

SVG auf dem Weg zum reichhaltigen Medienformat

– Neuerungen in Version 1.2

Dr. Thomas Meinike

thomas.meinike@et.fh-merseburg.de

<http://www.et.fh-merseburg.de/person/meinike/>

Fachhochschule Merseburg

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

[Vortrag auf der tekcom-Jahrestagung 2004 in Wiesbaden – gehalten am 11./12.11.2004]



Überblick

- ⇒ Aufbau und Ziel des Vortrags
- ⇒ SVG in 5 Minuten
- ⇒ Überblick zu SVG 1.2
- ⇒ SVG 1.2 im Detail
- ⇒ Zusammenfassung und Ausblick



Aufbau und Ziel des Vortrags

⇒ **Gegenstand:**

Vektorgrafikformat Scalable Vector Graphics (SVG)

⇒ **Ziel:**

Vermittlung von Informationen zum aktuellen Stand der in Entwicklung befindlichen SVG-Version 1.2

⇒ **Ausgangspunkt:**

bisherige SVG-Spezifikationen 1.x

⇒ **Vorgehensweise:**

Aufzeigen der neuen Möglichkeiten und praktische Demonstrationen



SVG in 5 Minuten – Was ist SVG?

- ⇒ SVG 1.0 ist eine 2001 verabschiedete W3C-Spezifikation zur Beschreibung von 2D-Vektorgrafiken in XML-Syntax [1.1 (2003) / 1.2 (2005)].
- ⇒ Enthalten sind geometrische Primitive wie Rechteck, Kreis, Ellipse, Linie, Polylinie, Polygon sowie weitere Elemente zur Darstellung von Pfaden, Text, Hyperlinks usw.
- ⇒ SVG-Inhalte lassen sich mit CSS formatieren und durch den Einsatz von JavaScript-Routinen dynamisch verändern.
- ⇒ Techniken wie Animationen, Gradienten, Filter und Transformationen ermöglichen auch komplexe Darstellungen.

Grundgerüst:

```
<?xml version="1.0" encoding="..."?>
<!DOCTYPE svg ...>
<svg xmlns="http://www.w3.org/2000/svg">
  <!-- Inhalte der Grafik -->
</svg>
```

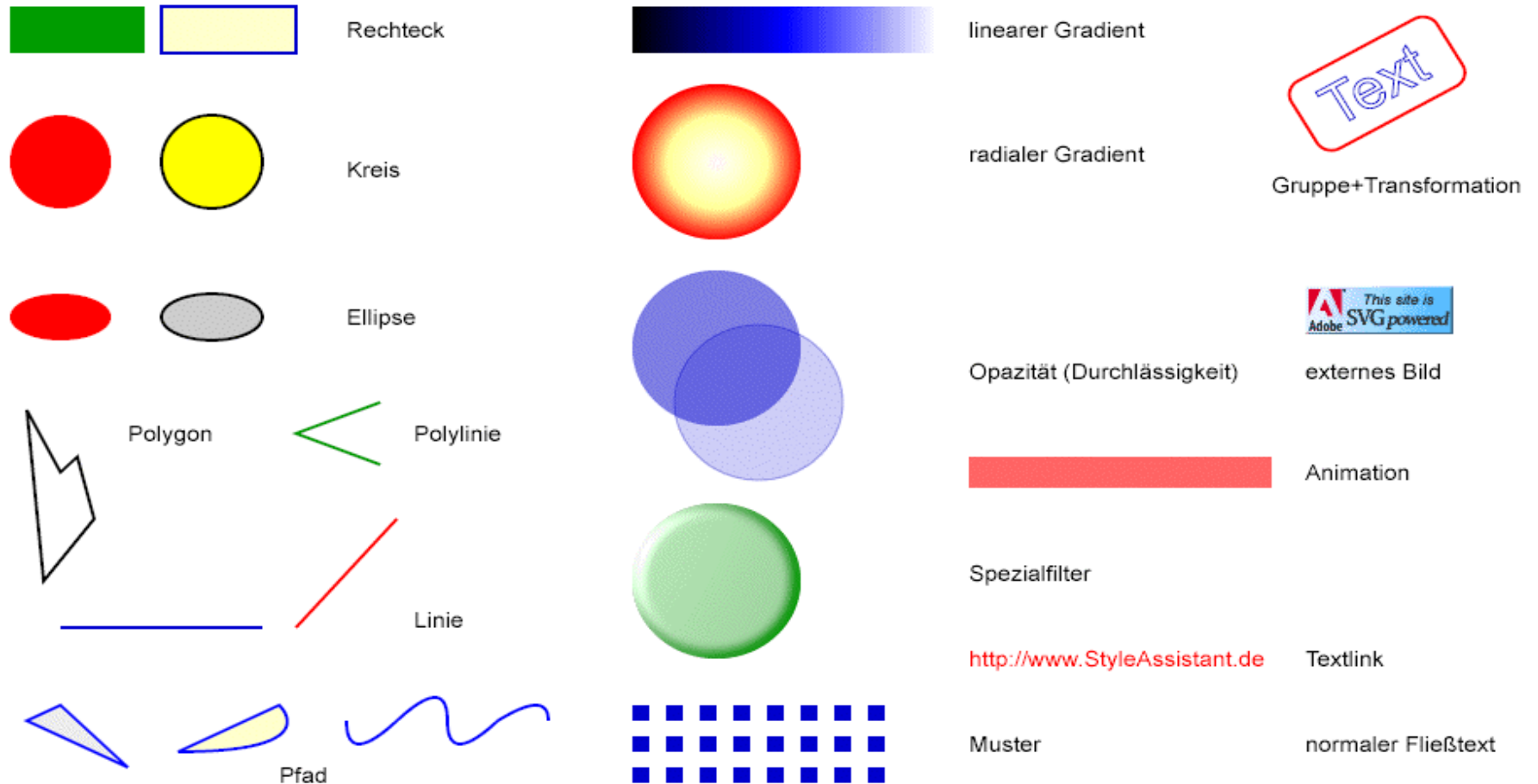
Einige SVG-Elemente mit Attributen:

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

SVG in 5 Minuten – Objekte und Effekte

Objekte und Effekte in SVG

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



© by Dr. Thomas Meinike 2002

TMs10kSVGDemo.svg

Überblick zu SVG 1.2

⇒ W3C-SVG-Roadmap:

This page contains descriptions of drafts that the [SVG WG](#) (Scalable Vector Graphics Working Group) is working on or has completed. If you want to follow the development of SVG, this page should be the place to start. Note that the content of this page changes often, so you should not draw any conclusions from the data here.

The following table lists the publication dates (past or *expected*) for all of the SVG documents. The predicted dates are intended to be indicative only.

Document	FWD	Next WD	LC	CR	PR	REC
SVG 1.0	11 Feb 1999	-	03 Mar 2000	02 Aug 2000	19 July 2001	5 Sep 2001
SVG 1.1	30 Oct 2001	-	15 Feb 2002	30 Apr 2002	15 Nov 2002	14 Jan 2003
SVG Mobile Profiles	30 Oct 2001	-	15 Feb 2002	30 Apr 2002	15 Nov 2002	14 Jan 2003
SVG Mobile1.2	9 Dec 2003	-	13 Aug 2004	<i>[Nov 2004]</i>	<i>[Mar 2005]</i>	<i>[May 2005]</i>
SVG 1.2	11 Nov 2002	-	27 Oct 2004	<i>[Dec 2004]</i>	<i>[Mar 2005]</i>	<i>[May 2005]</i>
DOM Level 3 Events	01 Sep 2000	-	31 Mar 2003	<i>[Dec 2004]</i>	<i>[Mar 2005]</i>	<i>[May 2005]</i>
DOM Level 3 XPath	18 Jun 2001	-	28 Mar 2002	31 Mar 2003	<i>[Mar 2005]</i>	<i>[May 2005]</i>
sXBL	01 Sep 2004	<i>[Nov 2004]</i>	<i>[Feb 2005]</i>	<i>[Apr 2005]</i>	<i>[Sep 2005]</i>	<i>[Nov 2005]</i>
SVG Print	15 July 2003	<i>[Dec 2004]</i>	<i>[Mar 2005]</i>	<i>[Jun 2005]</i>	<i>[Sep 2005]</i>	<i>[Nov 2005]</i>
Authoring Tool Guidelines	<i>[Feb 2005]</i>	-	-	-	-	-
Accessibility Techniques	<i>[May 2005]</i>	-	-	-	-	-

Legend: **FWD** = First working draft; **LC** = last call for comments (i.e., last WD); **CR** = Candidate Recommendation; **PR** = Proposed Recommendation; **REC** = W3C Recommendation. *[Feb 2005]* = expected date.

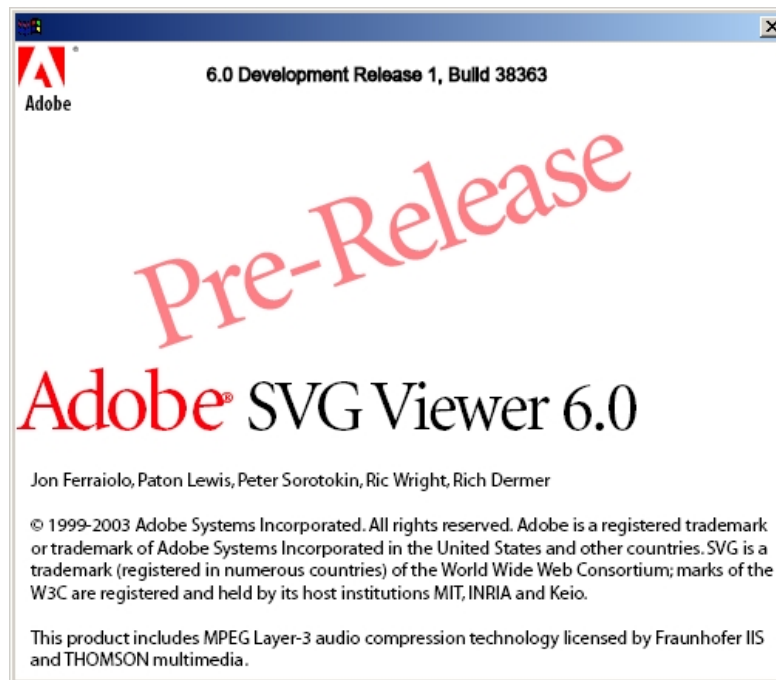
Quelle: W3C

Überblick zu SVG 1.2

⇒ Entwicklungsstand: bisher 8 Working Drafts

- 7. Working Draft“ vom 10.05.2004 (Grundlage für Tagungsband)
- „Last Call Working Draft“ vom 27.10.2004 (Ende 24.11.2004)

