

SVG und SMIL

Einführung und Überblick



Georg Jahn
mail@gjahn.com

SVG und SML: Einführung und Überblick

Inhalt

- Scalable Vector Graphics (SVG)
 - Einführung und Entstehung
 - Aufbau und Elemente
 - Praktischer Teil: Hands-On und Tools
- Synchronized Multimedia Integration Language (SMIL)
 - Einführung und Entstehung
 - Aufbau und Elemente
 - Beispiel und Unterstützung
- Exkurs: SVG und XSLT

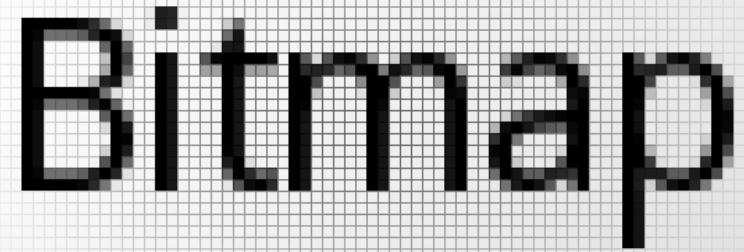
SVG

Scalable Vector Graphics

SVG – Einführung und Entstehung

Prinzip der Vektorgrafik

- Bitmap / Rastergrafik
 - „Karte“ der Farben für Pixel.
 - Inhärent pixelorientiert.
- Vektorgrafik
 - Grafik wird durch ihre Linienzüge, Pfade, Text, ... beschrieben.
 - Das Koordinatensystem ist pixelunabhängig.



Bitmap

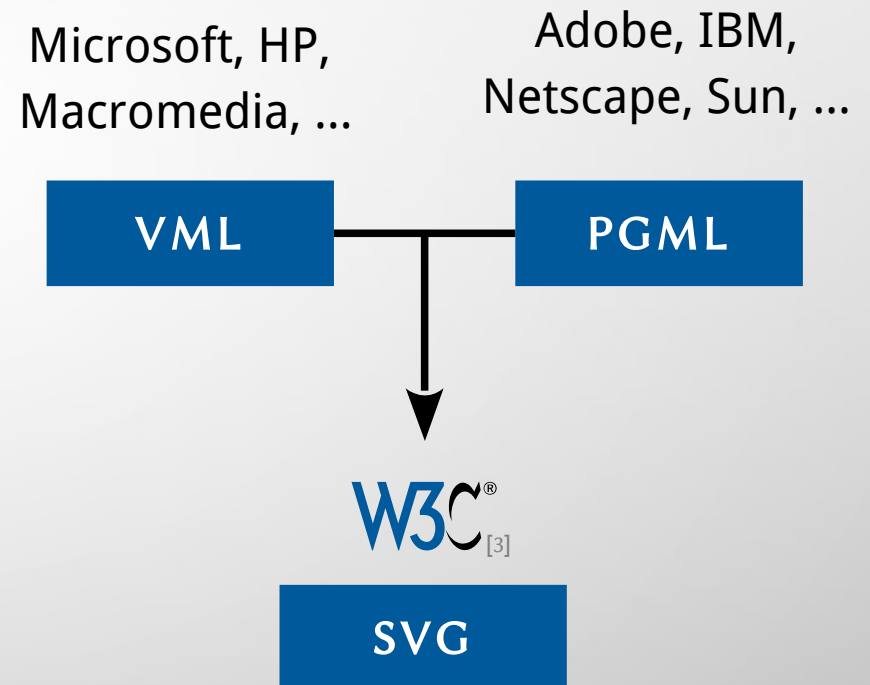


Vector

SVG – Einführung und Entstehung

Entstehung

- 1980er: PostScript (Adobe)
 - Turing-Vollständigkeit:
→ PostScript-Fehler
 - Schlechte Performance und hohe Dateigrößen.
- 1998: Konkurrierende XML-basierte Sprachen beim W3C eingereicht.
- 1999: Das W3C entwickelt daraus den neuen offenen Standard SVG.



