

SVG - Flash-Killer oder Totgeburt?

Christian Wenz & Tobias Hauser

Copyright © 2003 Christian Wenz & Tobias Hauser

Inhaltsverzeichnis

Einführung	1
SVG-Elemente	2
SVG-Scripting	2
Serverseitiges SVG	4
Plugin-Situation	5
Editoren-Situation	5
Fazit	6

Einführung

Glaubt man einigen Ideologen, macht SVG, das Vektor-Format für das Web, zurzeit Furore; hin und wieder ist sogar davon zu lesen, es handle sich hierbei um "The Next Big Thing". Glaubt man jedoch unabhängigen Statistiken und Berichten, so stockt die Verbreitung des Formats. Liegt es am übermächtigen, proprietären Konkurrenten SWF? Wo liegen die Unterschiede? Fragen, die in diesem Beitrag zumindest annähernd geklärt werden sollen.

Der Lebenszyklus vieler neuerer Technologien verläuft häufig mehrstufig. Zunächst wird die Technologie über den Klee gelobt, teils ideologisch getrieben ("es ist Open Source, also per definitionem gut" oder auch "es ist teuer, das bürgt für Qualität"), teils aufgrund einer Nachrichtenflaute verschiedener Publikationen, teils sogar gerechtfertigt. Nach einiger Zeit ist jedoch kommt es zu einer kritischen Phase. Der Hype legt sich, die breite Masse fordert nach echten Vorteilen einer Technologie und realitätsnahe Anwendungen, die weit über einen anfänglichen "Proof-of-Concept" hinausgehen. Gelingt dies, hat die Technologie eine Chance, ein Misslingen hat oftmals den Tod des Konzepts zur Folge. Ein Beispiel für ein Scheitern ist WAP, das Wireless Application Protocol. In den Jahren 1999 bis 2001 stets in den Medien, scheiterte es zunächst an den nur spärlich vorhandenen Mobiltelefonen und später an den sagenhaft schlechten Übertragungsraten. Trotz einer wirklich guten Technologie ist WAP heute nur eine Randnotiz der Internetgeschichte und dient nur noch als traische Anekdote für eine Mischung aus vorschneller, undifferenzierter Begeisterung und unzureichender Unterstützung sowohl von Hersteller- als auch von Anwenderseite.

Einige Stimmen befürchten, dass SVG, dem Scalable Vector Format, ein ähnliches Schicksal bevorstehen könnte. Das vom W3C absegnete Format für Vektoranimationen im World Wide Web hat eine technologisch gute Grundlage. Die Frage ist nur, wie der nächste Schritt aussieht: Gibt es demnächst sinnvolle und überzeugende SVG-Anwendungen? Wann wird das Format auch auf wirklich großen Websites eingesetzt, wann wird das Stadium der reinen Demos wirklich verlassen?

Eine gute Messlatte sind die verschiedenen Publikationen zu diesem Thema. Nur eine Handvoll Autoren beschäftigt sich mit der Technologie. Die Reaktion auf Artikel ist zahlenmäßig allerdings mau. In vielen Fällen melden sich lediglich vermeintliche "Mitstreiter", bedanken sich für die "Unterstützung" (sic) und bemäkeln jedes kritische Wort. So wird sich das Format sicherlich nie durchsetzen, das steht zweifelsohne fest.

Dieser Beitrag zeigt an einigen Beispielen, wozu SVG fähig ist. Dabei wird häufig ein Vergleich zum SWF, dem Small Web Format gezogen. Dieses halbproprietäre Format wird von der Firma Macromedia entwickelt; deren Flaggschiff-Editor "Macromedia Flash" diese so genannten Flash-Filme erzeugen kann. Halbproprietär ist das Format deswegen, weil Macromedia die Formatspezifikation zwar veröffentlicht, allerdings das Format unter der eigenen, strikten Kontrolle hat. Zunächst als stumpfes Format für nervige, zappelnde Website-Intros verpönt, ist der Feature-Umfang mittlerweile stark angewachsen und nicht nur Spiele, sondern auch professionelle Anwendungen möglich. Das Versprechen, dass die verschiedenen Browser-(In)kompatibilitäten durch das SWF-Format umgangen werden können, ist sehr verlockend. Hier verspricht SVG Ähnliches - sowohl SWF als auch