⇒ Implementierung: ansatzweise im Adobe SVG Viewer 6.0



Überblick zu SVG 1.2

- ⇒ Wichtige Neuerungen kompakt:
- Verwendung von Fließtext / Eingabetechniken
 - Neue Darstellungs- und Zeichentechniken
 - Multimedia-Erweiterungen
 - Entwurf und Einsatz eigener Komponenten (RCC/sXBL)
 - Zusatzfunktionen und spezielle Ergänzungen
 - DOM-Anpassungen



SVG 1.2 im Detail

⇒ Verwendung von Fließtext:

- bisher nicht oder nur eingeschränkt mit `text` / `tspan`-Kombination realisierbar gewesen
- neue SVG-Elemente zum Arbeiten mit Fließtext
- Text (und Grafiken) können innerhalb von beliebigen Grundformen
- mit automatischen Zeilenumbrüchen eingebunden werden

- Textfluss-Elemente

`flowDiv`

`flowImage`

`flowLine`

`flowPara`

`flowRef`

`flowRegion`

`flowRegionBreak`

`flowRegionExclude`

`flowRoot`

`flowSpan`

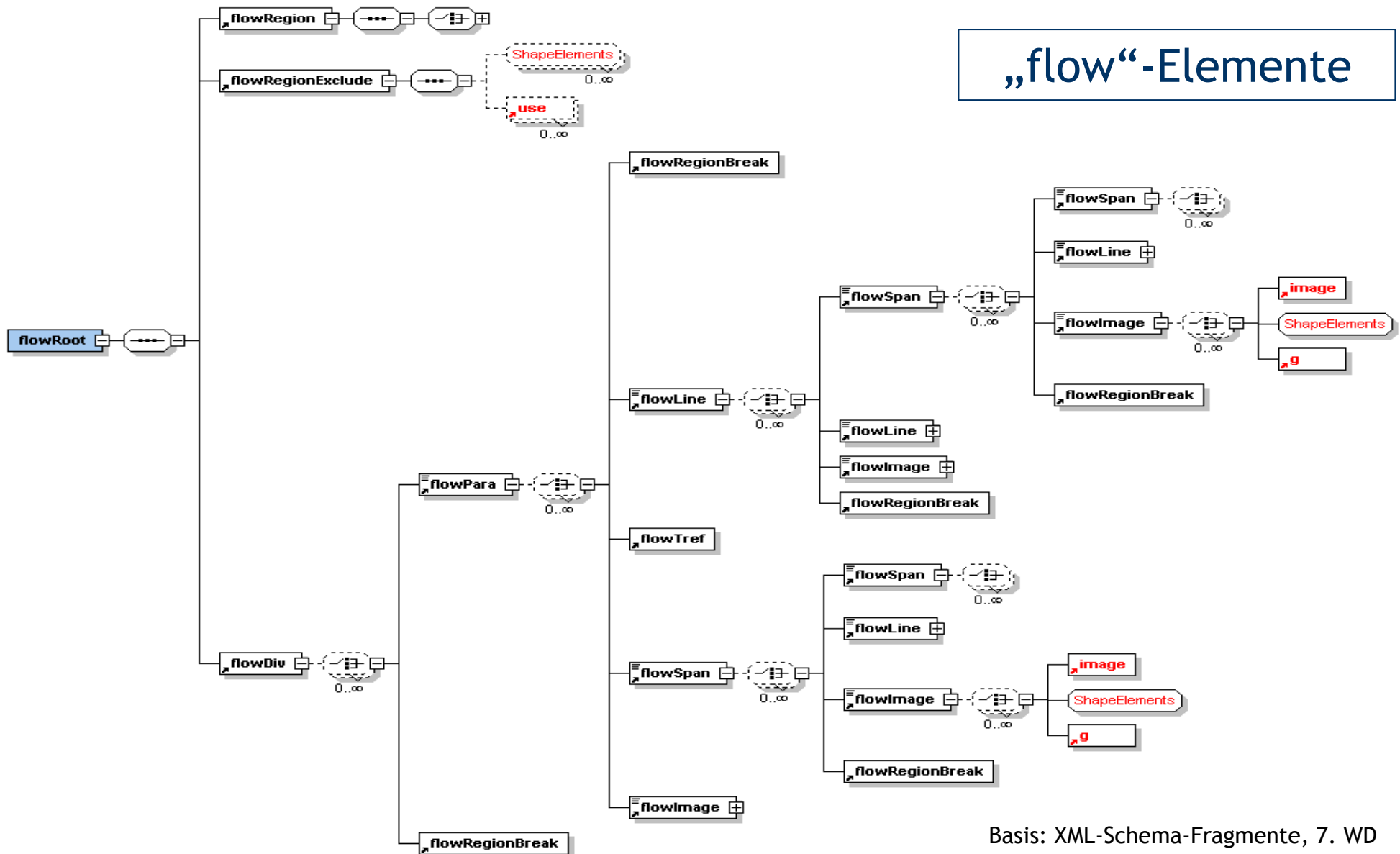
`flowTref`

- Textausrichtung mit `text-align` / `progression-align`



SVG 1.2 im Detail

„flow“-Elemente



Basis: XML-Schema-Fragmente, 7. WD

SVG 1.2 im Detail

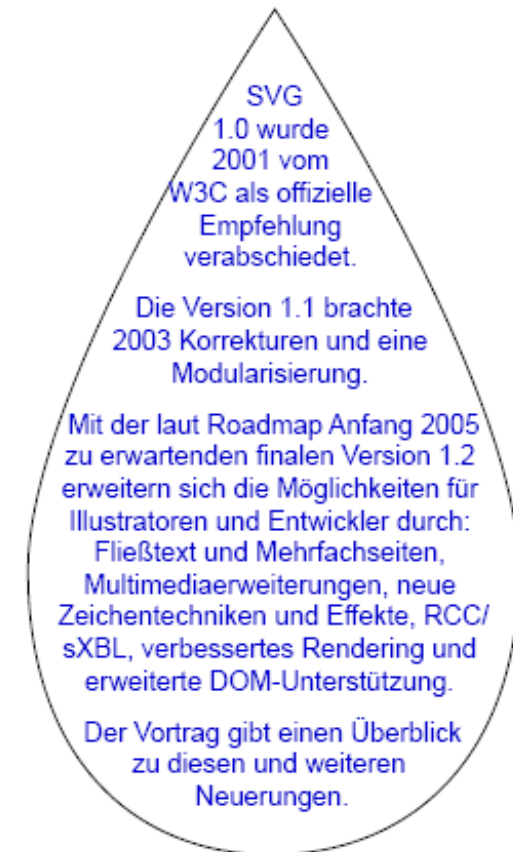
⇒ Fließtext-Beispiel:

```
<path id="tropfen" d="..." />
...
<flowRoot>

  <flowRegion>
    <use xlink:href="#tropfen" />
  </flowRegion>

  <flowDiv>
    <flowPara>Absatzinhalt</flowPara>
    <flowPara>Absatzinhalt</flowPara>
    <flowPara>Absatzinhalt</flowPara>
  </flowDiv>

</flowRoot>
```



Hinweis zum ASV 6: flowRoot = flow, use = region

SVG 1.2 im Detail

⇒ Eingabetechniken:

- bisher keine direkte Möglichkeit für Nutzereingaben vorgesehen
- die Elemente `text` und `flowDiv` erhalten das Attribut `editable` (Werte: `true` / `false`)