SVG – Einführung und Entstehung

Versionen

- 2001: **SVG 1.0** wird Empfehlung des W3C.
- 2003: **SVG 1.1** wird Empfehlung des W3C.
 - Modularisierter Aufbau. Dies erlaubt die Einführung von Profilen, die nur eine Teilmenge definieren:



SVG 1.1 Tiny

SVG 1.1 Basic

SVG 1.1 Full



- Ansonsten nur geringe Unterschiede zu SVG 1.0.
- 2008: **SVG 1.2 Tiny** wird Empfehlung des W3C.
 - Entstand über Umwege aus SVG 1.1 Tiny, SVG 1.2 Full bleibt aus.
- 2014: **SVG 2.0** wird voraussichtlich veröffentlicht.
 - Anpassung an neue Technologien wie HTML5, CSS 2, WOFF, ...

SVG – Einführung und Entstehung

Features

- XML 1.0-basiertes Format.
 - **Vektorelemente**: Gängige Primitive bis allgemeine Pfade.
 - **Textelemente**: Verwendung externer Schriftarten.
 - **Eingebundene Elemente**: U.a. Rastergrafiken.
 - **Styling, Clipping** und **Filtereffekte** können angewendet werden.
- Explizit zur Verwendung mit XML-Namespaces gedacht:
 - Interaktivität und Animation durch **SMIL**.
 - Einbettbar in **HTML, XHTML, ...**
- Kompatibel mit CSS, XSL, XML ID Referencing, XML Associated Style Sheets, DOM Level 1, ...

SVG – Einführung und Entstehung

Limitierungen

- Kann nur eine Grafik beinhalten.
 - Für gesamte Dokumente: [PDF](#), [SVG Print](#) (Entwurf)
- Nur sehr grundlegende Animationen direkt möglich.
 - Erweiterte Möglichkeiten bietet der Gebrauch von [SMIL](#).
 - Über [Scripting](#) kann auf das DOM zugegriffen werden.
- Unterstützt kein Z-Indizes.
 - Allein die Reihenfolge der Elemente zählt.
- Datendichte durch XML-Struktur nicht besonders hoch.
 - Mit gzip komprimiertes SVGZ-Format typischerweise $\frac{1}{4}$ der Größe.

SVG – Aufbau und Elemente

Das SVG-Fragment

```
<svg xmlns="http://www.w3.org/2000/svg">  
  version="1.1"  
  baseProfile="tiny|basic|full"  
  viewBox="0 0 1024 768"  
  x="0" y="0" width="12cm" height="9cm">  
  ... </svg>
```

- Die Elemente in einem SVG-Element bilden ein SVG-Fragment.
- Das SVG-Element definiert dabei das Koordinatensystem innerhalb (`viewBox`) und die Anzeige außerhalb (`x`, `y`, ...).
- `baseProfile` zeigt an, welches SVG-Profil mindestens benötigt wird, um den Inhalt zu rendern.

SVG – Aufbau und Elemente

Das SVG-Dokument

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg.dtd">  
<svg xmlns="http://www.w3.org/2000/svg"  
  width="4cm" height="3cm">  
  
  <desc>Leeres SVG-Dokument</desc>  
  
</svg>
```

- Das SVG-Fragment kann in einem SVG allein stehen, indem es mit gültiger XML- und DocType-Deklaration versehen wird.

SVG – Aufbau und Elemente

SVG in anderen XML-Dokumenten

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:svg="http://www.w3.org/2000/svg">
  <body>
    <svg:svg width="100px" height="75px">
      <svg:desc>Leeres SVG-Dokument</svg:desc>
    </svg:svg>
  </body>
</html>
```

- Es kann auch über den Namespace-Mechanismus in andere XML-Dokumente eingebettet werden.

