

Scalable Vector Graphics (SVG) – ein XML-basierter Grafikstandard für 2D-Vektorgrafiken

Dr. Thomas Meinike
thomas.meinike@et.fh-merseburg.de

Fachhochschule Merseburg
Fachbereich Elektrotechnik, Informationstechnik und Medien
Studiengang „Kommunikation und Technische Dokumentation“



Überblick

- ⇒ Was ist SVG?
- ⇒ XML auf einen Blick
- ⇒ Aufbau von SVG-Dokumenten
- ⇒ Vorteile und Einsatzgebiete
- ⇒ Software mit SVG-Unterstützung
- ⇒ Praktische SVG-Beispiele
- ⇒ Zusammenfassung und Ausblick




Was ist SVG?

- ⇒ SVG 1.0 ist eine im September 2001 verabschiedete W3C-Spezifikation zur Beschreibung von 2D-Vektorgrafiken in XML-Syntax (Entwicklung seit 1999, 1.1 in Arbeit).
- ⇒ Dieses Format soll langfristig die gegenwärtig im Internet verwendeten Grafikformate ersetzen (GIF, JPEG, PNG).
- ⇒ SVG ermöglicht das Erstellen strukturierter Dokumente und Verarbeitung im Umfeld anderer XML-Technologien.
- ⇒ SVG-Dokumente lassen sich statisch und dynamisch generieren und relativ einfach in Server-seitige Anwendungen integrieren. ←



Vektor- vs. Pixelgrafik

- ⇒ Pixelgrafiken bestehen aus den individuellen Bildpunkten, d. h. Rasterposition und Farbzuzuweisung (bei True Color $3 \times 8 = 24$ Bit (3 Bytes) für Rot-, Grün-, Blau-Werte).
- ⇒ Beispiel: Kreis mit Durchmesser = 100 Pixel ergibt 10000 Pixel, als BMP ca. 30000 Bytes: 
- ⇒ in einem SVG-Dokument belegt dieser Kreis 50 Bytes:

```
<circle cx="50" cy="50" r="50" style="fill: red"/>
```
- ⇒ Vektorgrafiken beschreiben also nicht einzelne Pixel, sondern geometrische Elementarobjekte („Primitive“).
- ⇒ Daraus folgt beliebige und verlustfreie Skalierung. ←



SVG im Überblick

Objekte und Effekte in SVG

[Der rote Kreis, die Erklärungstexte und die Textlinks sind mit JavaScript-Funktionen verknüpft.]



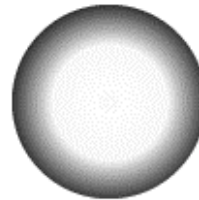
Rechteck



linearer Gradient



Kreis



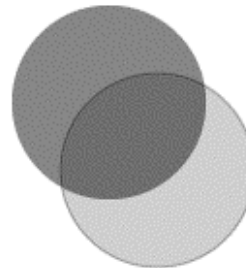
radialer Gradient



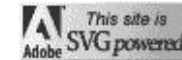
Gruppe+Transformation



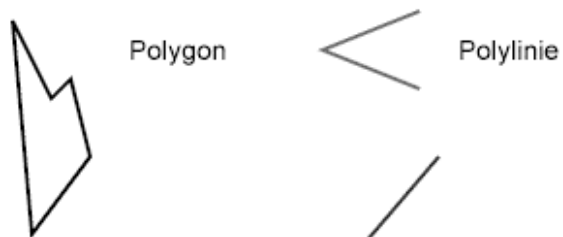
Ellipse



Opazität (Durchlässigkeit)

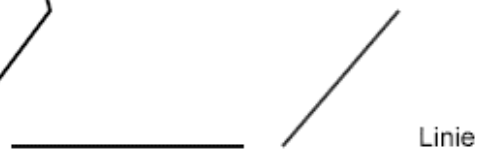


externes Bild



Polygon

Polylinie



Linie



Animation



Pfad



Spezialfilter

<http://www.StyleAssistant.de>

Textlink



Muster

normaler Fließtext

Warum XML?

- ⇒ XML ist ein W3C-Standard zur Definition eigener Auszeichnungssprachen für **strukturierte Dokumente** mit Trennung zwischen Inhalt und Formatierung (Layout).
- ⇒ Im Kontext mit anderen XML-Technologien wie XSLT, XPath, XLink und XPointer kann die Verarbeitung von XML-Daten universell und plattformunabhängig erfolgen.