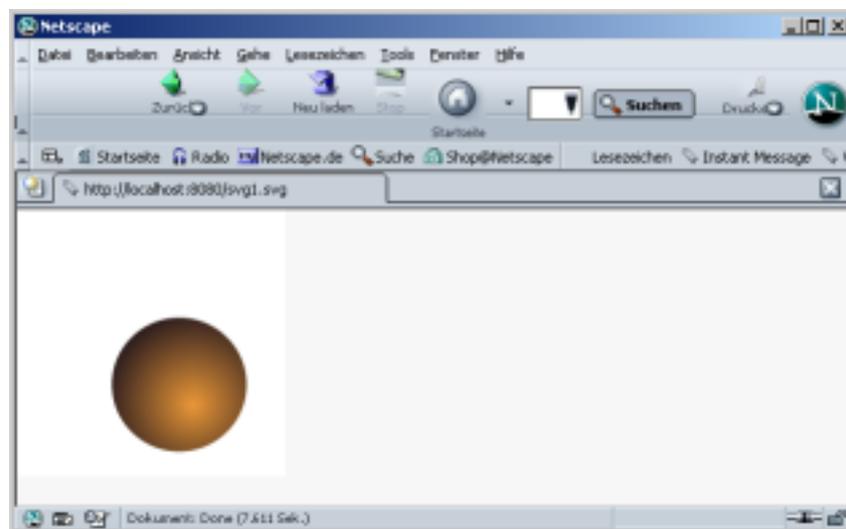
SVG werden durch ein eigenes Plugin im Browser dargestellt; Browser-Bugs sind somit in der Regel nicht entscheidend, da sich alles innerhalb des eingebetteten Bereichs abspielt.

SVG-Elemente

Als Vektorgrafikformat unterstützt SVG alle primitiven geometrischen Figuren und bietet so maximale Flexibilität, da sich alles auf Basis dieser Formen darstellen lässt. Ebenfalls sind zahlreiche Filter möglich. Dritte große Stärke sind die Animationsmöglichkeiten, welche auch ohne Programmierkenntnisse erstaunliche Effekte ermöglichen. Hier ein illustratives Beispiel:

```
<?xml version="1.0" ?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">

<svg width="200px" height="200px">
  <defs>
    <radialGradient id="orange" cx="60%" cy="60%" r="80%" fx="60%" fy="65%">
      <stop offset="0%" style="stop-color:rgb(234,152,55)" />
      <stop offset="100%" style="stop-color:rgb(0,0,25)" />
    </radialGradient>
  </defs>
  <circle id="ball" cx="100" cy="-50" r="50" style="fill:url(#orange)">
    <animateMotion dur="5s" path="M0 0 Q10 400 50 -50" />
  </circle>
</svg>
```



SVG-Scripting

Statische Animationen alleine können nicht dafür sorgen, den Platzhirschen Flash auch nur ansatzweise zu verdrängen; Dynamik muss her. SVG setzt hier auf einen weiteren W3C-Standard, ECMAScript. Durch das eigene DOM eines SVG-Dokuments ist es möglich, auf jedes Element in der SVG-Datei per Skript zuzugreifen und zu modifizieren. Ein klarer Vorteil gegenüber Flash, wo der Film nicht in einer hierarchischen Baumstruktur zur Verfügung steht. Somit sind - entsprechende Programmierkenntnisse vorausgesetzt - den Bemühungen, dynamische SVG-Applikationen zu erstellen, kaum Grenzen gesetzt.

In diesem Vorteil liegt allerdings auch gleichzeitig ein großer Nachteil begründet: So flexibel die Animationen auch sein mögen, Nicht-Programmierer haben hier kaum eine Chance, ansprechende Ergebnisse zu erzielen. Da

die Editoren-Situation zurzeit eher trist ist, ist für die meisten Grafiker zurzeit ein Umstieg auf SVG nicht attraktiv genug. Der Flash-Editor ermöglicht es dagegen sogar, bestimmte Befehlsabläufe innerhalb eines Flash-Films "zusammenzuklicken".