SVG – Aufbau und Elemente

SVG in anderen XML-Dokumenten

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" ...>
<html xmlns="http://www.w3.org/1999/xhtml">
  <body>
    <svg:svg xmlns:svg="http://www.w3.org/2000/svg"
      width="100px" height="75px">
      <svg:desc>Leeres SVG-Dokument</svg:desc>
    </svg:svg>
  </body>
</html>
```

- Es kann auch über den Namespace-Mechanismus in andere XML-Dokumente eingebettet werden.

SVG – Aufbau und Elemente

SVG in anderen XML-Dokumenten

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" ...>
<html xmlns="http://www.w3.org/1999/xhtml">
  <body>
    <svg xmlns="http://www.w3.org/2000/svg"
        width="100px" height="75px">
      <desc>Leeres SVG-Dokument</desc>
    </svg>
  </body>
</html>
```

- Es kann auch über den Namespace-Mechanismus in andere XML-Dokumente eingebettet werden.

SVG – Aufbau und Elemente

Primitive

```
<rect x="0"  
      y="0"  
      width="100"  
      height="75"  
      rx="5"  
      ry="5" />
```



- Eine Rechteck kann mit dem `rect`-Element definiert werden.
- Position wird über `x` und `y` angegeben, die Größe über `width` und `height`.
- Die Größe der abgerundeten Ecken bestimmen `rx` und `ry`.

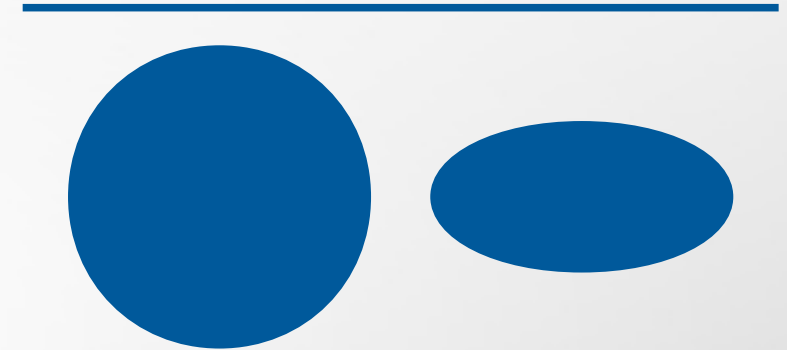
SVG – Aufbau und Elemente

Primitive

```
<circle cx="50" cy="50" r="80" />
```

```
<ellipse cx="150" cy="50"  
         rx="80" ry="40" />
```

```
<line x1="0" y1="0" x2="200" y2="0" />
```



- Bei Kreisen und Ellipsen wird der Mittelpunkt durch `cx` und `cy` festgelegt. Kreise erhalten Radius `r`.
- Ellipsen haben die beiden axialen Radien `rx` und `ry`.
- Linien erhalten die Endpunkte `(x1, y1)` und `(x2, y2)`.

SVG – Aufbau und Elemente

Styling

- Grafische Elemente erhalten Attribute zum Styling:

```
<rect x="0" y="0" width="50" height="50"  
      fill="#00599b" fill-opacity="0.5"  
      stroke="black" stroke-opacity="1"  
      stroke-width="2" stroke-dasharray="10,2"/>
```

- Folgende Styling-Attribute wurden exakt aus CSS2 übernommen:

font-*, letter-spacing, text-decoration, cursor, display,
overflow, visibility, direction, unicode-bidi, word-spacing,
clip, color

- Zur Darstellung von Text wird das `text`-Element verwendet:

```
<text x="0" y="0" font-family="Verdana"  
      font-size="20" fill="#00599b"> Hallo Welt!  
</text>
```

- Es gibt keinen automatischen Zeilenumbruch, dieser muss mit dem `tspan`-Element realisiert werden:

```
<text x="0" y="0" font-size="20" fill="#00599b" dy="30">  
  <tspan>Ach, was muss man oft von bösen</tspan>  
  <tspan>Kindern hören oder lesen!</tspan>  
</text>
```

- Zum automatisierten Zeilenumbruch führt SVG 1.2 ein:

```
<flowRoot font-size="16">
```

```
  <flowRegion>
```

```
    <rect x="0" y="0" width="200" height="100"/>
```

```
  </flowRegion>
```

```
  <flowPara> Wie zum Beispiel hier von diesen, welche  
Max und Moritz hießen. </flowPara>
```

```
</flowRoot>
```

