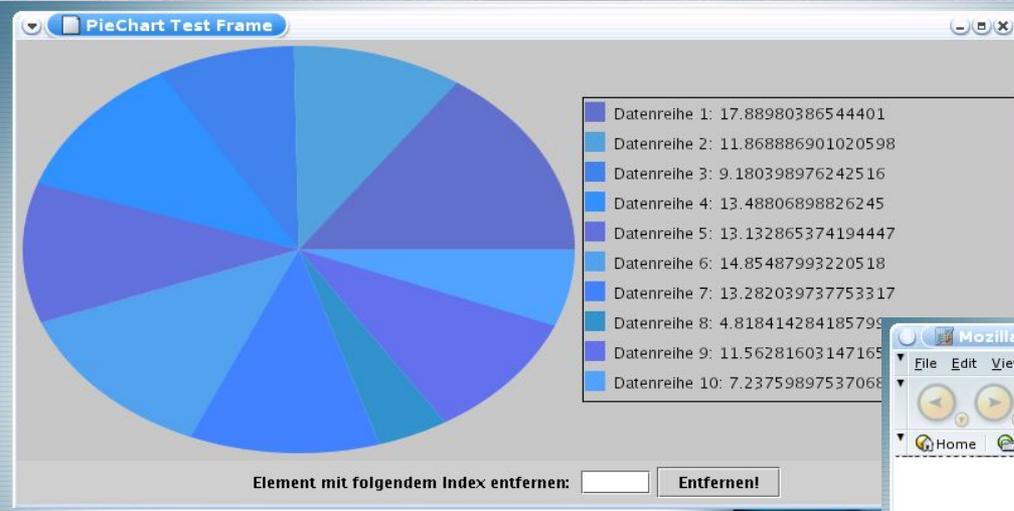
SVG: Besondere Eigenschaften (III)

- Unterstützung von Animationen
- Vollständiges Document Object Model (DOM)
 - Dynamischer Elementezugriff
 - Internes und Externes Scripting
 - Event Handling Mechanismus

Toolkits: Das Apache Batik-Projekt

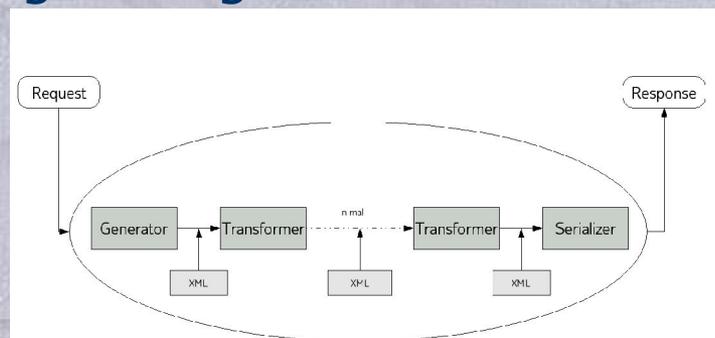
- Open Source Implementierung der SVG-Spezifikation
- Drei Hauptkomponenten:
 - SVG-Viewer; Klassen zum Rendern von SVG-Grafiken
 - Encoder: Klassen zur Überführung von SVG-Grafiken in das JPEG- oder PNG-Format
 - SVG-Export: Klassen zur Überführung von Java2D-Zeichenbefehlen in SVG-Code

SVG-Export mit Apache Batik



Web-Techniken

- Java-Servlets
 - Java basierte Technologie
 - Entstehung als Ersatz für CGI
 - Umgehung von Performanz- und Sicherheitsschwachstellen
- Cocoon
 - Java-basiertes Publishing-Framework
 - Separation of Concerns durch (XML)
 - Beliebige Ausgabeformate



Umsetzung

- Trennung in 2 Bereiche
 - SVG-Generierung: Framework
 - Serverseitige Komponenten
- Zusammenführung: Serverseitige Komponenten benutzen Framework

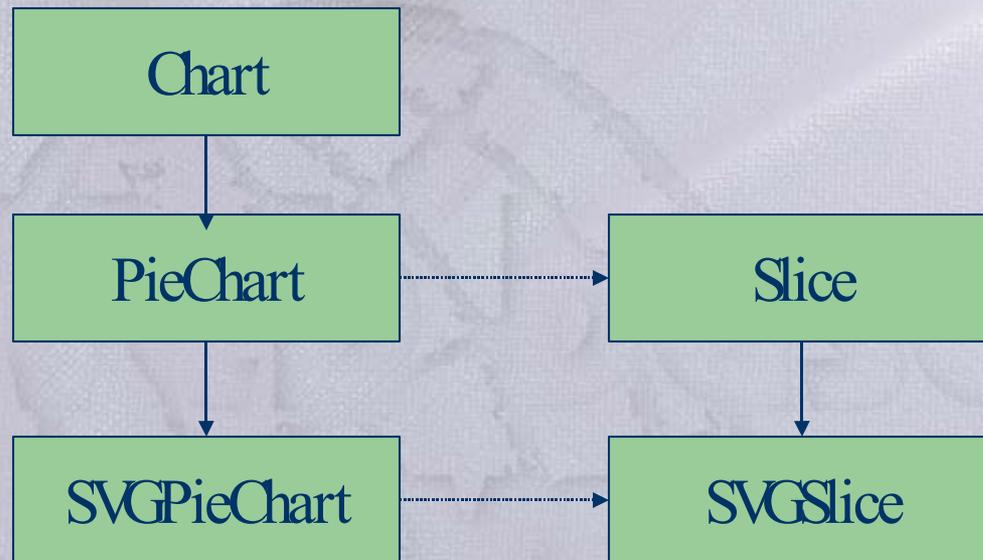
Framework

- Unterstützung von
 - Tortendiagramm
 - Säulendiagramm (2D und 3D)
 - Liniendiagramm (2D und 3D)
- Basiert auf Batik-Projekt

