Thema 4: Vektorbasierte Bildformate (SVG, Flash)

Matthias Stark

Zusammenfassung: *SVG und Flash stellen heute die zwei wichtigsten vektorbasierten Bildformate dar. SVG wurde vor kurzem vom W3C standardisiert während Flash von Macromedia schon seit einigen Jahren seinen Siegeszug v.a. im Web angetreten hat. Deshalb wird im Folgenden der Funktionsumfang und der Aufbau dieser beiden Formate anhand von mehreren Beispielen erläutert und verglichen. Anschließend werden Autorenumgebungen für die Formate und Einsatzgebiete vorgestellt. Die daraus gewonnen Kenntnisse lassen das abschließende Fazit ziehen, dass Flash aufgrund seiner hohen Popularität und Verbreitung im www praktisch eine Monopolstellung inne hat und damit den Standard darstellt. Das noch sehr junge SVG Format wird jedoch in Bälde seine Anhänger finden. Wie so oft zeigt sich, dass nicht ein Format für alle Anwendungen das klar bessere ist. Flash ist mit seiner komfortablen Autorenumgebungen für die breite Anwenderschicht geeignet, die z.B. eine ansprechende Homepage gestalten will. SVG hingegen erschließt auch ganz andere Gebiete als das Internet. Z.B. lassen sich technische Illustrationen oder wissenschaftliche Datenvisualisierungen damit realisieren.*

1 Einleitung

Für den Menschen sind „bildliche“ Eindrücke immer wichtiger als pure Textinformation. Dies liegt daran, dass er solche Wahrnehmungen aufgrund seiner Sinnesorgane sich besser einprägen und deshalb auch besser merken kann. Das Internet, das lange Zeit nur aus Text bestand, war deshalb gänzlich ungeeignet um einer breiten Anwenderschicht zum einen einen „angenehmen Surfgenuss“ zu bieten, zum anderen etwa der Werbeindustrie bzw. Unternehmen und Firmen bei den Benutzer des Mediums Internet einen bleibenden Eindruck zu hinterlassen. Deshalb mussten Möglichkeiten gefunden werden, um das Internet „lebendig“ zu machen, d.h. vor allem multimediale Inhalte wie z.B. Grafiken, Animationen und Sounds einzubinden. Das Problem war dabei natürlich der Speicherbedarf, den diese Elemente „verschlingen“, und die daraus resultierenden langen Ladezeiten für Webseiten. Es musste also ein schlankes Format entwickelt werden, das trotzdem den nötigen Anforderungen entsprach. Es entstanden die vektorbasierten Bildformate, die, da sie nur wenige Informationen abspeichern, sehr schlank sind, aber gleichzeitig auch so flexibel, dass sie die gewünschten Anforderungen entsprechen.

2 Vergleich von Pixel- und Vektorgrafik

Grundsätzlich gibt es zwei Möglichkeiten Bilder auf dem Computer abzuspeichern. Zum eine kam man jeden Pixel[1], aus dem das Bild besteht, einzeln abspeichern. Zum

[1] Kleinste Einheit eines Bildes oder Bildschirmes. Ein digitales Bild besteht immer aus einer Anzahl von Bildpunkten, die Pixel genannt werden. [Gal03]

anderen kann man versuchen das Bild in Vektoren - also geometrische Formen - zu zerlegen und dann nur die mathematischen Formeln der Vektoren zu speichern[2]. In den nächsten beiden Abschnitten wollen wir deshalb diese beiden Verfahren genauer unter die Lupe nehmen.

2.1 Pixelgrafik

Pixelbasierte Bildformate sind am weitesten verbreitet. Die bekanntesten Vertreter sind JPEG, GIF, PNG und TIFF. Mit JPEG (Joint Photographic Experts Group) kann man Bilder komprimieren, um sie z.B. mit geringem Speicherbedarf im Internet zu präsentieren. GIF (Graphic Interchange Format) hingegen kann geometrisch aufgebaute Grafiken mit wenigen Farbübergängen Platz sparend abspeichern und unterstützt transparente Hintergründe. Zusätzlich kann man damit Animationen erstellen. PNG (Portable Network Graphics) sollte die Vorteile des JPEG und GIF Formats vereinen und den Qualitätsverlust bei der Komprimierung verringern. TIFF (Tagged Image File Format) speichert Bilder ohne Qualitätsverlust und erlaubt das Speichern von vielen zusätzlichen Parametern.

Bei einem pixelbasierten Bildformat werden nach einem Raster, das z.B. beim Scannen angelegt wird, Bildpunkt für Bildpunkt einzeln abgespeichert. Bilder, die in einem solchen Bildformat abgespeichert sind, werden immer nur in ihrer ursprünglichen Größe und Auflösung optimal dargestellt. Werden sie vergrößert, so entstehen die bekannten „Treppenstufen“ (Abbildung 1). Diese entstehen dadurch, dass bei der Vergrößerung die einzelnen Bildpunkte vergrößert werden müssen und somit das Bild natürlich grobkörniger wird. Bei zunehmender Vergrößerung sieht der Betrachter schließlich die einzelnen Pixel des Bildes.

Bei vielen Anwendungen ist aber das Pixelformat das einzig sinnvolle Format. Bei Fotos z.B. trägt (fast) jeder Pixel eine andere Farbinformation, die natürlich auch einzeln abgespeichert werden muss. Eine Speicherung eines Fotos mit Hilfe eines vektorbasierten Grafikformats wäre unsinnig, da es seine Vorteile nur bei geometrischen Formen ausspielen kann (Abbildung 2).

2.2 Vektorgrafiken

Das am meisten verbreitete vektorbasierte Grafikformat ist sicherlich Flash. Seit einiger Zeit ist SVG hinzugekommen.

Bei Vektorformaten werden Bilder mittels geometrischen Formen wie Linien, Rechtecke, Kreise usw. dargestellt. Ein Kreis wird z.B. durch seinen Mittelpunkt und seinen Radius charakterisiert. Dazu werden Attribute wie z.B. Linienfarbe- und stärke angegeben. Dadurch müssen weit weniger Informationen, als bei Pixelformaten abgespeichert werden. Das lässt sich schon an einem einfachen Beispiel sehen. Nehmen wir an, wir wollen ein Rechteck abspeichern. Das Vektorformat speichert dazu die Länge, die Breite, die Linienfarbe- und stärke des Rechtecks. Das Pixelformat hingegen speichert Bildpunkt für Bildpunkt. Vergrößert man jetzt das Rechteck und füllt es noch mit einer Farbe, so benötigt das Vektorformat nur ein zusätzliches Attribut, nämlich die Farbe, mit der das Rechteck ausgefüllt wird. Die anderen Werte werden einfach

[2] Die Grundprinzipien von Pixel- und Vektorgrafiken seien hier nur kurz erwähnt. Wer noch mehr erfahren möchte findet unter http://www.blitzmerker.i-d.de/pdf/03_pixel.pdf eine gut verständliche Einführung. Außerdem werden die einzelnen Bildformate anhand von Bildbeispielen kurz vorgestellt.

abgeändert. Das Pixelformat hingegen muss eine Fülle an zusätzlichen Informationen abspeichern, da sich im vergrößerten Rechteck weitaus mehr Bildpunkte befinden.

Ein weiterer Vorteil ist, dass Vektorgrafiken beliebig skalierbar sind. D.h. Grafiken können beliebig vergrößert werden, ohne dass sie an Qualität einbüßen (Abbildung 1). Das geschieht einfach dadurch, dass der Computer die Grafik durch die im Grafikformat angegebenen Daten neu berechnet. Dies nimmt natürlich wiederum Rechenleistung in Anspruch, was bei komplizierten Grafiken nicht zu unterschätzen ist.