- Es gibt weiterhin die Möglichkeit, den Text entlang eines Pfades fließen zu lassen:

```
<textPath xlink:href="#Pfad1">Gekrümmt!</textPath>
```

SVG – Aufbau und Elemente

Definitionen

- Pfade und Farbverläufe werden im `def`-Element deklariert:

```
<svg ... xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<defs>
```

```
  <path id="Pfad1" d="M 100 200 L 200 100"/>
```

```
  <linearGradient id="Verlauf1">
```

```
    <stop offset="0%" stop-color="#00599b" />
```

```
    <stop offset="100%" stop-color="black" />
```

```
  </linearGradient>
```

```
</defs>
```

```
<textPath xlink:href="#Pfad1"
```

```
  fill="url(#Verlauf1)"> Gekrümmt! </textPath>
```

SVG – Aufbau und Elemente

Pfade

- Pfade können auch direkt angezeigt werden:

```
<path d="M 10 10 L 30 10 L 20 30 z" fill="#00599b"/>
```



- M|m X Y ... Position bewegen.
- L|l X Y ... Linie ziehen.
- z|z Pfad schließen.
- C|c X1 Y1 X2 Y2 X Y ... Kubische Bézier-Kurve nach X Y.
- Große Buchstaben stehen für absolute Koordinatenangaben, kleine für relative.

SVG – Aufbau und Elemente

Externe Elemente und Gruppen

- Externe Bilder können wie folgt eingebunden werden:

```
<image x="50" y="50" width="100" height="100"
      xlink:href="bild.png">
  <title>Bild-Titel</title>
</image>
```

- Dabei können laut Spezifikation weitere SVG-Dateien sowie JPEG- und PNG-Dateien eingebunden werden.

- Mehrere Elemente werden über das `g`-Element gruppiert:

```
<g fill="#00599b" id="group1">
  <rect x="10" y="10" width="10" height="10"/>
  <rect x="30" y="10" width="10" height="10"/>
</g>
```

SVG – Aufbau und Elemente

Alternative Inhalte

- Ein interessantes Konzept ist das `switch`-Element, bei dem das erste passende Tochterelement verwendet wird:

```
<switch>
```

```
  <text systemLanguage="fr" x="0" y="0">Salut!</text>
```

```
  <text systemLanguage="en" x="0" y="0">Hello!</text>
```

```
  <text systemLanguage="de" x="0" y="0">Hallo!</text>
```

```
  <text systemLanguage="es" x="0" y="0">iHola!</text>
```

```
</switch>
```

- `switch` überprüft auch weitere Attribute und prüft, ob Bilder geladen werden können.

SVG – Aufbau und Elemente

Transformationen

- Die meisten Elemente können mit Transformationen verschoben, gedreht oder verzerrt werden:

```
<rect x="10" y="10" width="50" height="50"  
      transform="rotate(10) scale(2.5)"/>
```

- `translate(x,y)` bewirkt eine Verschiebung um x, y .
- `matrix(a,b,c,d,e,f)` bewirkt eine Transformation mit der Transformationsmatrix

$$T = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

Punkte werden transformiert nach $P' = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

SVG – Aufbau und Elemente

Filtereffekte

Im `def`-Element können komplexe Filter definiert werden, die bei der Rasterung ausgewählter Elemente verwendet werden:

```
<defs>
```

```
  <filter id="blur" filterUnits="userSpaceOnUse"  
        x="0" y="0" width="200" height="200">
```

```
    <feGaussianBlur in="Source" stdDeviation="25"  
                  result="blur"/>
```

```
  </filter>
```

```
</def>
```

```
<circle cx="100" cy="100" r="50" filter="url(#blur)"/>
```