- editierbare Textfelder können mit den CSS-Pseudoklassen `:focus` und `:edited` behandelt werden
- Textauswahl soll sich über das Attribut `selection` (`svg`-Element) und das Ereignis `SVGSelectionChanged` überwachen lassen
- Eigenschaft `focusable` (Werte: `true` / `false` / `auto`) ermöglicht Navigation zwischen Objekten



SVG 1.2 im Detail

⇒ Neue Darstellungs- und Zeichentechniken:

- Mehrfachseiten (z. B. als Foliensatz einer Präsentation verwendbar)

```
<svg ...>
```

```
<!-- hier abgelegte SVG-Inhalte sind für alle mit page
      definierten Seiten sichtbar („Folien-Master“) -->
```

```
<pageSet>
```

```
  <page><!-- Inhalte der 1. Seite --></page>
```

```
  <page><!-- Inhalte der 2. Seite --></page>
```

```
  <page><!-- weitere Unterseiten --></page>
```

```
</pageSet>
```

```
</svg>
```

- Seitenausrichtung (Hoch- / Querformat)

```
<page page-orientation="-270|-180|-90|0|90|180|270">
```

- page-Navigation über SMIL-Zeitattribute **begin** / **end** / **dur**



SVG 1.2 im Detail

⇒ Neue Darstellungs- und Zeichentechniken:

- Vektoreffekte für komplexere Linien/Rahmen, Füllungen und Kombinationen von Formen

- Element `vectorEffect` mit wählbaren Kindelementen und Attributen

<code>veAffine</code>	<code>vePath</code>
<code>veExclude</code>	<code>vePathRef</code>
<code>veFill</code>	<code>veReverse</code>
<code>veIntersect</code>	<code>veSetback</code>
<code>veJoin</code>	<code>veStroke</code>
<code>veMarker</code>	<code>veStrokePath</code>
<code>veMarkerPath</code>	<code>veUnion</code>