XSLT-Prozessor



Beispiel XSLT:

- ⇒ XML-Daten + XSL-Formatvorlage --> Output (Text, HTML, XML)



XML auf einen Blick

⇒ Aufbau eines XML-Dokumentes

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE wurzelement [SYSTEM|PUBLIC] "doctype.dtd">

<wurzelement>

  <element1 attribut1="wert">

    <element2>Inhalt</element2>

  </element1>

  <!-- weitere Unterelemente und Inhalte -->

</wurzelement>
```



Aufbau von SVG-Dokumenten (1)

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"  
  "http://www.w3.org/TR/2001/REC-SVG-20010904/  
  DTD/svg10.dtd">  
  
<svg xmlns="http://www.w3.org/2000/svg">  
  
  <title>optionaler Titel</title>  
  <desc>optionale Beschreibung</desc>  
  
  <!-- weitere SVG-Inhalte -->  
  
</svg>
```



Aufbau von SVG-Dokumenten (2)

⇒ Grafische Grundformen wie Rechteck, Kreis, Ellipse, Linie, Polylinie, Polygon:

```
<rect x="..." y="..." width="..." height="..." />
```

```
<circle cx="..." cy="..." r="..." />
```

```
<ellipse cx="..." cy="..." rx="..." ry="..." />
```

```
<line x1="..." y1="..." x2="..." y2="..." />
```

```
<polyline points="x1,y1,...,xn,yn" />
```

```
<polygon points="x1,y1,...,xn,yn" />
```

⇒ Textinhalte:

```
<text x="..." y="...">Textinhalt</text>
```

Aufbau von SVG-Dokumenten (3)

⇒ Formatierung über Stylesheets (CSS)

- roter Kreis mit blauem Rand der Stärke 2 Pixel:

```
<circle cx="100" cy="200" r="50" style="fill:
  #FF0000; stroke: #0000CC; stroke-width: 2px"/>
```

- grüner Text, 14 Pixel hoch in der Schriftart Arial:

```
<text x="20" y="50" style="fill: #009900;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px">Text in SVG</text>
```

- interne bzw. externe Stylesheet-Definitionen (z. B. Klassen):

```
<defs><style type="text/css">
  <![CDATA[ /* CSS-Definitionen */ ]]>
</style></defs>
```

```
<?xml-stylesheet href="styles.css" type="text/css"?>
```

Aufbau von SVG-Dokumenten (4)

⇒ Pfade (Kurvenzüge, Bögen)

- elliptischer Bogen:

```
<path d="M 200,200 L 200,100 A 100,100 0 0,1  
289,154 Z" style="fill: #FFFF00"/>
```

[M: moveto, L: lineto, A: elliptical arc, Z: closepath]

⇒ Animationen

- Rechteckbreite erreicht innerhalb von 10 Sekunden 180 Pixel:

```
<rect x="600" y="360" width="0" height="20"  
style="fill: #F00;"><animate attributeType="XML"  
attributeName="width" begin="0s" dur="10s"  
fill="freeze" from="0" to="180"/></rect>
```



Aufbau von SVG-Dokumenten (5)

⇒ Gradienten (lineare und radiale Farbverläufe)

```
<defs>
  <linearGradient id="lingra1">
    <stop offset="0%" style="stop-color: #000"/>
    <stop offset="50%" style="stop-color: #00F"/>
    <stop offset="100%" style="stop-color: #FFF"/>
  </linearGradient>
</defs>

<rect x="400" y="70" width="180" height="30"
  style="fill: url(#lingra1)"/>
```

⇒ Filter (u. a. bekannt aus Adobe Photoshop/Illustrator)

Filterelemente sind relativ komplex, z. B. `feGaussianBlur`.



Aufbau von SVG-Dokumenten (6)

⇒ Textlinks (ähnlich zu Links in HTML)

```
<a xlink:href="http://ktd.et.fh-merseburg.de">  
  <text x="100" y="200" style="fill: #F00">  
    KTD-Website  
  </text>  
</a>
```

⇒ externe Bilder (ähnlich zu Bildern in HTML)

```
<image x="300" y="400" xlink:href="einbild.jpg"  
  width="..." height="..."/>
```



Aufbau von SVG-Dokumenten (7)

- ⇒ Gruppierung (Zusammenfassung von Objekten zu Gruppen)
- ⇒ Transformationen (Rotation und Translation)