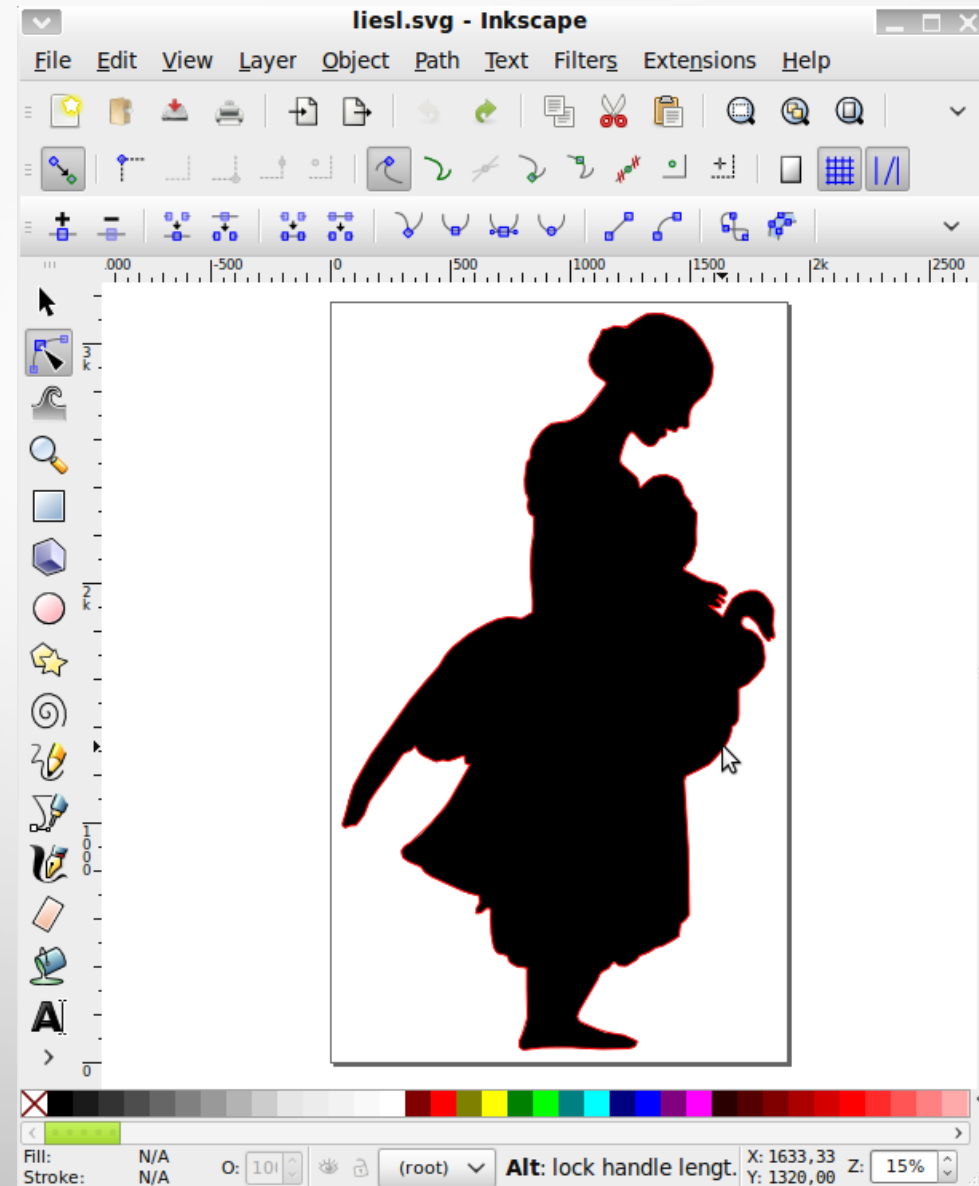
- Es gibt neben dem Gaußschen Weichzeichner noch eine ganze Menge weiterer Filter.