Ein weiteres großes Manko von SVG sind Tastatureingaben. Die Behandlung von Tastatur-Ereignissen ist im DOM-2-Standard, der von SVG unterstützt wird, nicht vorgesehen. Der Adobe SVG Viewer unterstützt dennoch Keyboard-Events, allerdings ist dies ein Entgegenkommen der ASV-Entwickler, nicht Teil des SVG-Standards. Eine weitere sinnvolle, aber nicht standardgemäße Erweiterung des ASV ist die Möglichkeit, im Hintergrund serverseitig Daten auszutauschen und somit eine Interaktion mit dem Webserver durchzuführen, ohne dass die HTML-Seite, die das SVG-Dokument enthält, neu geladen werden muss. Dies ist in Flash schon seit Längerem möglich, bei SVG jedoch nur mit dem ASV - noch.

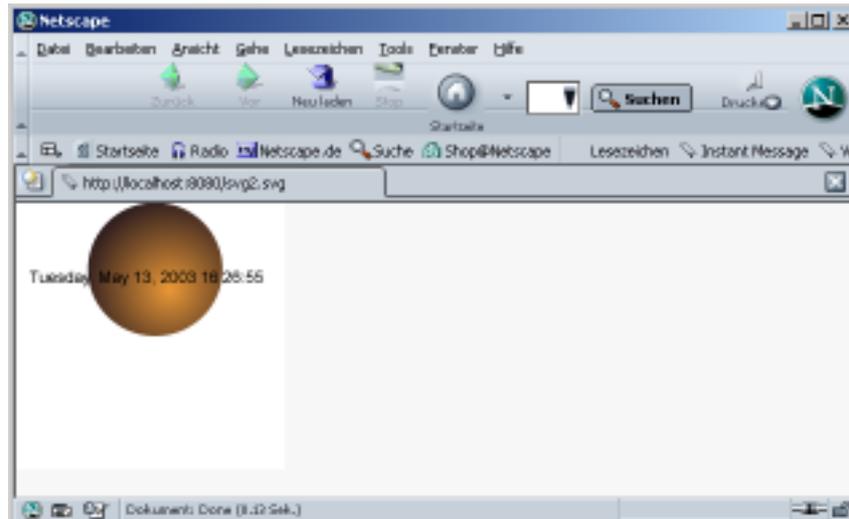
Hier ein einfacheres Scripting-Beispiel: Auf den animierten Ball wird die aktuelle Uhrzeit geschrieben:

```
<?xml version="1.0" ?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">

<svg width="200px" height="200px" onload="init(evt);">
  <defs>
    <radialGradient id="orange" cx="60%" cy="60%" r="80%" fx="60%" fy="65%">
      <stop offset="0%" style="stop-color:rgb(234,152,55)" />
      <stop offset="100%" style="stop-color:rgb(0,0,25)" />
    </radialGradient>
    <script type="text/ecmascript"><![CDATA[
      var svgdoc;

      function init(evt) {
        svgdoc = evt.getTarget().getOwnerDocument();
        setInterval("update()", 1000);
      }

      function update() {
        var d1 = svgdoc.getElementById("datum");
        var d2 = (new Date()).toLocaleString();
        d1.firstChild().setData(d2);
      }
    ]]></script>
  </defs>
  <circle id="ball" cx="100" cy="-50" r="50" style="fill:url(#orange)">
    <animateMotion dur="5s" path="M0 0 Q10 400 50 -50" />
  </circle>
  <text id="datum" x="10" y="60"> </text>
</svg>
```



Serverseitiges SVG

Wenngleich die serverseitige Interaktion von SVG nur mit Schwierigkeiten möglich ist, die XML-Basis des Formats bringen hier deutliche Vorteile. Flash-Filme lassen sich nur mühsam und eingeschränkt von einem serverseitigen Skript erstellen, die Ming-Bibliothek ist ein Beispiel hierfür. Bei SVG sieht es natürlich komplett anders aus. Das Skript muss nur Text ausgeben, schon wurde SVG erzeugt. Für diverse Programmiersprachen gibt es fest stehende Module, mit denen SVG noch bequemer erstellt werden kann. Ein Beispiel hierfür ist SVG.pm von Ronan Oger, ein Perl-Modul das einen objektorientierten Zugriff auf ein SVG-Dokument bietet.