– Gruppe aus Rechteck und Text wird um -30° gedreht:

```
<g transform="rotate(-30)">  
  <rect x="620" y="500"  
    rx="10" ry="10" width="100" height="50"  
    style="fill: none; stroke: #F00; stroke-width:  
    2px"/>  
    <text x="635" y="537" style="font-size: 36px;  
      fill: none; stroke: #00C; stroke-width:  
      1px">Textinhalt</text>  
</g>
```



Aufbau von SVG-Dokumenten (8)

⇒ Einsatz von Scripts:

- Einbindung im <defs>...</defs>-Bereich (auch externe Dateien möglich)
- Sprachumfang von JavaScript bzw. ECMAScript verwendbar
- Zugriff auf eigene Funktionen über Event-Handler wie in HTML

```
<circle id="kreis1" cx="100" cy="200" r="50"
style="fill: #FF0000; stroke: #0000CC; stroke-width:
2px" onmouseover="NeueFarbe('kreis1', '#FFFF00') "
onmouseout="NeueFarbe('kreis1', '#FF0000')"/>
```

```
function NeueFarbe(objid, fillcol)
{
  var svgdoc, element;
  svgdoc=document.documentElement;
  element=svgdoc.getElementById(objid);
  element.getStyle().setProperty("fill", fillcol);
}
```

Aufbau von SVG-Dokumenten (9)

⇒ SVG-Einbindung in HTML:

über das Element `object` mit Alternativinhalt (ggf. mittels `embed`)

```
<object data="datei.svg" width="..." height="..."  
type="image/svg+xml" >
```

```
<p>Zur Anzeige der Grafik wird der <a href=  
"http://www.adobe.com/svg/viewer/install/main.html">  
SVG Viewer von Adobe</a> benötigt!</p>
```

```
</object>
```



Vorteile und Einsatzgebiete

⇒ Vorteile:

- hohe Auflösung/Qualität, Detailansicht durch „Zoomen“
- kleine(re) Dateigröße gegenüber Rastergrafiken
- dyn. Interaktivität durch Scripting, DOM-Integration
- auf jeder Plattform mit XML-Werkzeugen erstellbar
- [bei nativer Browser-Unterstützung] universell einsetzbar

⇒ Einsatzgebiete:

- Kartografie
- Technisches Zeichnen / CAD, Illustration
- Messtechnik, Datenverarbeitung
- Cross-Media- bzw. Single-Source-Publishing auf XML-Basis
- wissenschaftlich-technische Anwendungen ←



Software mit SVG-Unterstützung

⇒ Grafikprogramme:

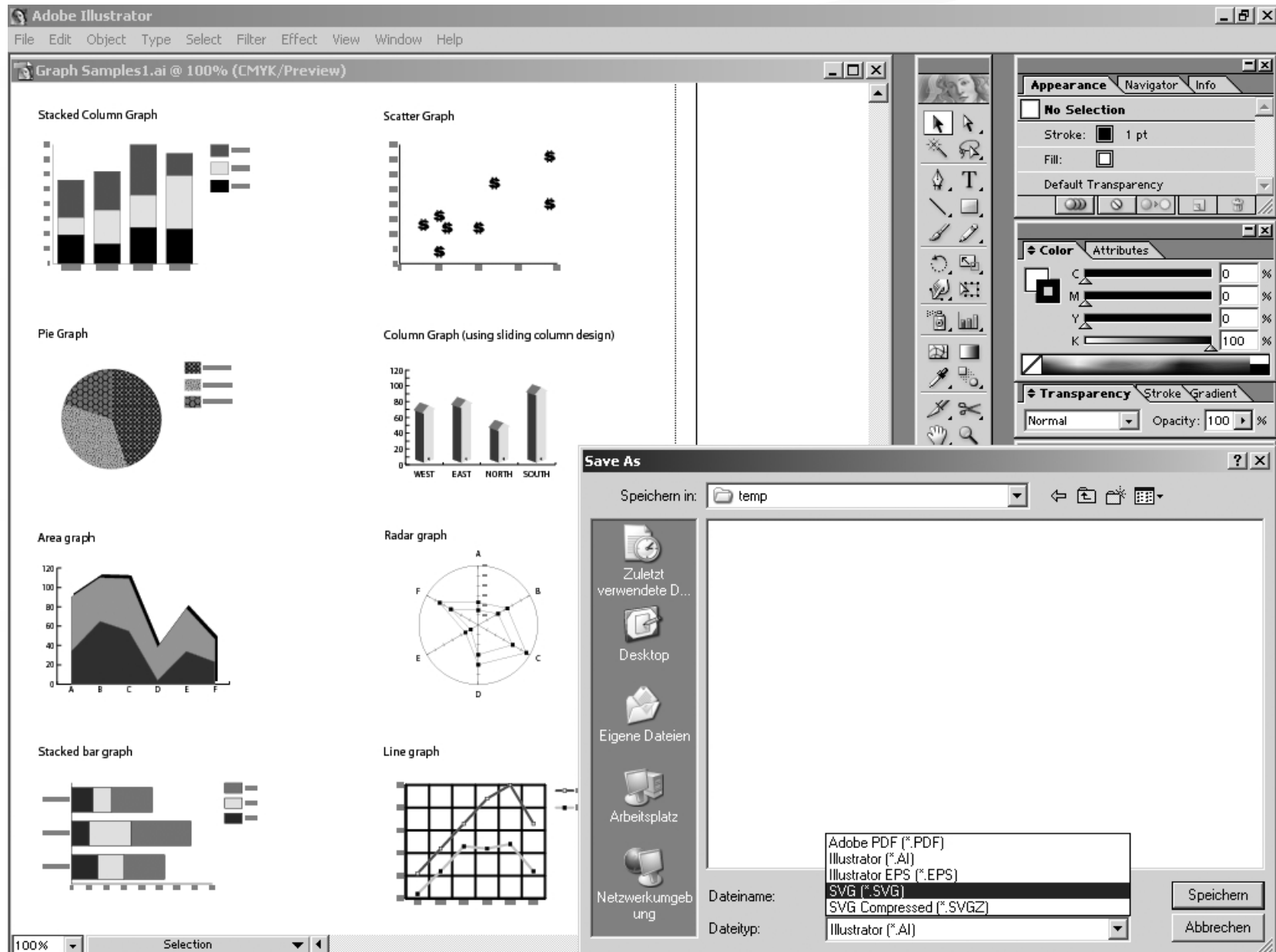
- Adobe Illustrator
- Jasc WebDraw
- sphinx SVG
- MediaChance Real-DRAW
- PCX Software SVG Studio

⇒ Tools:

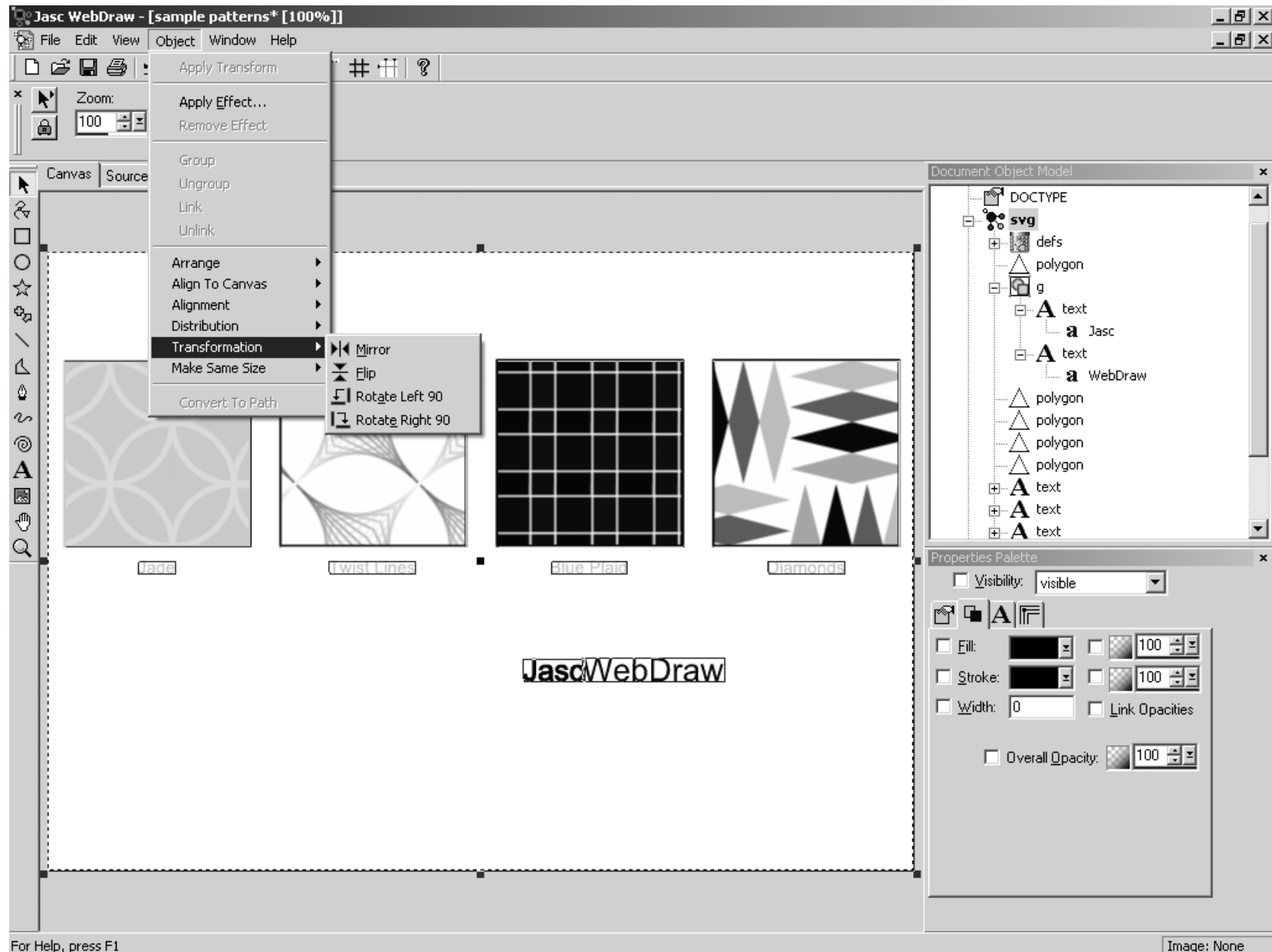
- Adobe SVG Viewer (Browser-PlugIn)
- SVG maker („Druckertreiber zur SVG-Ausgabe“)
- Batik SVG Toolkit (JAVA)
- SVG Composer
- Amaya (Editor/Browser vom W3C)