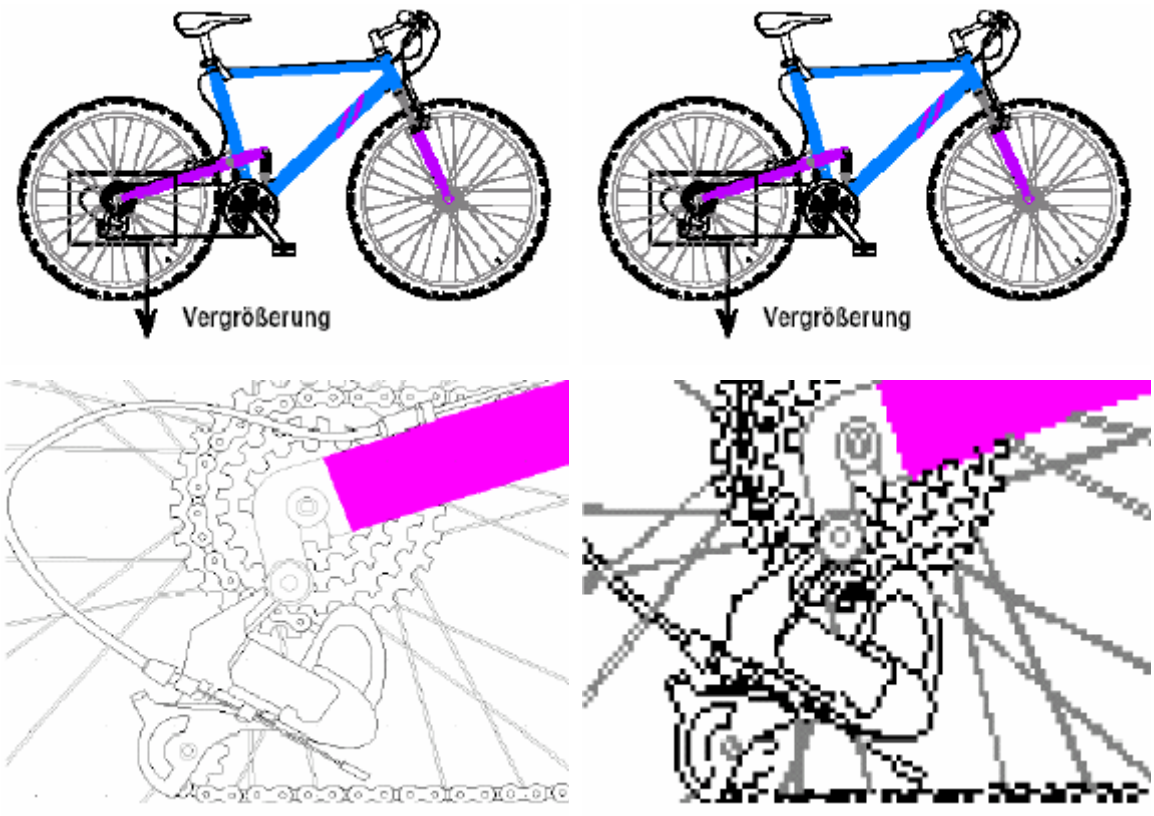

Abbildung 1 : links Vektorgrafik, rechts Pixelgrafik. Man sieht, dass bei der Vergrößerung die Vektorgrafik durch Neuberechnung „in bester Qualität“ dargestellt wird, während bei der Pixelgrafik „Treppenstufen“ und Pixel sichtbar werden. [Blitz01]

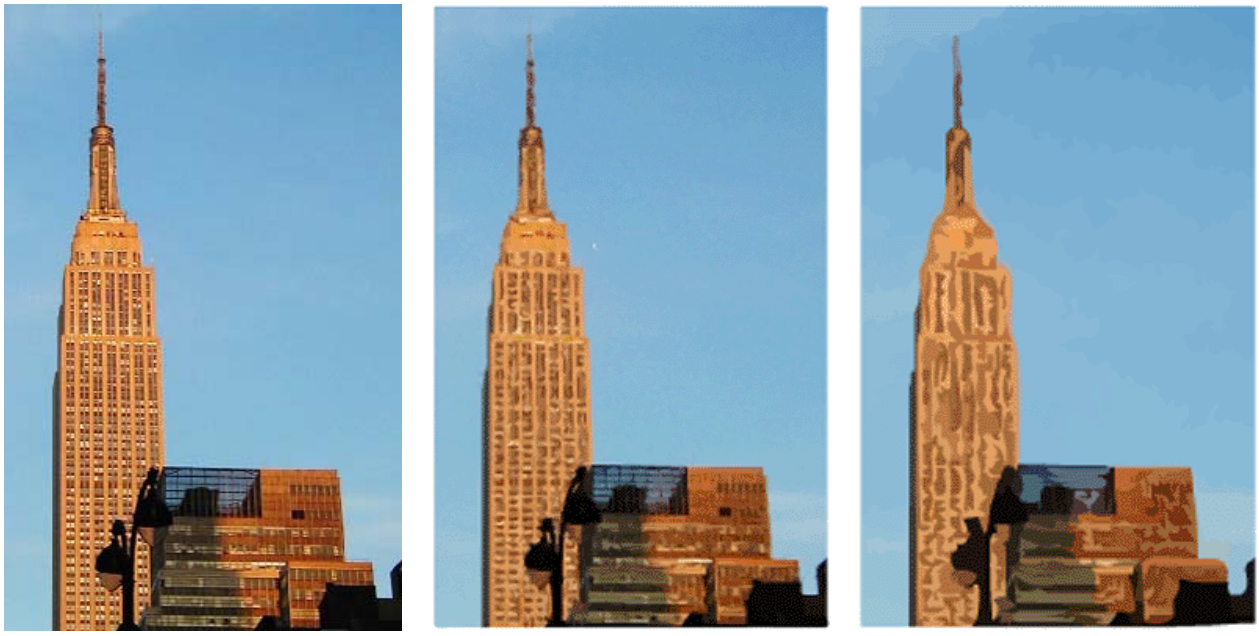

Abbildung 2: links Pixelgrafik, Mitte u. rechts: Pixelgrafik wurde in Vektorgrafik umgewandelt[3]. Obwohl sich das Bild relativ gut als Vektorgrafik eignet, da es sehr geometrisch aufgebaut ist, sieht man besonders rechts wie bei der Vektorgrafik viele Details zu geometrischen Formen zusammengefasst werden. [Kerman02]

3 Macromedia Flash

Zuerst betrachten wir kurz die Entwicklungsgeschichte von Flash. Dann gehen wir auf die Eigenschaften des Grafikformats ein. Danach werfen wir an Hand von Beispielen einen Blick auf die Gestaltungsmöglichkeiten, die uns Flash bietet. Zum Schluss stellen wir noch das „eigentliche Programm Flash“ und seine Programmiersprache ActionScript vor. Außerdem wird kurz erläutert wie Flash mit Hilfe eines „Generators“ dynamische Seiteninhalte darstellt.

3.1 Entwicklungsgeschichte

Jonathan Gay entwickelte mit seiner Firma FutureWave den „FutureSplash Animator“, ein Grafikprogramm mit Animationseigenschaften. FutureWave wurde von Macromedia 1996 aufgekauft und es entstand Flash 1.0 [Stark01]. Flash wurde daraufhin weiterentwickelt und ist momentan in der aktuellsten 7. Version[4], die jetzt Flash MX 2004 heißt, verfügbar.

3.2 Das Dateiformat

Betrachtet man das Dateiformat von Flash, so stellt man schnell fest, dass es sich eigentlich um zwei Formate handelt. Zum einen ist dies FLA (FlashAuthor), zum anderen SWF (ShockwaveFlash). Die FLA-Dateien sind sozusagen die Quelldateien. Beim Erzeugen einer Flash-Datei, wird diese als FLA-Datei gespeichert. In diesem Format sind Änderungen an der Datei möglich. Ist die Arbeit abgeschlossen und soll die

[3] Die Umwandlung fand in Marcomedia Flash MX statt.
Mitte: Farbschwelle: 1, kleinste Fläche: 10; rechts: Farbschwelle: 10, kleinste Fläche: 100
[4] Stand: November 2003

Flash-Datei veröffentlicht werden, so wird die FLA-Datei in eine SWF-Datei exportiert, die dann in einem Flash-Player[5] abgespielt, jedoch nicht mehr überarbeitet werden kann.