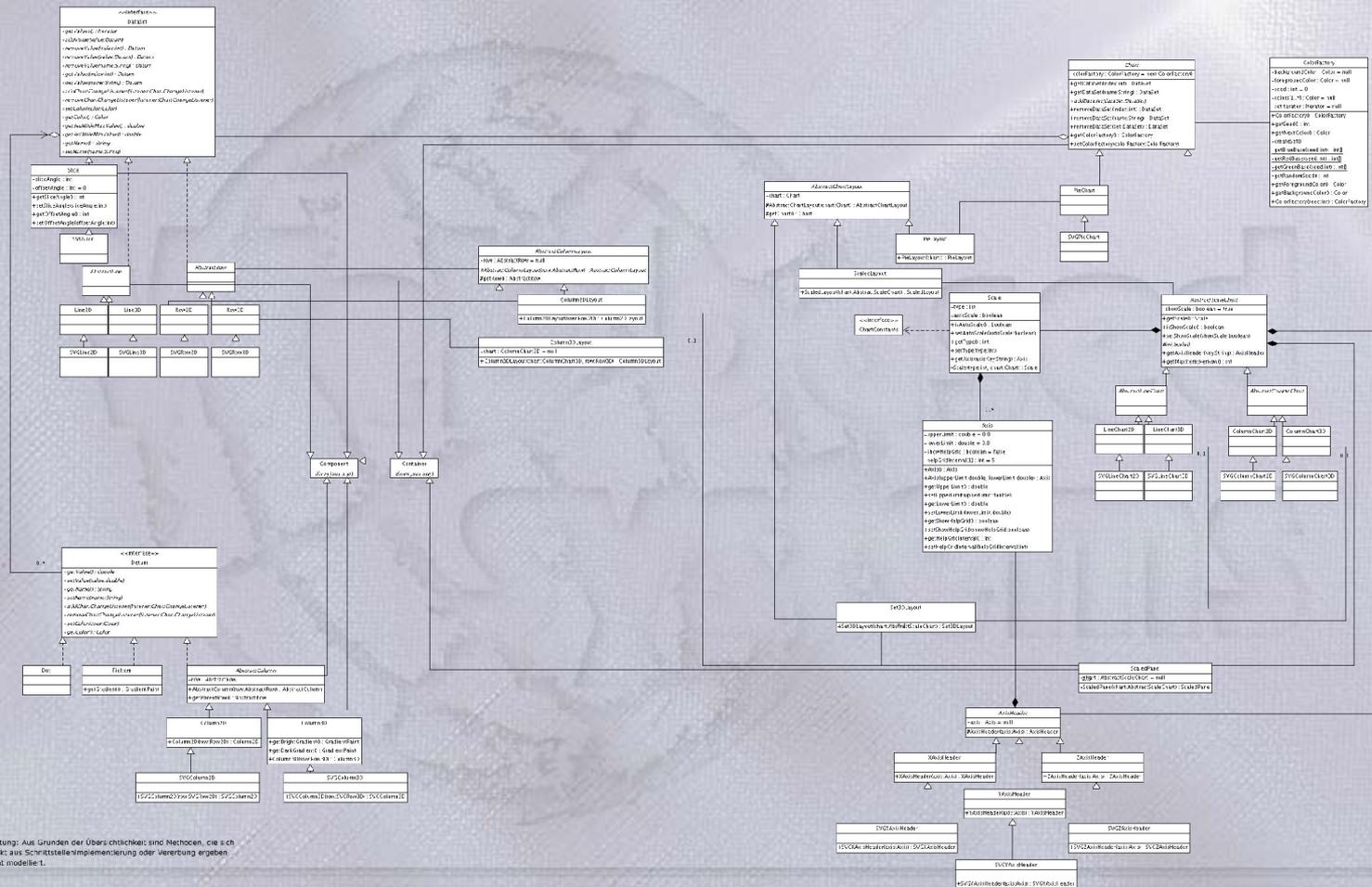
Implementierung des Frameworks

- Zunächst AWT-Framework
- Mit Batik-Projekt Erweiterung zur Generierung von SVG-Code
- Hinzufügen gewünschter Eigenschaften mit DOM
- Ansatz: Businessgrafik Darstellung von Reihen mit Daten
- Übertragung des Ansatzes auf Grafiktypen
 - Abstraktion
 - Ausbaufähigkeit

Vererbung im Framework



Framework



Achtung: Aus Gründen der Übersichtlichkeit sind Methoden, die sich direkt aus Schrittstapelimplementierung oder Vererbung ergeben, nicht modelliert.

Serverkomponenten

- Cocoon-Transformer
 - Zeichnet mit Framework
 - Benutzt DOM zur Generierung von SAX-Events
- Servlet
 - Generiert SVG mit Skript
 - Dynamische Informationsanzeige beim Client

Fazit

- SVG hat hohe Eignung für dynamische Generierung
 - Dynamisch
 - Hohe Qualität
 - Interaktiv
- Batik-Projekt als mächtiges Werkzeug
 - Einfache Transformation
 - Nachträgliche Änderung über DOM