Das einzige, was hierbei beachtet werden muss, ist dass der korrekte MIME-Typ für das SVG-Dokument angegeben wird. Wird das unterlassen, geht der Browser davon aus, dass es sich um ein HTML-Dokument handelt; das Plugin kann dann die Inhalte zumeist nicht korrekt darstellen. Um den MIME-Typ an den Browser zu schicken, muss im HTTP-Header der folgende Eintrag integriert werden:

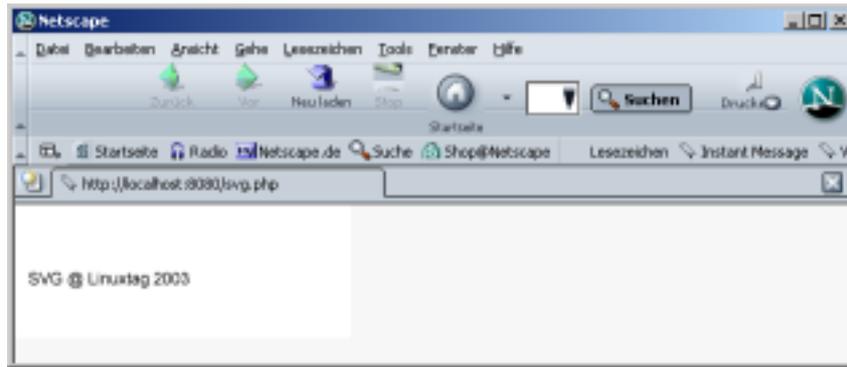
```
Content-type: image/svg+xml
```

Mit PHP geht das beispielsweise folgendermaßen:

```
header("Content-type: image/svg+xml");
```

Hier ein komplettes PHP-Beispiel, das ein bewusst simpel gehaltenes SVG-Dokument ausgibt:

```
<?php
header("Content-type: text/svg+xml");
echo("<?xml version='1.0' ?>\n");
?>
<svg width="250px" height="100px">
  <text x="10" y="60">SVG @ Linuxtag 2003</text>
</svg>
```



Plugin-Situation

Entscheidend für den Erfolg oder Misserfolg einer Technologie ist, wie die Geschichte zeigt, nicht immer die Qualität, sondern letztendlich die Verbreitung. Hier hat Macromedia Flash zurzeit deutlich das Plus auf seiner Seite. Laut unabhängigen Erhebungen haben 98% aller Web-Clients einen Flash-Player installiert; bei der neuesten Version 6 des Plugins sind es immerhin über 60%. Während es für SWF nur ein Plugin gibt - das offizielle von Macromedia - sieht es für SVG zumindest zahlenmäßig besser aus. Flaggschiff ist der Adobe SVG Viewer (kurz: ASV), aktuell in Version 3 erhältlich, Version 4 angeblich im Beta-Stadium. Seit einiger Zeit hat auch Corel mit dem Corel SVG Viewer nachgezogen. Ebenfalls relativ bekannt ist Squiggle, der SVG-Browser des Batik-Apache-Projekts. Allerdings ist lediglich der ASV relevant. Corel hat bis dato erst eine einzige Vorab-Version veröffentlicht; Squiggle ist ein Standalone-Programm, das zudem noch eine Java-Laufzeitumgebung benötigt, als für den Massen-Web-Markt untauglich ist.

Zu diesem SVG-Plugin gibt es noch keine aussagekräftigen Verbreitungsstatistiken, Adobe und Corel veröffentlichen auch keine exakten Download-Zahlen. Es ist allerdings davon auszugehen, dass der Anteil noch im einstelligen Bereich vor sich hin dümpelt. Da noch keine große Website auf das SVG-Plugin setzt, ist die Bereitschaft der großen Masse, 3 MB herunter zu laden, nicht sonderlich groß; es sind also hauptsächlich Entwickler und "Early Adopters", die das Anzeigeprogramm einsetzen. Die teilweise kolportierte Behauptung, bei der Installation des Adobe Acrobat Readers (Verbreitung übrigens zwischen 50% und 60%) würde das SVG-Modul automatisch in den Browser integriert werden, ist schlichtweg falsch, wie ein einfacher Test beweist.