3.2.1 Anforderungen

Das Flash-Format wurde entwickelt um Vektorgrafiken und Animationen schnell und einfach im Internet übertragen zu können. Deshalb wurden folgende Anforderungen gestellt:

- **Bildschirmdarstellung:** Flash sollte hauptsächlich auf dem Bildschirm dargestellt werden. Es unterstützt deshalb Anti-Aliasing[6], schnelles Rendern[7], Animation und interaktive Schaltknöpfe.
- **Erweiterbarkeit:** Flash-Dateien werden durch „Tag-Blöcke[8]“ aufgebaut. Es können neue „Tags“ hinzugefügt werden, ohne dass dadurch die Abwärtskompatibilität verloren geht (Alte Flash-Player ignorieren einfach unbekannte „Tags“).
- **Schnelle Netzwerkübertragung:** Die Dateien müssen auch über langsame Netzwerkverbindungen wie z.B. Modems schnell übertragen werden können. Das wird dadurch erreicht, dass SWF-Dateien stark komprimiert (binäres Dateiformat) und streamingfähig[9] sind.
- **Einfachheit:** Das Format ist von seiner Struktur her einfach aufgebaut und alle Elemente (Animation, Bilder, Sound…) liegen in einer Datei. Somit sind Flash-Player sehr klein[10] und können schnell über das Internet übertragen werden.
- **Plattformunabhängigkeit:** Durch die Verwendung eines Flash-Players zum Abspielen des Films ist man vom Betriebsystems, vom Browser usw. unabhängig.
- **Skalierbarkeit:** Die Filme sollen auf unterschiedlichen Ausgabegeräten wie z.B. Handydisplays, Monitore (mit unterschiedlichen Auflösungen) usw. korrekt dargestellt werden.
- **Geschwindigkeit:** Schnelles Rendern bei hoher Qualität.

[Stark01]

Ein Konzept zur Umsetzung dieser Vorgaben ist z.B. die Kompaktheit:

---

[5] siehe hierzu auch Abschnitt 3.4.23.4.2 Player

[6] Kantenglättung zur Verminderung des Treppeneffekts durch Interpolation, d.h. Hinzufügen von ähnlichen Farbwerten.

[7] Im Zusammenhang mit Flash/SVG bezeichnet Rendern die Umsetzung der mathematischen Beschreibung der Vektorgrafik in eine Pixeldarstellung.

[8] Tags unterteilen sich in Start- und Ende-Tags und markieren den Anfang und das Ende des Gültigkeitsbereichs eines Elements. [Gal03]

[9] Technologie, bei der Inhalte aus dem Internet angezeigt werden, während sie noch geladen werden. [Gal03]

[10] Ein Flash-Player ist gerade 394 KB groß. [Player03]

- **Wiederverwendbarkeit:** Durch das Prinzip des „Dictionary“ müssen mehrfach verwendete Objekte nur einmal abgespeichert werden.
- **Default-Werte:** Werden Standardwerte für bestimmte Attribute bei einem Objekt verwendet, so müssen diese nicht extra angegeben werden.
- **Change-Encoding:** Bei Animationen werden nur die relativen Veränderungen zum ursprünglichen Objekt abgespeichert und nicht das komplette neue Objekt.
- **Shape-Datenstruktur:** Die meisten Shapes (Formen) beruhen auf der gleichen Datenstruktur, was eine Platzersparnis beim Kodieren sowie ein schnelles Rendern möglich macht.

[Fünder03]

3.2.2 Aufbau

SWF-Dateien bestehen aus einem Kopf (Header) und den danach folgenden „Tag-Blöcken“, die durch einen „End-Tag“ abgeschlossen werden. Der Kopf besteht aus folgenden Informationen: Signatur (SWF), Versionsnummer, Dateigröße, Breite und Größe des Flash-Films, Bilder pro Sekunde (Framerate), Gesamtzahl der Bilder (Frames) in der Animation.

Abbildung 3: Dateistruktur einer SWF-Datei [Fünder03]

Grundsätzlich muss man zwei Arten von „Tags“ unterscheiden. Einerseits gibt es „Definition-tags“, andererseits „Control-Tags“. „Definition-Tag“ beschreiben Grafikobjekte (Linien, Rechtecke, Kreise, …), importierte Pixelgrafiken und Sounds. Wird ein Element definiert, so bekommt es eine eindeutige Kennung (ID) zugewiesen und wird in einem „Dictionary“ hinterlegt. „Control-Tag“ werden dazu verwendet Objekte zu verändern und die Animation zu steuern. So kann z.B. einer Grafik eine andere Farbe zugewiesen werden oder ein Sound gestartet werden. „Control-Tags“ greifen auf die „Characters“ im „Dictionary“ zu, um die nötigen Informationen über die zu verändernden Grafikobjekte zu erhalten.

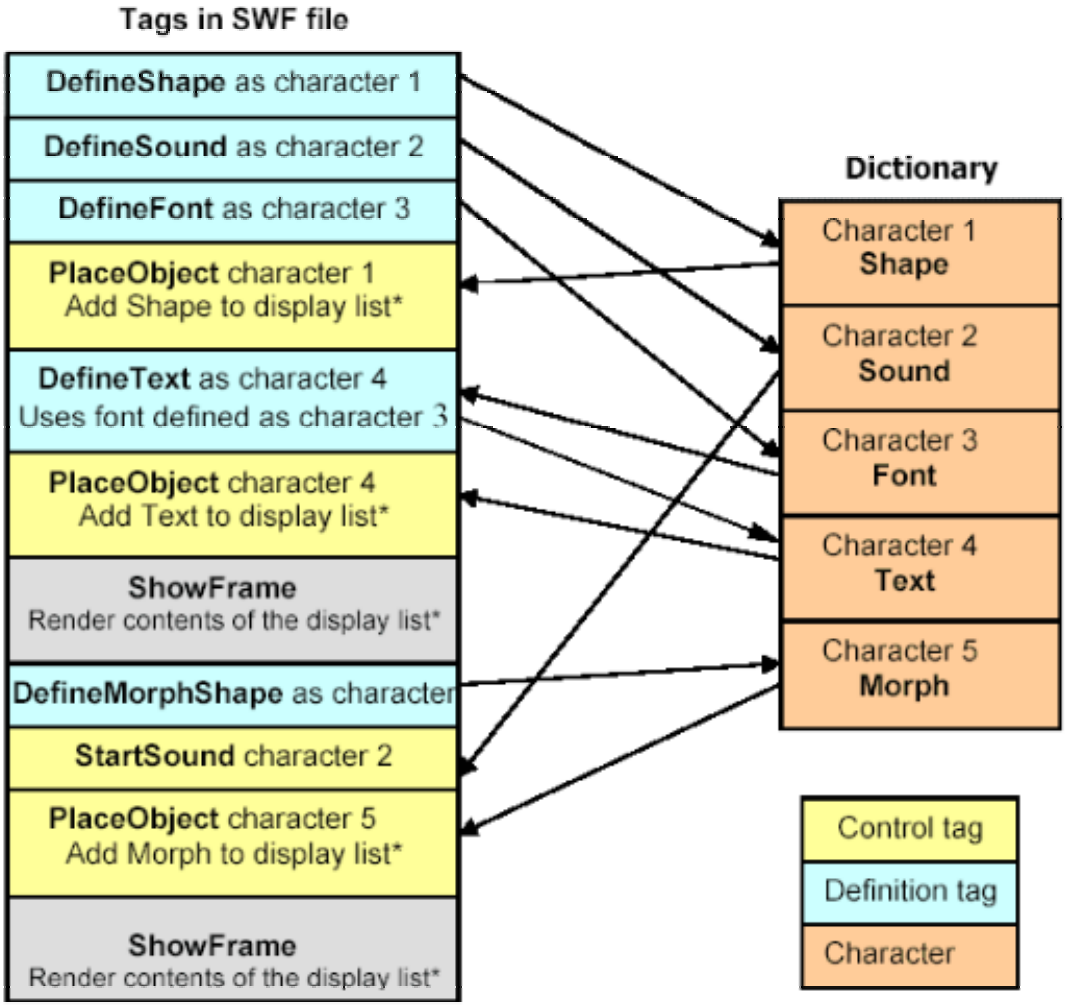

Abbildung 4: Interaktion zwischen „Definition-Tags“, „Control-Tags“ und „Dictionary“ [Fünder03]

3.2.3 Vom proprietären Format zu SWF-SDK

Lange Zeit war SWF ein „proprietäres[11] Format“, d.h. Macromedia hielt die Spezifikation „geheim“. Um jedoch die „Vormachstellung“ ihres Formats für Vektorgrafiken und Animationen zu sichern, veröffentlichte Macromedia zunächst 1998 die Spezifikation des Dateiformats und den Quellcode des Flash-Players. Im Jahr 2000 überarbeitete Macromedia dann diese und stellte eine Programmierschnittstelle zur Erzeugung von Flash-Animationen bereit. Mit SWF-SDK (Software Development Kit) war es auch andere Firmen möglich Autorenumgebungen und Player für Flash zu entwickeln. Daraufhin implementierten u.a. Adobe im Illustrator 9 und Corel in CorelDRAW 10 das SWF Format. Macromedia behält sich aber weiter vor das Format weiterzuentwickeln.