Adobe Illustrator 10

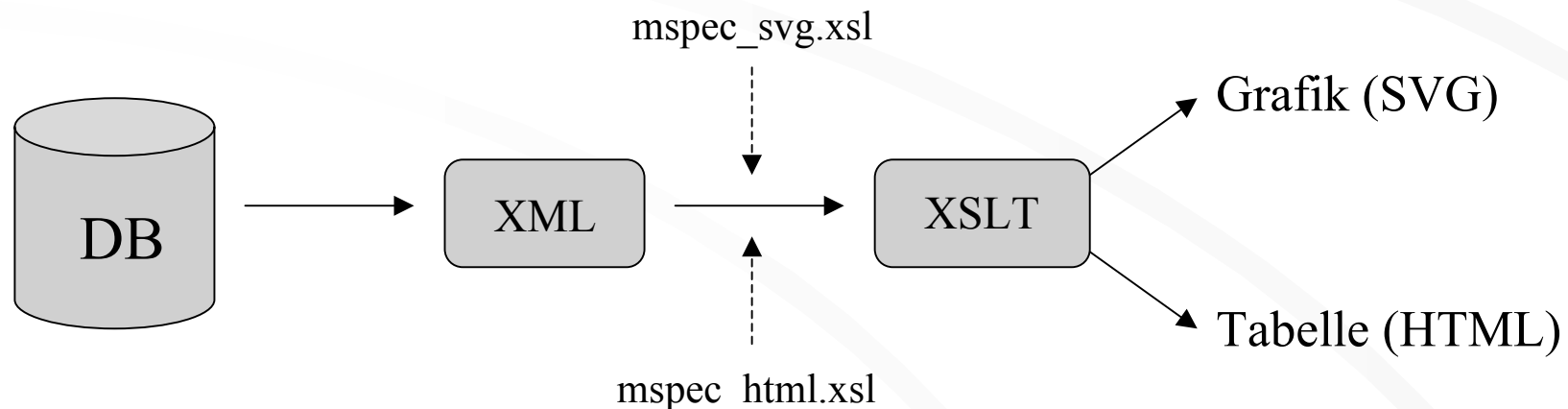


Jasc WebDraw 1.0x



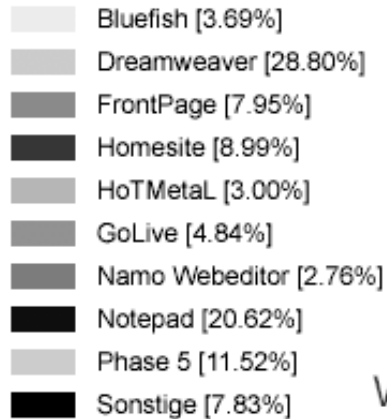
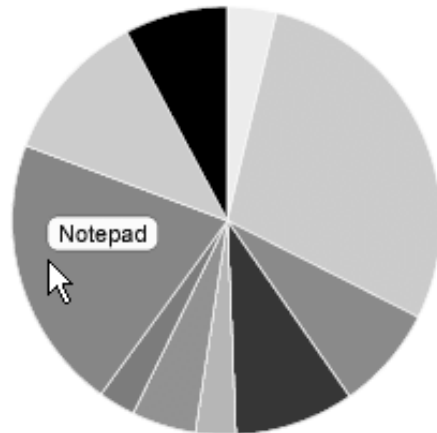
Praktische SVG-Beispiele

- ⇒ PHP-Funktionsbibliothek für Kreis- und Balkendiagramme
- ⇒ Generierung von Massenspektren im SVG-Format
 - Spektrendaten in MySQL-Datenbank
 - PHP als Programmiersprache
 - Datenverarbeitung auf der Basis von XML/XSLT
 - Ausgabe wahlweise als Grafik oder Tabelle