SVG Programmierung

SVG – Praktischer Teil

Tools

- Inkscape
- Scribus
- SVG Optimiser

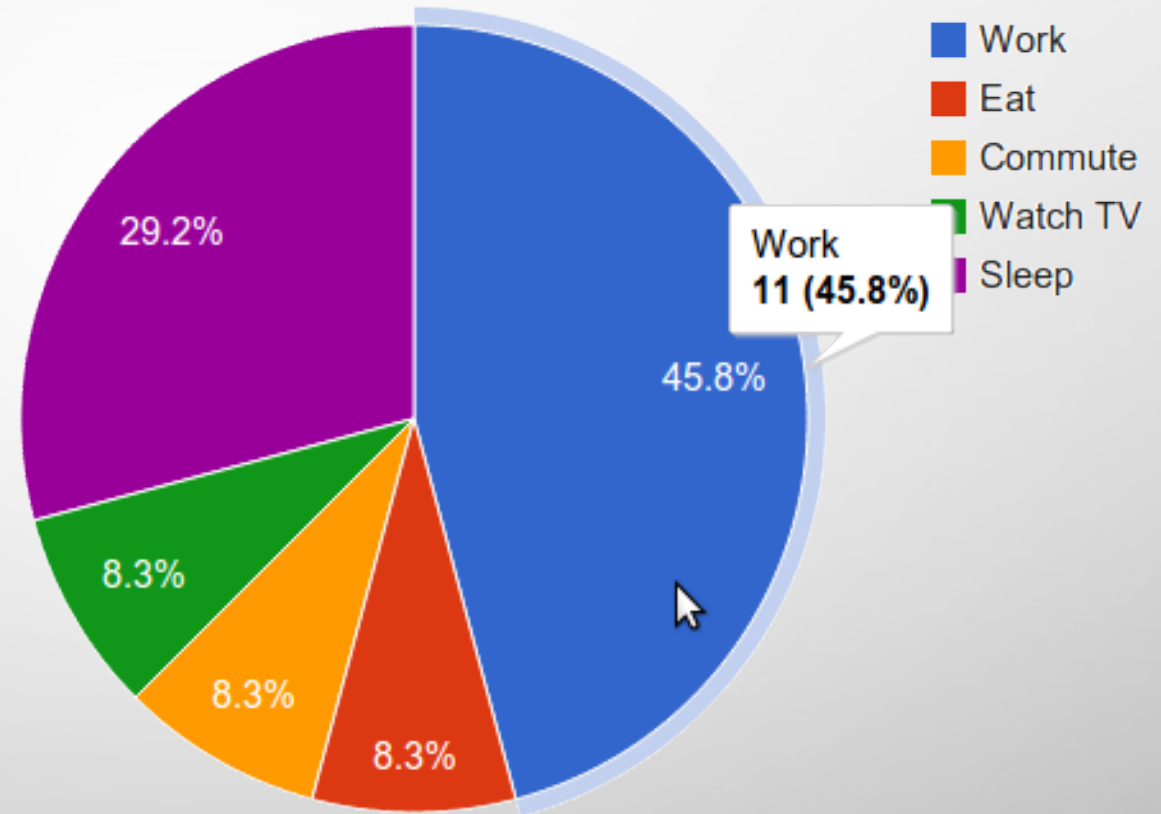


SVG – Praktischer Teil

Tools

- **Google Charts**

My Daily Activities




- **Matplotlib (Python)**

- **PySVG**

SVG – Praktischer Teil

Tools

- W3C Markup Validation Service
- W3C SVG Definition

**Markup Validation Service**
Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Notes and Potential Issues](#) [Congratulations · Icons](#)

This document was successfully checked as SVG 1.1 + XHTML5 + MathML 3.0!

Result:	Passed, 1 warning(s)	
File :	<input type="button" value="Choose File"/> No file chosen <i>Use the file selection box above if you wish to re-validate the uploaded file background_title.svg</i>	
Encoding :	utf-8	<input type="button" value="(detect automatically)"/>
Doctype :	SVG 1.1 + XHTML5 + MathML 3.0	<input type="button" value="(detect automatically)"/>
Root Element:	svg	
Root Namespace:	http://www.w3.org/2000/svg	

SVG – Praktischer Teil

Unterstützung

- Browser-Unterstützung von SVG 1.1 mittlerweile **sehr gut**.
- Unterstützung in OpenOffice und LibreOffice mangelhaft.
- GIMP rastert SVG 1.1 perfekt.
- Unterstützung von SVG 1.2 kaum vorhanden.