[11] Proprietär: Eigentümer

3.3 Funktionsumfang

Flash stellt nicht nur ein Vektorgrafikformat dar, sondern man kann auch „interaktive Vektorgrafiken und Animationen“ erstellen. Nachdem anfangs (1996) wegen geringer Bandbreite bei der Internetanbindung nur einfache Navigationselemente wie z.B. interaktive Buttons erstellt wurden, sind heute komplette Websites, Präsentationen und Werbespots „geflasht“. Dafür werden Sounds im MP3-Format, komplexe Animationen und Techniken wie Bewegungs- und Form-Tweening verwendet. Deshalb spricht man auch oft von „Flash-Filmen“. [Stark01]

3.3.1 Einzelbild- oder „Bild-für-Bild“ Animation

Eine Einzelbildanimation kann man mit einem Daumenkino vergleichen, bei dem sich nur sehr gering voneinander abweichende Bilder auf den nacheinander befindlichen Seiten befinden. Lässt man die Bilder jetzt schnell nacheinander „über den Daumen laufen“, so erscheinen die vielen schnell aufeinander folgende Bilder wie ein Film. Genau so wird dies bei Flash verwirklicht. Man erstellt viele ähnliche Grafiken, die bei schneller Abfolge[12] wir ein Film wirken. Die mühsame Arbeit, diese vielen Grafiken zu erstellen, wird von Flash durch den „Zwiebelschalenmodus“ unterstützt. Wird dieser aktiviert, so sieht man die im zeitlichen Filmablauf vor und nach dem gerade in Bearbeitung befindlichen Bild vorkommenden Bilder.

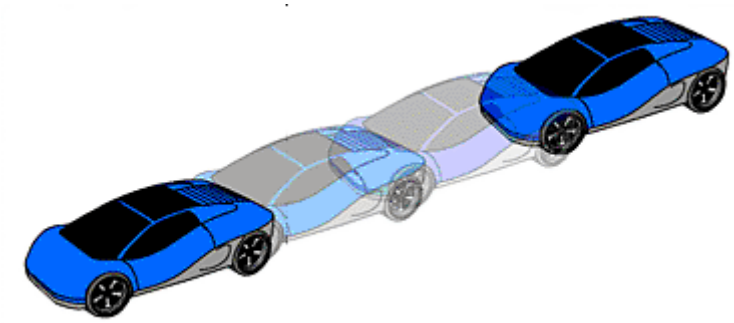

Abbildung 5: Mehrere Einzelbilder (hier im Ziebelschalenmodus) hintereinander ergeben ein fahrendes Auto. [Hilfe02]

3.3.2 Bewegungs-Tweening

Wie wir an vorigem Beispiel gesehen haben sind Bewegungen ziemlich mühsam zu generieren, da viele Einzelbilder erstellt werden müssen. Flash kann uns dabei aber den größten Teil der Arbeit abnehmen, indem wir die Start- und Endposition eines Objekts vorgeben. Jetzt müssen wir nur noch die Art der Bewegung definieren und schon berechnet Flash die noch fehlenden Zwischenbilder der Bewegung. Es gibt sieben unterschiedliche Arten von „Bewegungs-Tweening“, bei denen folgende Attribute geändert werden: Position, Skalierung, Drehung, Kippwinkel, Helligkeit, Farbton und Alphawert.

[12] Bereits ab 12 Bilder pro Sekunde wirkt die Bilderfolge als Film.

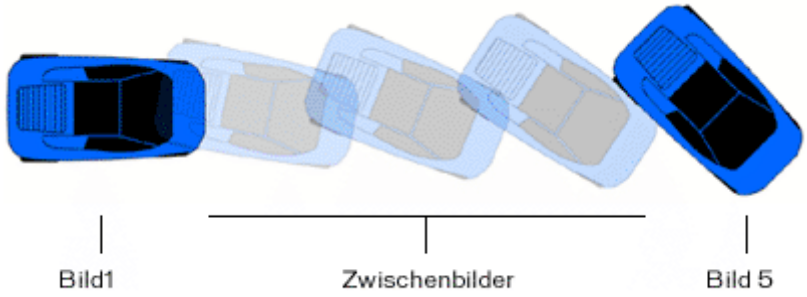

Abbildung 6: Tweening mit Änderung der Position und einer Drehbewegung [Hilfe02]

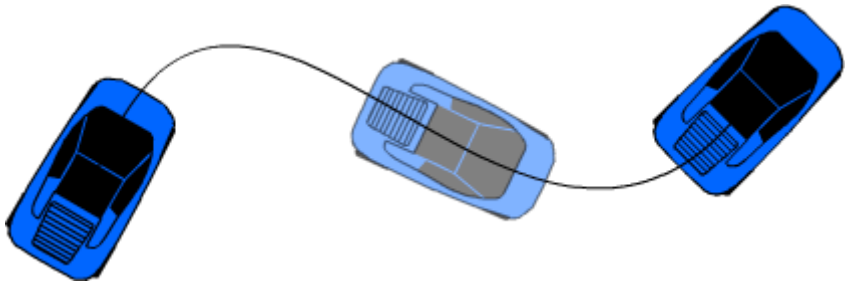

Abbildung 7: Bewegungs-Tweening entlang eines Pfads [Hilfe02]

3.3.3 Form-Tweening / Morph-Effekt

„Ein Morph ist eine Art von Animation, die eine Form auf sehr natürlich aussehende Weise in eine andere Form umwandelt.“ [Kerman01]

Abbildung 8: Morphing vom Hahn zum Schaf [Hilfe02]

3.3.4 ActionScript

Action ist die Progammiersprache von Flash. Wie der Name schon sagt beruht ActionScript auf der Zuweisung von Aktionen an bestimmte Objekte. So kann man z.B. einem Schalter die Aktion zuweisen zu einer anderen Stelle im Film zu springen (goto). Filme laufen deshalb nicht immer linear ab, sondern ein Zuschauer kann sich interaktiv am Filmablauf beteiligen. Weitere Möglichkeiten wären z.B. dass der Anwender den Film starten (play) und stoppen (stopp) kann oder Sounds abspielen kann. Über ActionScript lassen sich also die Flash-Filme steuern. Es lassen sich aber auch die visuellen Eigenschaften von Grafikobjekten ändern oder neue Flash-Filme von anderen Webadressen laden (getURL).

3.4 Programme, die in Verbindung mit Flash eingesetzt bzw. benötigt werden

3.4.1 Autorenumgebungen

Wie bereits erwähnt kann jede Firma nach der Offenlegung des Flash-Formats Autorenumgebungen zur Erzeugung von Flash-Filmen schreiben. Stellvertretend möchte ich kurz das „Original“ Macromedia Flash in der Version 6.0 (Flash MX) vorstellen. Die Oberfläche (Abbildung 4) besteht aus drei Fenstern:

- Auf der **Bühne** werden alle Grafiken bzw. Animationen erstellt und bearbeitet.
- Im **Werkzeugfeld** findet man dazu die nötigen Zeichengeräte und Hilfsmittel.
- In der **Zeitleiste** ist die Abfolge der Bilder für die Animation beschrieben. Es können mehrere Ebenen vorhanden sein, so dass mehrere Animationen gleichzeitig ablaufen können.
- In den **Bedienfeldern** (z.B. dem Eigenschaftenfeld) findet man alle restlichen Funktionen und Informationen, die man zum Erstellen einer Präsentation braucht. Man kann sie frei anordnen oder ausblenden wenn man sie nicht benötigt.