- Zuweisung erfolgt analog zu Gradienten oder Filtern

```
<path vector-effect="url(#resID)" d="..." />
```



SVG 1.2 im Detail

⇒ Neue Darstellungs- und Zeichentechniken:

➤ Beispiel „Schiene“



- statisch mit 2 Linien

```
<line x1="20" y1="20" x2="520" y2="20"
      fill="none" stroke="#000" stroke-width="8"/>
<line x1="20" y1="20" x2="520" y2="20" fill="none"
      stroke="#FFF" stroke-width="4" stroke-dasharray="10,2"/>
```

- als beliebig verwendbarer Vektoreffekt

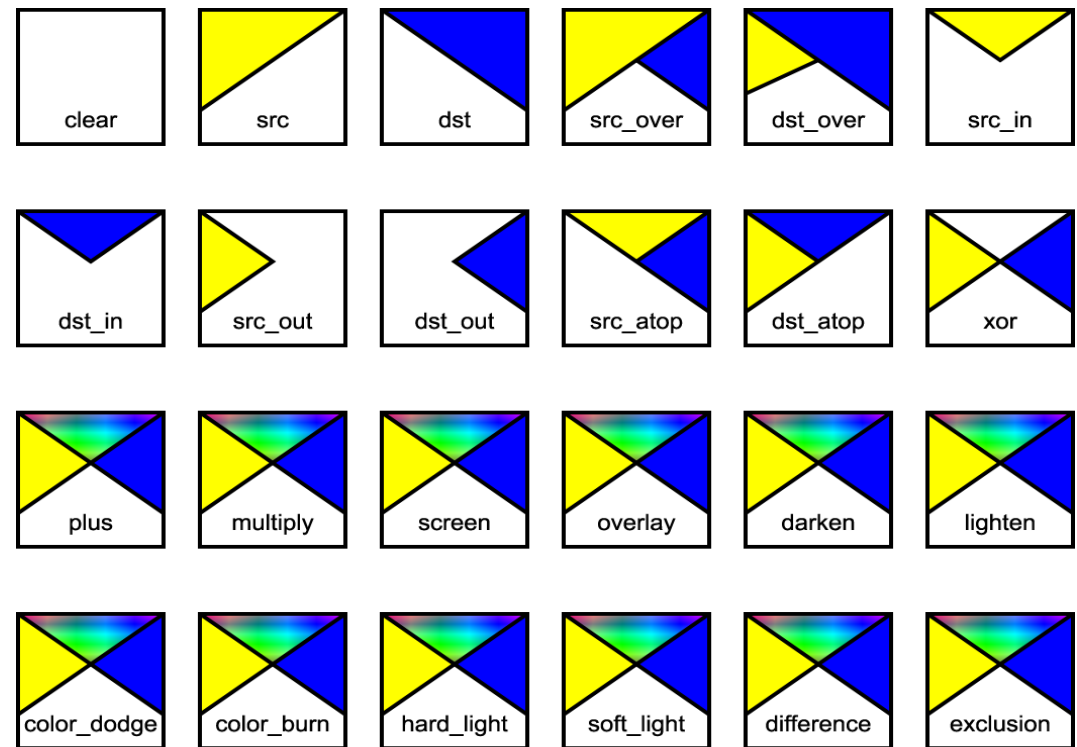
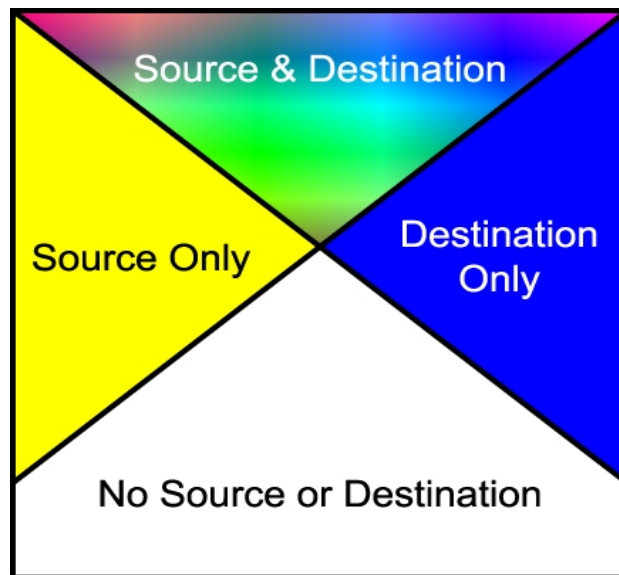
```
<vectorEffect id="schiene">
  <veStroke fill="none" stroke="#000" stroke-width="8"
            in="SourcePath"/>
  <veStroke fill="none" stroke="#FFF" stroke-width="4"
            stroke-dasharray="10,2" in="SourcePath"/>
</vectorEffect>
...
<line x1="20" y1="20" x2="520" y2="20"
      vector-effect="url(#schiene)"/>
```



SVG 1.2 im Detail

⇒ Neue Darstellungs- und Zeichentechniken:

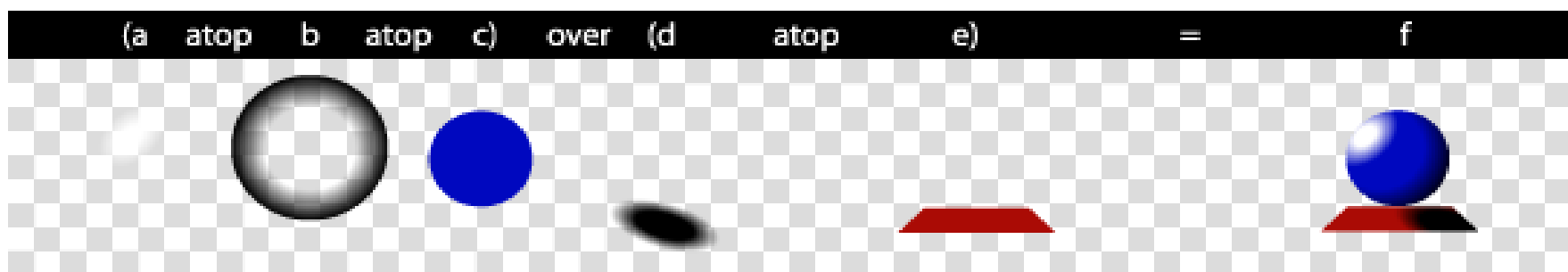
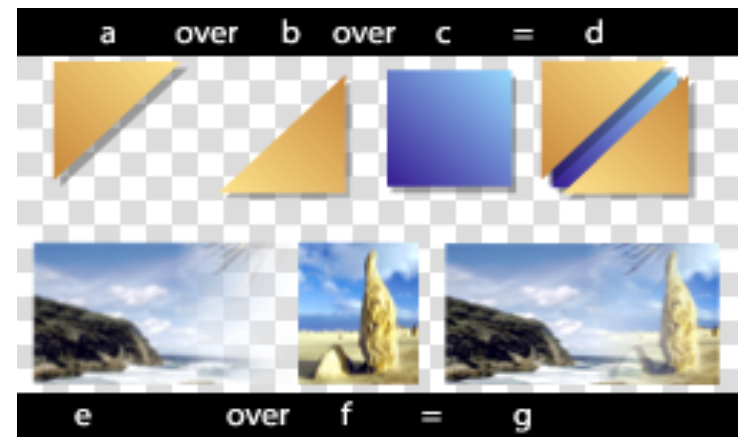
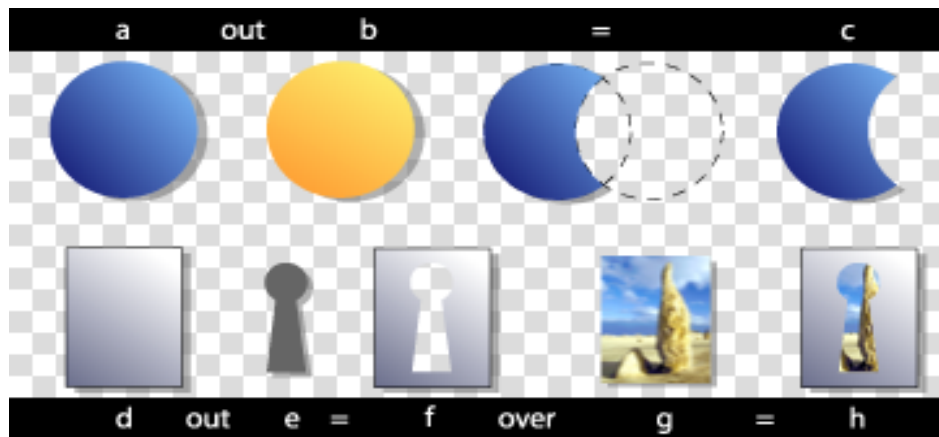
- „Alpha-Compositing“ von Grafikelementen / Hintergrund
 - 24 Porter-Duff- und Überblend-Operationen kombinierbar
 - Operationen werden auf Pixel-Regionen angewendet



SVG 1.2 im Detail

⇒ Neue Darstellungs- und Zeichentechniken:

➤ „Alpha-Compositing“ von Grafikelementen / Hintergrund



Quelle: SVG 1.2 WD

SVG 1.2 im Detail

⇒ Neue Darstellungs- und Zeichentechniken:

- Eigenschaften **background-fill** und **background-fill-opacity**

```
<svg ... background-fill="#FF0" background-fill-opacity="0.8"/>
```

- Element **solidColor** / Eigenschaften **solid-color** und **solid-opacity**

```
<solidColor solid-color="#F00" solid-opacity="0.6" id="col"/>
```

```
<element ... fill="url(#col)"/>
```

- Element **deviceColor** zur Nutzung von Geräte-spezifischen Farben (z. B. für Druckausgaben)

```
<deviceColor name="drucker" xlink:href="http://.../farbprofil" value="farbe1, farbe2, ..." />
```

```
<element ... fill="rgb(r,g,b) device-color(wert1,wert2,...)"/>
```