SMIL

Synchronized Multimedia Integration Language

SMIL – Einführung und Entstehung

Einführung

- SMIL als Auszeichnungssprache zur Erstellung von interaktiven multimedialen Präsentationen aller Art.
- Eine Vielzahl an Anwendungsmöglichkeiten, die aber oft schon durch speziellere Programme abgedeckt sind:
 - PowerPoint / Keynote
 - Video-Formate
 - Flash
 - HTML
- Idee eines umfassenderen, offenen, XML-basierten Formats.
 - Trennung von Inhalten und den anordnenden Metadaten.

SMIL – Einführung und Entstehung

Entstehung

- 1998: **SMIL 1.0** wird Empfehlung des W3C.
- 2001: **SMIL 2.0** wird Empfehlung des W3C.
 - Drastische Steigerung des Umfangs: Die Beschreibung des W3C steigt von 30 auf etwa 600 Seiten.
 - Modularisierung und Implementation des Namespace-Mechanismus wie bei SVG 1.1.
- 2005: Erster großer Einsatz von SMIL bei der HD DVD.
 - Das Format unterliegt jedoch bald der BluRay.
- 2008: **SMIL 3.0** wird Empfehlung des W3C.
 - Bis heute nur wenige Implementationen.
- 2012: W3C SYMM Working Group schließt.

SMIL – Einführung und Entstehung

Anwendungen

- Theorie:
 - Komposition von bestehenden multimedialen Inhalten.
 - Interaktiv oder nur zur zeit-synchronen Wiedergabe.
- Praxis:
 - Hat kaum Verwendung gefunden, heute kaum noch Beispiele zu finden.
 - Hinzufügen von Untertiteln zu Videos.
 - Anwendung beim eLearning.
 - Möglichkeiten zum Video-Schnitt.

SMIL – Aufbau und Elemente

Strukturelemente

```
<?xml version="1.0"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta name="title" content="Präsentationstitel"/> ...
  </head>
  <body> ... </body>
</smil>
```

- Ein SMIL-Dokument besteht aus einer XML- und DocType-Deklaration und einem `smil`-Element.
- Das `smil`-Element besteht aus `head` und `body`.

SMIL – Aufbau und Elemente

Layoutsdefinition

- Im `head`-Bereich wird die Anordnung der Medien definiert.
- Es werden `region`-Elemente angelegt, die später referenziert werden. Ein `root-layout`-Element definiert die Gesamtgröße.

```
<head>
```

```
  <layout>
```

```
    <root-layout width="320" height="480" />
```

```
    <region id="titel" left="5" top="5"  
          width="310" height="50" />
```

```
    <region id="inhalt" left="10" top="70"  
          backgroundColor="yellow" />
```

```
  </layout>
```

```
</head>
```

SMIL – Aufbau und Elemente

Medienelemente

```
  
<audio src="ton.wav" begin="2s" title="Willkommen!" />  
<animation src="ani.svg" region="r2" begin="0s" />  
<video src="clip.mpg" region="r3" begin="0s"  
      clipBegin="4s" clipEnd="60s" />  
<textstream src="untertitel.rt" region="r4" />
```

- Medienelemente werden mit Angabe einer Region (visuelle Elemente) und mit zeitlichen Angaben definiert.
- Bei Elementen mit intrinsischer Dauer muss keine Dauer festgelegt werden, kann aber.