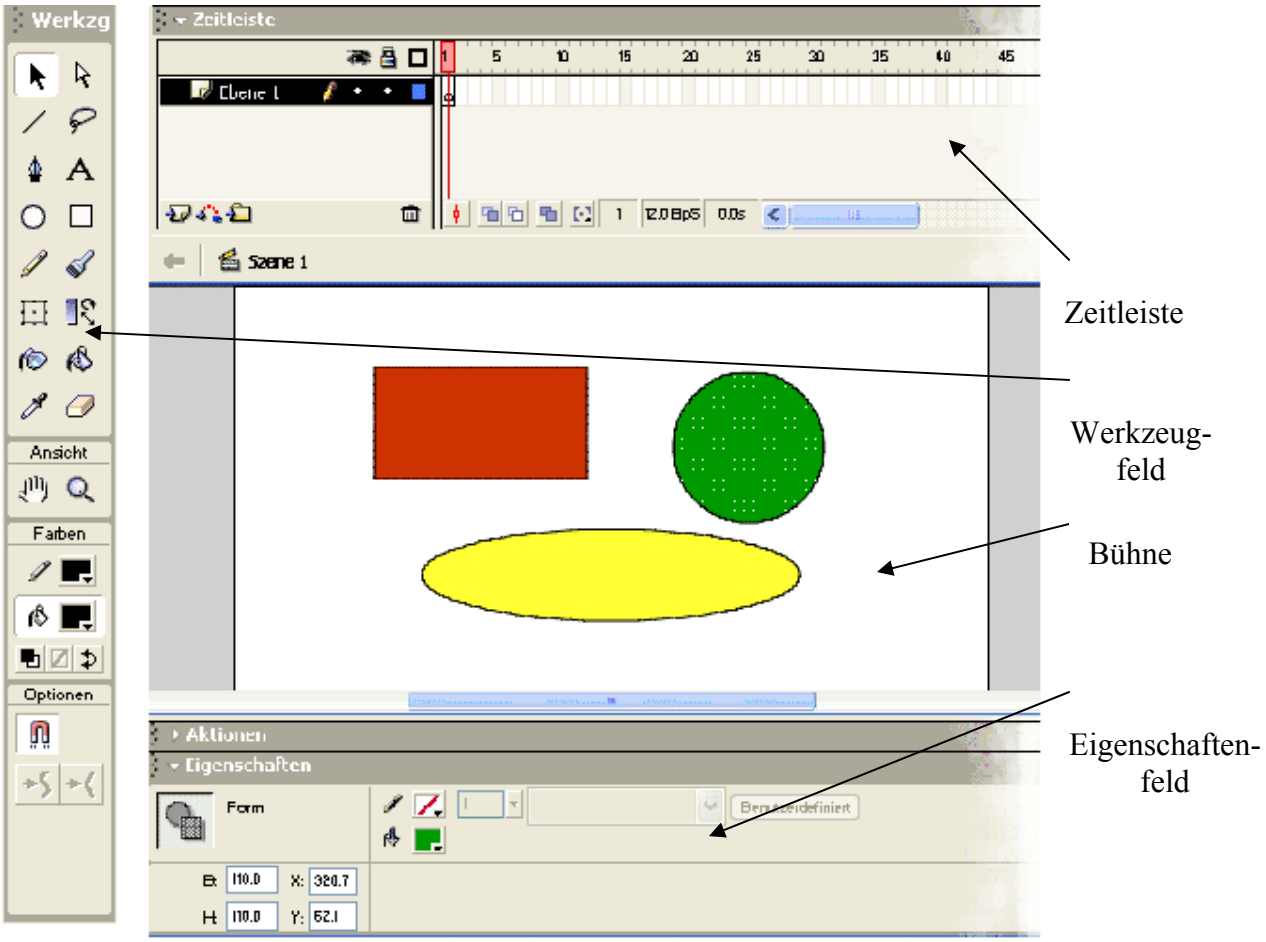

Abbildung 4: Die Arbeitsoberfläche von Flash MX (Version 6.0). Die Zeitleiste, Werkzeug und Eigenschaftenfeld wurden vergrößert, damit sie besser zu erkennen sind.

Auf der Bühne ist der grüne Kreis ausgewählt, weshalb seine Eigenschaften in der unteren Leiste angegeben werden (x-y Koordinaten, Füllfarbe, …).

3.4.2 Player

Zum Abspielen eines Flash-Films wird in der Regel ein Flash-Player benötigt. Dies kann entweder ein Webbrowser-Plugin oder eine Standalone-Version sein. Die Standalone-Version liegt momentan in der 7. Version vor [Player03]. Außerdem kann es sein, dass der Flash-Film mit dem Player verbunden ist, so dass die Datei (*.exe) direkt ausgeführt werden kann. Dies nennt man dann einen Projektor. Bei fast allen aktuellen Browser ist außerdem schon ein Player integriert, so dass Flash-Filme direkt angeschaut werden können. Heute verfügen bereits 97,4%[13] der Internetbenutzer über einen Flash-Player.

3.4.3 Generator

Ohne Hilfsmittel kann Flash keine dynamischen Seiteninhalte, wie z.B. fortlaufend aktualisierte Wetterdaten oder Börsenkurse, darstellen. Hier kommen die Generatoren ins Spiel. Ein Browser fordert von einem Server eine „dynamische“ Flash-Datei an. Auf dem Server ist jedoch nur eine Schablone (Template) der Datei mit den „statischen Daten“ (z.B. verschiedene börsennotierte Unternehmen) hinterlegt. Die Schablone wird jetzt mittels der „Gerneratorsoftware“ auf dem Server gefüllt, indem diese jetzt ebenfalls eine Anfrage an einen anderen Server stellt, der über eine Datenbank mit den angeforderten Daten (z.B. Börsenkurse) verfügt. Der Server schickt die benötigten aktuellen Daten an den Generator, der dann die Schablone mit den angeforderten Daten füllt und die fertige Flash-Datei an den Browser schickt.

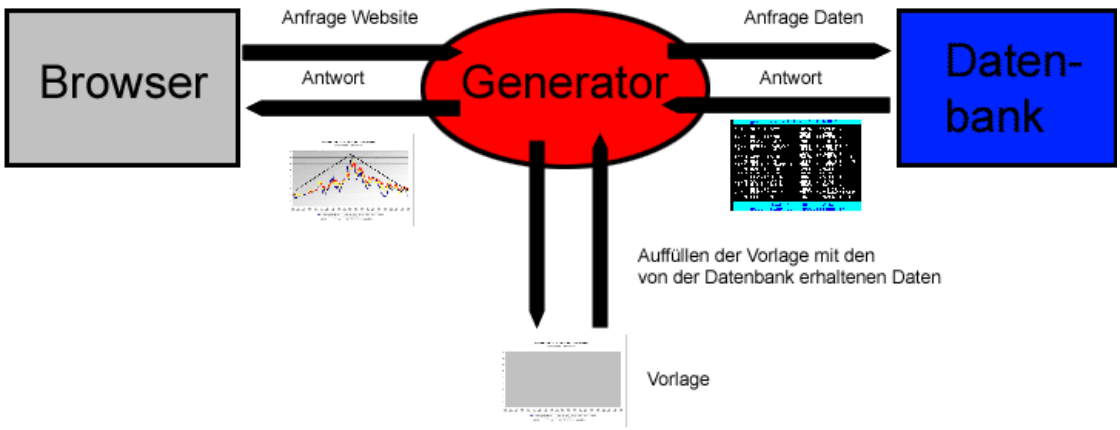

Abbildung 9: Zusammenarbeit zwischen Browser, Generator und Datenbank

4 SVG

SVG steht für „Scalable Vector Graphics“. Dieser noch relativ junge Standard für 2 dimensionale Vektorgrafiken wurde vom W3C (World Wide Web Consortium) im Jahr 2001 freigegeben. Somit stellt SVG einen freien Standard dar, der auf XML basiert und

[13] Stand: Juni 2003 [Player03]

auch die gesamte DOM[14]-Spezifikation unterstützt. SVG ist also sehr „mächtig“ und erlaubt neben dem Erzeugen von Vektorgrafiken auch das Erstellen von interaktiven Animationen und Websites mit dynamischem Inhalt. In den folgenden Ausführungen beziehen wir uns auf die SVG 1.1 Spezifikation.