Aber es gibt weitere Gründe für die verheerende Plugin-Situation. Die einzigen offiziell unterstützten Plattformen sind der Netscape Navigator 4 (auf General-Interest-Sites immer noch en par mit Netscape 6, 7, Mozilla und Abkömmlingen) und der Internet Explorer - unter Windows und Mac. Neuere Mozilla-Derivate lassen sich mit etwas Tricksen zumindest zur allgemeinen Kooperation bewegen - was jedoch sicherlich nicht Massenmarkt-kompatibel ist. Schlimmer noch: Sobald versucht wird, zwischen HTML-Seite (und eingebettetem JavaScript-Code) und integriertem SVG-Dokument eine Kommunikation aufzubauen, scheitert dies - häufig stürzt der Browser mit Bravour ab. In diesem Zusammenhang gibt es eine traurige Anekdote, die in einer Diskussion zwischen Adobe- und Mozilla-Entwicklern kumuliert. Für die experimentelle Anpassung des Plugins auf Mozilla wurden die aktuellen, aber noch nicht als "frozen" gekennzeichneten APIs des Open-Source-Browsers genutzt. Diese APIs wurden später jedoch geändert. Viel schlimmer noch: Laut Aussage von Adobe machen die neuen APIs eine Zusammenarbeit mit dem Adobe SVG Viewer auch in Zukunft sehr unwahrscheinlich. Corel geht noch einen Schritt weiter als Adobe. Statt einer "experimentellen Unterstützung" werden neue Netscape-Versionen (offiziell) schlicht überhaupt nicht unterstützt.

Auch aus diesem Grund wird bei Mozilla zurzeit daran gearbeitet, dass eine der nächsten Versionen eine komplette, native SVG-Unterstützung mit sich bringt. Die ersten Früchte dieser Bemühungen sind bereits sichtbar, können sich aber noch nicht mit dem messen, was ASV vorgelegt hat - die Messlatte liegt hoch.

Editoren-Situation

Weiterhin von großer Bedeutung in Hinblick auf die Durchsetzung eines Formats ist die Qualität geeigneter Editoren. SVG hat hier zunächst den Vorteil aufzuweisen, dass überhaupt kein spezieller Editor nötig ist. SVG ist als Plain-Text-XML mit vi oder Emacs komplett zu erstellen. Dies ist allerdings nur auf den ersten Blick ein

Vorzug des Formats. Spätestens, wenn komplexere Animationen erstellt oder Elemente pixelgenau positioniert werden müssen, ist ohne einen Editor kaum etwas zu machen. Einer der wenigen wirklich ernst zu nehmenden Vertreter in diesem Gebiet ist Jasc WebDraw. Dieser bietet unter anderem eine Zeitleiste, ein Muss beim Erstellen von Animationen. Allerdings ist die Scripting-Unterstützung noch als ausbaufähig zu betrachten. Mit Spannung erwartet wurde außerdem die neue Version von LiveMotion, dem Animationstool von dem Hersteller des ASV. Als schließlich die zugehörige Pressemitteilung erschienen ist, folgte die große Ernüchterung: SVG wurde mit keinem einzigen Wort erwähnt.

Auf der Flash-Seite sieht es wiederum ganz anders aus. Macromedia Flash MX ist mittlerweile eine ausgereifte Entwicklungsumgebung, mit der die Erstellung von Flash-Filmen wirklich sehr einfach fällt. Natürlich liegt das auch daran, dass an der Software schon seit mehreren Versionen gearbeitet wird - die interne Versionsnummer von Flash MX ist 6. Hier besteht für SVG noch großer Aufholbedarf.

Fazit

SVG muss seinen Open-Source-Elfenbeinturm verlassen, um mit Flash ernsthaft konkurrieren zu können. Unterstützung von Standards ist eine tolle Sache, dies allein jedoch genügt nicht. Die Plugin-Situation muss sich verbessern, hier ist Adobe am Zug. Außerdem sind gute Editoren gefragt, um der breiten Masse das Erstellen von SVG schmackhaft zu machen. Der Vorsprung von SWF ist enorm, aber die Vergangenheit hat mehrfach gezeigt dass das Internet ein schnelllebiges Medium ist. Als Fazit für SVG bleibt festzuhalten: Eine interessante, zukunftssträchtige Technologie? Auf jeden Fall. The Next Big Thing? Keineswegs.