SMIL – Aufbau und Elemente

Exkurs RealText

- Für `textstreams` wird das RealText-Format empfohlen:

```
<window height="80" width="300" duration="12">  
  Ach, was muss man oft von bösen<br/>  
  <time begin="3"/>Kindern hören oder lesen!<br/>  
  <time begin="6"/>Wie zum Beispiel hier von diesen,<br/>  
  <time begin="9"/>Welche Max und Moritz hießen.  
</window>
```

```
<window type="marquee" duration="12">  
  Die, anstatt durch weise Lehren sich zum Guten zu  
  bekehren, oftmals noch darüber lachen und sich heimlich  
  lustig machen.  
</window>
```

SMIL – Aufbau und Elemente

Inhaltsauswahl

```
<switch>
```

```
  <audio src="a_nederlands.wav" systemLanguage="nl"/>
```

```
  <audio src="a_deutsch.wav" systemLanguage="de"/>
```

```
  <audio src="a_english.wav"/>
```

```
</switch>
```

- Wie bei SVG existiert das `switch`-Element, Auswahl-Attribute sind aber hier unter anderem:
 - `systemBitrate` (in Bits pro Sekunde)
 - `systemCaptions` (on|off)
 - `systemScreenSize` (z.B. 1024X768)
 - `SystemOperatingSystem`

SMIL – Aufbau und Elemente

Synchronisation

- Synchronisation durch Timing-Elemente `par` und `seq` für parallele und sequentielle Ausführung:

```
<seq>
```

```
  <par end="next.activateEvent">
```

```
    <video region="v" src="walking_on_sunshine.mpg"/>
```

```
    
```

```
    
```

```
  </par>
```

```
  <audio src="applaus.wav" id="cheering"/>
```

```
  
```

```
</seq>
```

SMIL – Aufbau und Elemente

Animationen

- Es werden unter anderem die von SVG wieder aufgegriffenen Animationsmöglichkeiten geboten:

```
<animate attributeName="top" from="0" to="10" dur="5s"
    repeatCount="1" />
```

```
<animateMotion path="m 50 50 c 80 90 70 50 100 0 z"
    dur="5s" repeatCount="2" />
```

```
<set attributeName="class" to="highlight"
    begin="mouseover" end="mouseout" />
```

```
<animateColor attributeName="backgroundColor" from="red"
    to="blue" dur="5s" repeatCount="1" />
```


SMIL – Beispiel und Unterstützung Player

- Einziger Player, der die Versionen nach SMIL 2.0 versteht:
 - [Ambulant 2.4](#)
- Folgende Player spielen bis zu Version SMIL 2.0:
 - Quicktime
 - RealPlayer
 - Windows Media Player
 - ...
- Infolge der Integration in den Standard HD DVD gibt es sogar Hardware Player, die SMIL unterstützen.

SMIL – Beispiel und Unterstützung

Authoring Software

- Freie Software:
 - Kino 1.3.4 (Linux, sehr geringer Funktionsumfang)
 - LIMSee 2 (Java, Installation nicht mehr zu finden)
 - LIMSee 3 (Java, komplizierte Oberfläche)
- Kommerzielle Software:
 - Adobe GoLive!
 - Dreamweaver
 - ...

SMIL – Beispiel und Unterstützung

HelloWorld-Beispiel

```
<smil>
  <head>
    <layout>
      <root-layout width="248" height="300" background-color="blue" />
      <region id="a" top="20" left="64" />
      <region id="b" top="120" left="20"/>
    </layout>
  </head>
  <body>
    <par>
      
      
      <audio src="http://www.content-networking.com/smil/hello.wav"
        begin="4s"/>
    </par>
  </body>
</smil>
```

Exkurs

SVG und XSLT

- XSL Transformation ist eine Turing-vollständige XML-Sprache zur Transformation von XML-Dokumenten.
- Ebenfalls eine Empfehlung des W3C, seit 2007 XSLT 2.0.
- Wird von den meisten gängigen Browsern direkt unterstützt.

Danke für eure Aufmerksamkeit.

Gibt es noch Fragen?

Georg Jahn
mail@gjahn.com

Anhang

Bildquellen

[1] <http://www.w3.org/Graphics/SVG/>

[2] <http://www.w3.org/AudioVideo/>

[3] <http://www.w3.org/Consortium/Legal/logo-usage-20000308>