4.1 Geschichte

1998 gründete das W3C die „Scalable Vector Graphics Arbeitsgemenschaft“. Namhafte Firmen wie Adobe, Apple, Canon, Corel, HP, IBM, Macromedia, Microsoft und Sun sollten auf der Grundlage von den bereits vorhandenen Formaten PGML (Precision Graphics Markup Language) und VML *(Vector Markup Language*) das SVG Format entwickeln. Im Jahr 2000 wurde die erste Test-Suite der Spezifikation herausgegeben, 2001 schließlich SVG 1.0 offiziell veröffentlicht. Bereits ein Jahr später wurde die überarbeitete Spezifikation 1.1 und SVG Mobile für Handys und PDAs bereitgestellt.

4.2 Anwendungsgebiete

Da SVG hochqualitative Vektorgrafiken, Animationen und dynamische Seiteninhalte unterstützt und zudem ein freier Standard ist, ergibt sich ein weites Feld an Einsatzgebiete:

- Webdesign
- Technische Illustrationen / Wissenschaftliche Datenvisualisierung
- Navigationssysteme
- Handys und PDAs (SVG Mobile)
- Druckindustrie (SVG Print)
- Multimedia und Unterhaltung (z.B. E-Commerce, E-Learning ...)
- Austauschformat zwischen Grafikprogrammen und Plattformen

[NeuWin01]

4.3 Grundgerüst und Eigenschaften

SVG beruht auf dem XML Standard und ist somit aus einem „header“ und einem „root-element“ (dem eigentlichen SVG Block) aufgebaut:

```
<?xml version="1.1" encoding="iso-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C/DTD SVG 20001102//EN"
"http://www.w3.org/TR/2000/CR-SVG-20001102/DTD/svg- 20001102.dtd">
<svg width= "[Breite]" height= "[Höhe]">
 ...
 </svg>
```

In der obersten Zeile wird die Versionsnummer (`"1.1"`) und der verwendete Zeichenstandard (`"iso-8859-1"`) festgelegt. Außerdem wird angegeben, dass sich die SVG Datei in Abhängigkeit eines anderen Dokuments befindet (`standalone="no"`). In dem vorliegenden Beispiel ist das DTD (Document Type Definition)[15] wie aus der

[14] SVG ist kompatibel mit dem DOM (Document Object Model) Level 2 [Spez03].
[15] Mit Hilfe von DTD wird von SGML ("Standard Generalized Markup Language") eine konkrete Auszeichnungssprache abgeleitet [NetLex03]. In unserem Fall ist das SVG.

zweiten Zeilen mit der Angabe des öffentlichen „Identifiers“ ersichtlich ist (`DOCTYPE svg PUBLIC "-//W3C/DTD SVG 20001102//EN"`). Schließlich beginnt danach der eigentliche SVG Block (`<svg ... </svg>`), an dessen Anfang die Breite und Höhe des Bildschirmausschnitts definiert sind, in dem die Grafik angezeigt wird (`width= "[Breite]" height= "[Höhe]"`). [Umgeh02]

Da SVG DTD entspricht, ist jede SVG Datei „gültige“ (valid) und „wohlgeformt“ (well-formed).

Im Gegensatz zu Flash ist SVG kein Binärformat sondern ein Textformat. Der Quelltext ist deshalb jederzeit mit einem einfachen Texteditor einsehbar. Das Erstellen oder Abändern von SVG Dateien kann deshalb natürlich auch mittels eines einfachen Texteditors erfolgen.

Neben der Dateiendung „svg“ gibt es auch noch das Format „.svgz“ für gepackte SVG-Inhalte.

4.4 „Render Model“ und Koordinatensystem

SVG erlaubt das Einbinden von drei Arten von Objekten: Vektorgrafiken, Rasterbildern und Text.

Das Format benutzt zum Rendern[7] das „Maler Model“ (Painter’s Model), bei dem zur Darstellung der Grafiken ein Objekt über das andere gelegt wird. Sollten die Objekte sich teilweise oder komplett überdecken, so liegt das Grafikobjekt, das zuletzt im Quelltext steht, „auf der obersten Ebene“ und überdeckt die darunter liegenden Objekte. Ob das ganz oben liegende Objekt die darunter liegenden komplett verdeckt, oder ob aufgrund von Transparenz des oberen Objekts die unten liegenden Objekte „durchschimmern“, ist durch die Angabe von Opazitätswerten[16] festgelegt. Die Reihenfolge, in der die Grafikobjekte in der SVG Datei beschrieben werden, ist also von Bedeutung. [Umgeh02]

Zur Positionierung der Objekte wird ein Koordinatensystem verwendet, das in der linken oberen Ecke seinen Ursprung hat; nach unten wird die positive y-Achse abgetragen, nach rechts die positive x-Achse. Mittels des `transform` Attributs kann auch eine andere Position des Nullpunkts festgelegt werden. Zu Anfang des SVG Blocks wird mittels `width` und `heigt` ein rechteckiger Bildschirmausschnitt (Viewport) angegeben, auf dem die Grafikobjekte dargestellt werden. Die Maßeinheiten können beliebig gewählt werden (px, inch, mm, cm, …). [Spez03]

[16] Opazität: Begriff für die Lichtundurchlässigkeit von Material. Opazität ist also der Gegensatz zu Transparenz, was Lichtdurchlässigkeit bedeutet. [NetLex03]

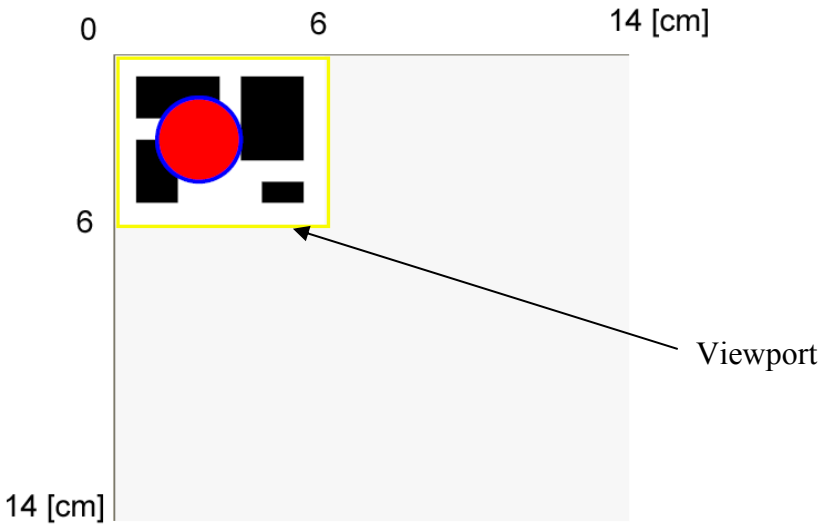

Abbildung 10: SVG Objekte werden im Viewport angezeigt (gelber Rahmen).

4.5 Grafische Möglichkeiten

4.5.1 Basisformen

SVG stellt mehrere geometrische „Basisformen“ zur Verfügung, mit denen man einfach und schnell Grafiken erzeugen kann: Rechteck (optional mit abgerundeten Ecken), Kreis, Ellispse, Linie, Ploylinie und Polygon.

Für alle Formen gilt, dass man sie mit einer beliebigen Farbe füllen kann, bzw. die Stärke und Farbe der Linien ändern kann.