SVG 1.2 im Detail

⇒ Multimedia-Erweiterungen:

➤ Element audio

```
<audio xlink:href="datei.mp3" type="audio/mp3"  
  volume="..." begin="..." repeatCount="..." />
```

➤ Element video

```
<video xlink:href="datei.mpg" type="video/mpeg"  
  volume="..." begin="..." repeatCount="..."  
  x="..." y="..." width="..." height="..." />
```

- Formate: noch offen, möglichst lizenzfrei
- ASV 3/6: a:audio
- ASV 6: a:video



SVG 1.2 im Detail

⇒ Multimedia-Erweiterungen:

- Element `animation` (wird als separates Grafikobjekt betrachtet)

```
<animation x="..." y="..." width="..." height="..."  
  begin="..." xlink:href="#resID"/>
```

- Element `transition` (Übergangseffekte / SMIL)

```
<transition id="..." type="..." subtype="..." dur="..."  
  direction="..." fadeColor="..." startProgress="..."  
  endProgress="..." />
```

- Anwendung bei Animationen von verlinkten Inhalten
(Elemente: `image`, `video`, `use`, `page` und `felmage`)



```
<image ... transIn="inID" transOut="outID">  
  <animate ... attributeName="xlink:href" begin="..."  
    dur="..." values="a.jpg;b.jpg;c.jpg" fill="freeze"/>  
</image>
```

SVG 1.2 im Detail

⇒ Multimedia-Erweiterungen:

➤ Streaming von Inhalten

- Attribut **timelineBegin**="onLoad | onStart"
- onLoad: Zeitskala startet bei vollständig geladenem Element (**page**)
- onStart: Zeitskala startet nach erfolgreichem Parsen des Start-Tags (**svg**)
- svg-Attribut **streamedContents**="keep | discard"
steuert das Erhalten / Verwerfen von **page**-Inhalten

➤ Vorladen von Inhalten (**prefetch**-Element)

```
<defs>  
  <prefetch xlink:href="..." />  
  <prefetch xlink:href="..." />  
</defs>
```

- Attribute zur Steuerung: **mediaSize**, **mediaTime**, **bandwidth**



SVG 1.2 im Detail

- ⇒ Entwurf und Einsatz eigener Komponenten (RCC/sXBL):
 - Einbindung beliebiger XML-Strukturen in SVG-Dokumente und direkte Umsetzung der Inhalte in den SVG-Kontext
 - Verwendung eigener Namensräume möglich
 - Kopplung der Inhalte über XML-Ereignisse an Skript-Routinen (eigene Inhalte werden in einem **shadowTree**-DOM verwaltet)
 - bis zum 7. WD als Rendering Custom Content (RCC) bezeichnet
 - neue Basis: SVG's XML Binding Language (sXBL) (separate Spezifikation ab 09/2004, im WD-Stadium)
 - Anwendungen: Formularkomponenten, Flussdiagramme, ...



SVG 1.2 im Detail

⇒ Entwurf und Einsatz eigener Komponenten (RCC/sXBL):

➤ Beispiel: Buttons aus zwei versetzten Rechtecken und Text

```
<svglbc:simplebutton x="170" y="90" bb="180" bh="90"  
  bf="#CFC" tf="#00C" th="36" at="Hallo Button 2!">
```

Text 2

```
</svglbc:simplebutton>
```

XML-Code wird als SVG
gerendert!



Text 2

```
<rect x="." y="." .../>  
<rect x="." y="." .../>  
<text ...>Text 2</text>
```

SVG 1.2 im Detail

⇒ Entwurf und Einsatz eigener Komponenten (RCC/sXBL):

➤ Prinzip der sXBL-Verarbeitung

```
<defs>
  <xbl:xbl>
    <xbl:definition name="svglbc:simplebutton">
      <xbl:template>
        <g><rect id="unten"/><rect id="oben"/>
        <text id="txt"><xbl:content/></text></g>
      </xbl:template>
      <xbl:handlerGroup>
        <handler ev:event="xbl:bound" type="text/javascript">
          // simplebutton-Element bezogen auf xbl:template verarbeiten
          el=evt.target;st=el.xblShadowTree;
          // XML-Daten über DOM-Methoden abfragen/zuweisen ...
        </handler>
      </xbl:handlerGroup>
    </xbl:definition>
  </xbl:xbl>
</defs>
```



SVG 1.2 im Detail

⇒ Zusatzfunktionen und spezielle Ergänzungen:

- „Tooltips“ mittels `hint`-Element einfach definierbar

```
<element ... >  
  <hint>anzuzeigender Tooltip-Text</hint>  
</element>
```

- „Highlighting“ von Inhalten über die neue Pseudoklasse `:highlight`

```
:highlight {stroke-width: 3; stroke-color: #F00;}  
:highlight {vector-effect: url(#schiene);}
```

- Allgemeine Verbesserungen in den Bereichen Painting / Rendering
 - bessere Nutzbarkeit von Farbprofilen
 - Überlagerung von Grafikinhalten (`overlay`-Eigenschaften)
 - Beschränkte Transformationen mit `transform="ref(...)"`
 - Darstellung alternativer Inhalte in Abhängigkeit von der Auflösung