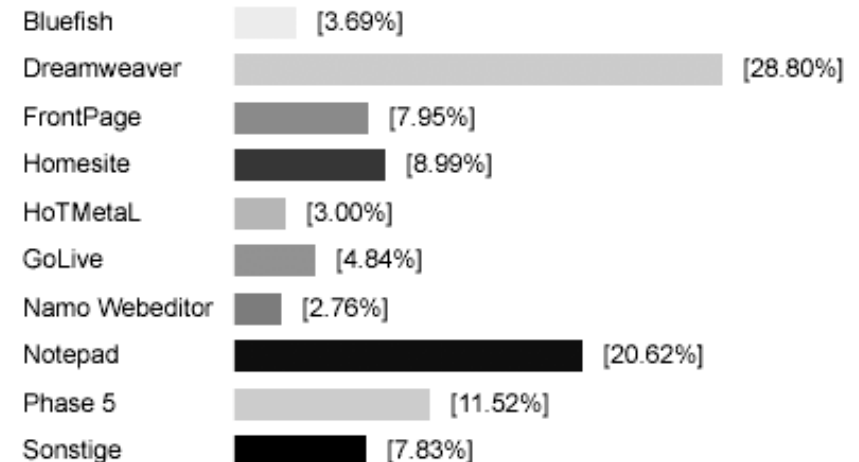
SVG::PHP

Web-Editoren



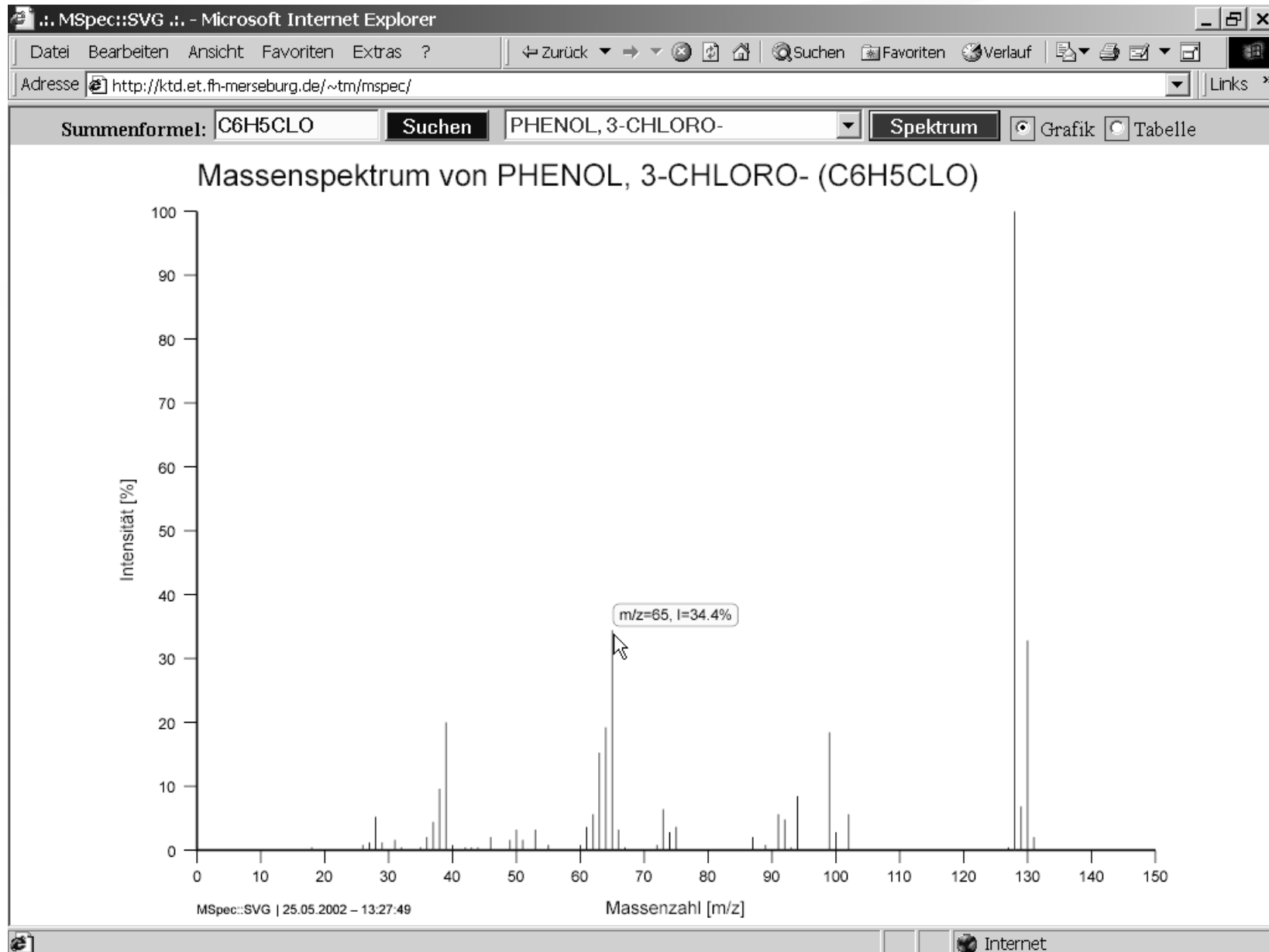
kreisdiagramm.svg – dynamisch generiert: 19.05.2002 – 16:17:52
SVG::PHP-Modul 1.00 – © by Dr. Thomas Meinike 2002

Web-Editoren



balkendiagramm.svg – dynamisch generiert: 19.05.2002 – 16:20:55
SVG::PHP-Modul 1.00 – © by Dr. Thomas Meinike 2002

MSpec::SVG



Zusammenfassung und Ausblick

- ⇒ Grafiken im SVG-Format sind strukturierte XML-Dokumente.
- ⇒ Daraus resultieren vor allem neue Möglichkeiten bei der Erstellung und Verarbeitung von Inhalten.
- ⇒ Die erstellten Grafiken lassen sich wieder in die zugrunde liegenden Daten (z. B. Koordinaten) zurückführen und durchsuchen (wichtig für Suchmaschinen).
- ⇒ Plattformunabhängige Entwicklungen auf der Basis offener bzw. freier Standards und Technologien sind im SVG-Umfeld realisierbar.
- ⇒ Wichtiges Etappenziel ist eine verbesserte SVG-Unterstützung in den Web-Browsern. ←



Einige Online-Ressourcen zu SVG

- ⇒ <http://www.w3.org/Graphics/SVG/Overview.html>
- ⇒ <http://www.w3.org/TR/SVG/>
- ⇒ <http://www.scale-a-vector.de>
- ⇒ <http://www.jasc.com/products/webdraw/>
- ⇒ <http://www.adobe.com/svg/>
- ⇒ <http://xml.apache.org/batik/>
- ⇒ <http://www.mozilla.org/projects/svg/>
- ⇒ <http://www.styleassistant.de/tips/TMs10kSVGDemo.htm> (*)
- ⇒ <http://www.datenverdrahten.de/svgphp/> (*)
- ⇒ <http://ktd.et.fh-merseburg.de/~tm/mspec/> (*)
- ⇒ <http://www.zvon.org/xxl/svgReference/Output/>
- ⇒ http://dmoz.org/Computers/Data_Formats/Graphics/Vector/SVG/

(*) = Beispiele
zum Vortrag