Der Quellcode zur Erzeugung dieser Formen ist einfach und intuitiv zu lesen. Dazu einige Beispiele. Stellvertretend ist der Code für das Rechteck ausführlich erklärt.

Linie („line“-Element):

```
<line x1=“[x1-Koordinate]" y1=“[y1-Koordinate]"
x2="[x2-Koordinate]" y2="[y2-Koordinate]" />
```

Rechtecke („rect“-Element):

```
<rect x="1cm" y="2cm" width="7cm" height="3cm" rx="50"
fill="yellow" stroke="navy" stroke-width="10" />
```

Abbildung 11: Rechteck mit abgerundeten Ecken

Durch Angabe der x/y-Koordinaten des oberen linken Ecks (`x="1cm" y="2cm"`) und der Länge und Breite (`width="7cm" height="3cm"`) wird ein Rechteck definiert. Über `rx="50"` entstehen abgerundete Ecken. Mittels `fill="yellow"` wird das

Rechteck gelb gefüllt und mit `stroke="navy"` und `stroke-width="10"` bekommt es einen blauen, 10 Pixel breiten Rand.

Kreis („circle“-Element):

```
<circle cx="[Mittelpunkt x-Koordinate]" cy="[y-Koordinate Mittelpunkt]" r="[Radius]"/>
```

Ellipse („ellipse“-Element):

```
<ellipse cx=“[x-Koordinate Zentrum]" cy=“[y-Koordinate Zentrum]" rx="[x-Radius]" ry="[y-Radius]" />
```

Polylinie („polyline“-Element):

```
<polyline points="[Punktliste mit x- und y-Koordinaten]" />
```

Erzeugt mehrere zusammenhängende Linien.

Ploygon („polygone“-Element):

Identisch zur Polylinie nur, dass der Anfangs- und Endpunkt miteinander verbunden werden.

Abbildung 12: links Polylinie, rechts Polygon [Spez03]

4.5.2 Pfadelemente

Das vielseitigste Werkzeug von SVG sind Pfadelemente. Mit ihnen kann man beliebige Formen erzeugen. Mittels Pfadelementen lassen sich Punkte, Linien, Bézierkurven und elliptische Formen erzeugen. Pfadelemente müssen nicht zusammenhängen, können also „Löcher“ enthalten. Hier wird nur kurz auf die Möglichkeiten der Pfadelemente eingegangen werde. Anhand von zwei Beispielen für Bezierkurven soll jedoch die „Mächtigkeit“ dieses Werkzeugs deutlicht gemacht werden.

Abbildung 13: Pfadelemente können mit Kontrollpunkten versehen werden. Im Beispiel kann dadurch die Bezierkurve verzerrt werden (rechts). [Spez03]

4.5.3 Farbverläufe und Füllmuster

Farbverläufe beschreiben den weichen Übergang von einer Farbe in eine andere entlang eines Vektors. Man unterscheidet zwischen linearen und radialen Farbverläufen.

Füllmuster können bei SVG beliebige Objekte ausfüllen. Das Muster muss dabei natürlich „replizierbar“ sein, so dass es zum Füllen des Objekts beliebig oft nebeneinander gelegt werden kann.

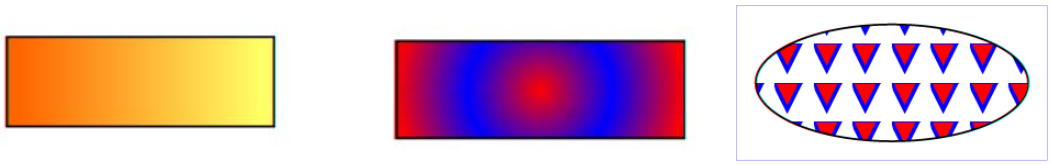

Abbildung 14: links: linearer Farbverlauf, Mitte: radialer Farbverlauf, rechts: Ellipse mit Füllmuster [Spez03]

4.5.4 Filter

SVG bietet die Möglichkeit auf Objekte Filtereffekte, wie z.B. Lichteffekte, Schärfe/Unschärfefilter, Überblenden, Fluten, Morphing[17], Turbulenz u.v.m., anzuwenden. Dabei wird ein oder mehrere Filter auf die ursprüngliche Garfik („original source graphic“) angewendet wodurch eine neue Grafik entsteht.

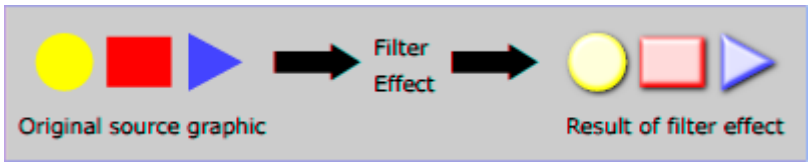

Abbildung 15: Anwendung von Filter [Spez03]

Dies kann man anhand eines Beispiels am eindrucksvollsten sehen:

Abbildung 16: Zahlreiche Filter wurden auf den Schriftzug „SVG“ angewandt. [Spez03]

[17] siehe Abschnitt 3.3.3 Form-Tweening im Flash Teil.

Folgendermaßen ist die Grafik entstanden:

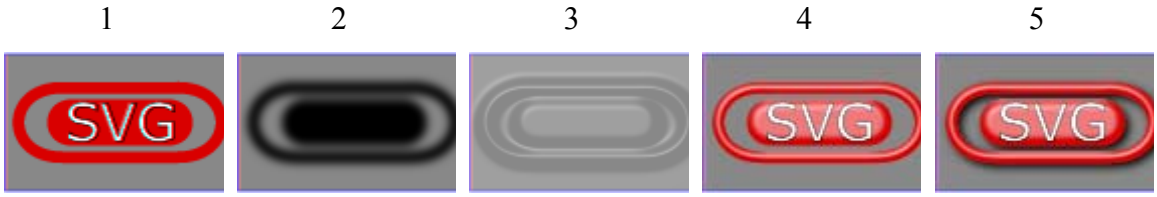

Abbildung 17: Mehrere Filter werden nacheinander auf die „Ursprungsgrafik“ angewandt (vereinfachtes Schema, einige Schritte wurden weggelassen). [Spez03]

Bild 1 ist die „Ursprungsgrafik“. Darauf wird zunächst ein Weichzeichner („feGaussianBlur“) angwendet (Bild 2). Mittels eines Farbfilters („feSpecularLighting“) wird Bild 3 erreicht. In Bild 4 wird der Farbfiltereffekt mit der „Ursprungsgrafik“ kombiniert („feComposite“). Die Endgrafik (Bild 5) entsteht schließlich durch Überlagerung („feMerge“) der zwei Schichten von Bild 2 und 4.

4.5.5 Animationen

Animationen lassen sich in SVG zwei Gruppen unterteilen: Bewegungen und Attributänderungen. So lassen sich z.B. Pfadanimationen („animateMotion“), Farbänderungen („animateColor“), Transformationen wie z.B. Größenänderungen, Drehungen usw. („animateTransform“), Fading (ein/ausblenden) u.v.m. realisieren. Für diese Effekte bedient sich SVG teilweise SMIL (Synchronized Multimedia Language)[18], falls bestimmte Animationen nicht in SVG vorgesehen sind. Eine weitere Möglichkeit Animationen zu erzeugen ist SVG- DOM verbunden mit einer Scriptsprache zu verwenden.

Im folgenden Beispiel ist eine Animation entlang eines Pfades verwirklicht. Außerdem wird eine Skalierungs- (Größenänderung) und Opazitätsanimation (Fading) durchgeführt.