SVG 1.2 im Detail

⇒ Zusatzfunktionen und spezielle Ergänzungen:

➤ Nicht-grafische Erweiterungen

- Attribut **xlink:href** auch für die Elemente **title**, **desc** und **metadata** definiert, um externe beschreibende Informationen einzubinden
- erweiterte Metadaten zur Angabe von Informationen zu Copyrights und Veröffentlichung (Verteilung, Vervielfältigung, Lizenzen)
„Creative Commons Metadata Set“: **cc:license**, **cc:permits**, **cc:prohibits**, **cc:requires**, ...

```
<metadata>  
  ...  
  <cc:license rdf:resource="http://www.example.org/copyright"/>  
  ...  
</metadata>
```



SVG 1.2 im Detail

⇒ DOM-Anpassungen:

- Unterstützung von **DOM Level 3** Core, Events, XPath sowie der bis SVG 1.1 spezifizierten Eigenschaften und Methoden
- Einbeziehung der Multimedia-Neuerungen in das SVG-DOM, um dynamischen Zugriff auf Inhalte zu ermöglichen
- neue Programmier-Schnittstellen (APIs)
 - Interface **SVGTimer** (Zeitsteuerung)
 - Interfaces **URLRequest** / **Connection** (Zugriff auf externe Ressourcen)
 - Interface **FileDialog** (Dateizugriff)
 - Interface **Global** (allgemeine Techniken, formal am window-Objekt von Skriptsprachen wie JavaScript orientiert)



SVG 1.2 im Detail

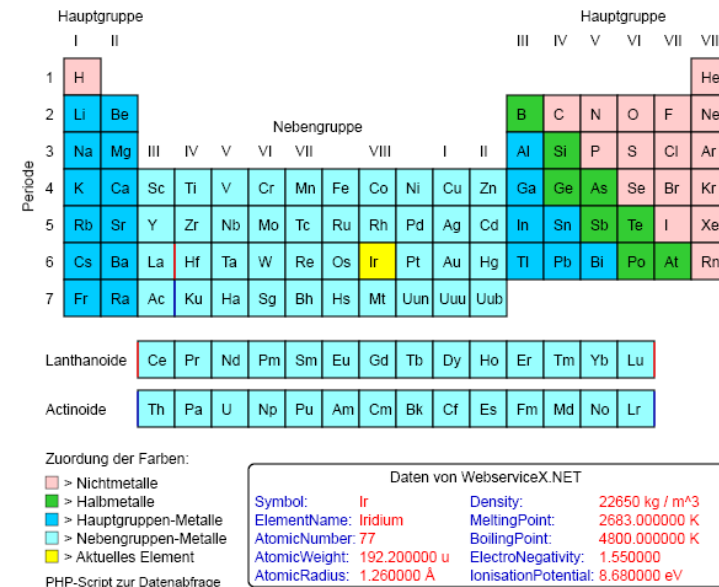
⇒ DOM-Anpassungen:

➤ SVG-Datenkommunikation

- die bisherigen ASV-Methoden `getURL()` und `postURL()` lassen sich durch die neuen `URLRequest`-Methoden nachbilden
- während `parseXML()` und `printNode()` dem Interface `Global` zugeordnet wurden

➤ Beispiel

Zugriff auf den Web Service
 „Periodic Table“ (WebserviceX.NET)



Zusammenfassung und Ausblick

- ⇒ SVG entwickelt sich mit Version 1.2 deutlich weiter in Richtung „Medienformat“.
- ⇒ Die Verabschiedung der Spezifikation ist 2005 zu erwarten.
- ⇒ Eine schnelle Entwicklung bzw. Anpassung von Werkzeugen für Entwurf und Darstellung von SVG-Inhalten ist notwendig.
- ⇒ SVG bietet als offenes XML-Format Vorteile bei der Erstellung und Verarbeitung im Rahmen von XML-basierten Publikationsprozessen.
- ⇒ Im Bereich Web dominieren alternative Technologien (Flash). Ohne entscheidende Verbesserungen der Zugangsmöglichkeiten und Softwareunterstützung wird SVG ein Nischenformat bleiben.
- ⇒ Die Modularisierung von SVG 1.2 soll auch die Entwicklung von mobilen Anwendungen besser unterstützen.



Online-Ressourcen

- ⇒ W3C: SVG - XML Graphics for the Web
<http://www.w3.org/Graphics/SVG/>
- ⇒ W3C: SVG-Roadmap
<http://www.w3.org/Graphics/SVG/Roadmap.html>
- ⇒ W3C: SVG 1.2-Working Draft
<http://www.w3.org/TR/SVG12/>
- ⇒ W3C: sXBL-Working Draft
<http://www.w3.org/TR/sXBL/>
- ⇒ Adobe: SVG Viewer 6.0 (Pre-Release)
<http://www.adobe.com/svg/viewer/install/beta.html>
- ⇒ Meinike, T.: SVG - Learning By Coding
<http://svglbc.datenverdrahten.de/>