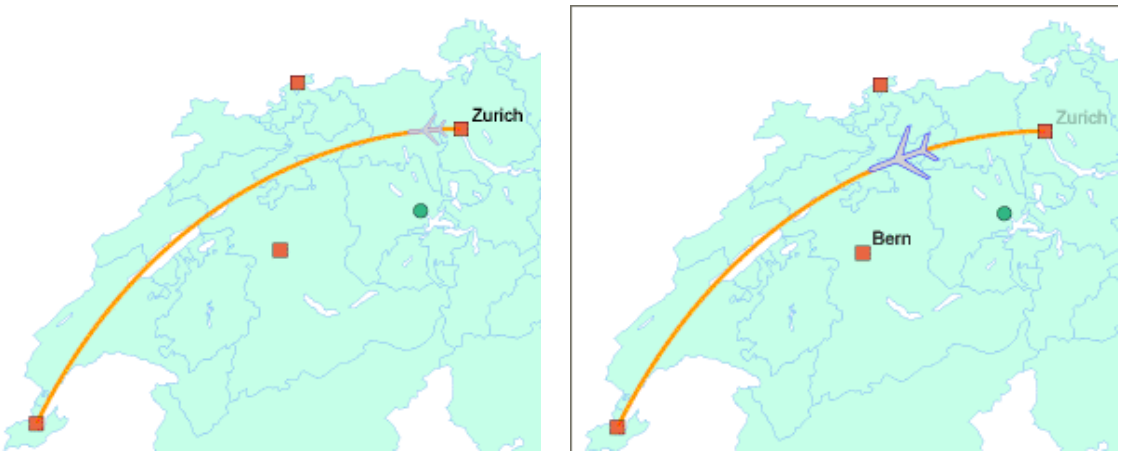

Abbildung 18: links: Das Flugzeug startet und wird langsam größer. Rechts: Das Flugzeug hat seine volle Größe erreicht und fliegt entlang der orangenen Route. Die

[18] SMIL spezifiziert Multimedia-Elemente und Animationen. [NeuWin01]

Städte, die es passiert werden zuerst ein- und dann ausgeblendet (Zürich, Bern). [NeuWin01]

4.5.6 Interaktivität

Der Benutzer kann mittels Eingabegeräte wie z.B. der Maus direkt mit einer SVG Grafik interagieren. Er kann die Visualisierung der Grafik in Echtzeit beeinflussen (z.B. wenn mit dem Mauszeiger über markante Punkte eines Stadtplans fährt), Animationen starten, Skripte ausführen (z.B. durch drücken eines Knopfs) oder Hyperlinks aktivieren. Auch das Aussehen des Mauszeigers kann beeinflusst werden. Schließlich kann der Betrachter in die Grafik hinein- und herauszoomen oder die gesamte Grafik verschieben.

Kommunikation mit einem Server, um z.B. Datenbankinformation abrufen zu können, kann mittels der Einbindung von PHP, Perl oder ASP realisiert werden. Mit Clientseitige Scriptsprachen wie z.B. JavaScript oder VBScript ist es möglich DOM Elemente anzusprechen und Aktionen auszuführen, falls ein bestimmtes Ereignis („event“[19]) eingetreten ist.

Ein einfaches Beispiel zeigt, wie man mit den Ereignissen „onmouseover“, „mouseout“, „onmouseup“, „mousedown“, „onload“ und „onclick“ und etwas JavaScript Farbwechsel interaktiv hervorrufen kann.

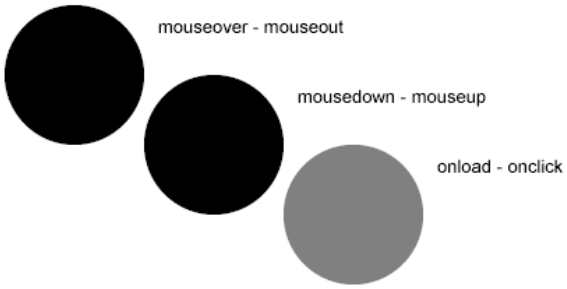

Abbildung 19: Beim „Überfahren“ (links), anklicken (Mitte) bzw. loslassen der Maustaste (rechts) ändert sich die Farbe von schwarz auf grau. [Umgeh02]

Ein sehr beeindruckendes interaktives Beispiel ist SVG Draw[20], ein online Malprogramm von Adobe, bei dem alle üblichen geometrischen Formen verwendet und deren Attribute verändert werden können.

4.6 Was zur Darstellung von SVG Dateien benötigt wird

Noch die wenigsten Browser unterstützen „von Haus aus“ die Anzeige von SVG Dateien. Mozilla testet in ihrem aktuellem Browser 1.5 eine SVG Unterstützung[21], die jedoch nicht die gesamte SVG Spezifikation unterstützt. Fast vollständige

[19] Eine komplette Liste der möglichen „events“ findet man in der SVG 1.1 Spezifikation unter „16.2. Complete list of supported events” auf http://www.w3.org/TR/SVG11/interact.html.
[20] Zu finden ist es in der Version 0.9 unter http://www.adobe.com/svg/demos/svgDraw/svgDraw.
[21] In den Standardeinstellungen ist die SVG Unterstützung nicht aktiviert. Mehr dazu unter man unter http://www.mozilla.org/projects/svg.

Unterstützung der Spezifikation bietet hingegen Adobes SVG Viewer[22] für Windows-, Linux- und Mac-Systeme, das als Browser Plug-in für den Internet-Explorer, den Netscape-Navigator, Opera u.a. verfügbar ist. [W3C03]

Der SVG Viewer unterstützt Anti-Aliasing[6]. Mit ihm lassen sich SVG Grafiken zoomen und verschieben oder Animationen anhalten. Ferner kann man nach Zeichensätzen suchen und den Quellcode betrachten oder speichern.

4.7 Was man zur Erstellung von SVG Dateien benötigt

Zur Erstellung und Bearbeitung von SVG Grafiken werden Autorenumgebungen benötigt. Obwohl SVG, da es textbasiert ist, in einem einfachen Texteditor geschrieben werden kann, ist es natürlich viel komfortabler eine Benutzeroberfläche zur Erstellung von Grafiken und Animationen zu benutzen. Diese WYSIWYG[23] Editoren bieten meist umfassende Möglichkeiten um anspruchsvolle Farbverläufe, Füllmuster, Filtereffekt, animierten Text usw, schnell und einfach zu erstellen. Einer der „mächtigsten" Autorenumgebungen ist zurzeit JASC WebDraw 1.0.

5 Vergleich von Flash und SVG

Im Gegensatz zu Flash stellt SVG ein offizieller Standard des W3C dar. Er ist XML basiert, wodurch es schnell erlernbar ist und sich andere Techniken wie DTD oder XSL übertragen lassen. SVG-Dateien lassen sich ohne weitere (kostenpflichtige) Programme in einem Texteditor programmieren. Dadurch ist auch ihr Quellcode für jeden einsehbar. Flash hingegen ist ein binäres Format, das nur für SWF-Dateien spezifiziert ist und sich nur mittels Autorenumgebungen erzeugen lässt. Dadurch ist es auch nicht durch Suchmaschinen indizierbar. SVG Dateien lassen sich direkt bearbeiten, da man ihren Quellcode einsehen kann. Flash-Filme (SWF-Format) kann man nicht bearbeiten; man benötigt dazu die ursprüngliche FLA-Datei.

Die Erzeugung von SVG-Dateien in einem Texteditor oder einer Autorenumgebung ist (bis jetzt) noch relativ kompliziert. Flash (z.B. Flash MX 2003) hingegen bietet die weitaus komfortabelste und benutzerfreundlichste Oberfläche. Mit ihr lassen sich komplizierte Effekte und Animationen spielend leicht erstellen.

Im multimedialen Bereich unterstützt Flash die Einbindung von Sounds sowie MP3-Kompression und Streaming. Weiterhin erlaubt Flash die Integration von Movie-Formaten wie externen Quicktime-Movies. SVG hingegen muss bei diesen Elementen auf SMIL zurückgreifen.

Zur Interaktion mit dem Benutzer verwendet Flash die eigene Programmiersprache ActionScript, SVG ist unabhängiger und erlaubt die Einbindung von sämtlichen Scriptsprachen wie z.B. JavaScript.

Bei der Verbreitung der Plug-ins bzw. Standaloneplayern hat Flash die Nase vorn. Bereits 97% der Internetuser haben die Möglichkeit Flash-Filme anzuschauen. SVG-Dateien hingegen können nur 10% der „Surfer" betrachten. [Umgeh02]

---

[22] Die aktuellste Version von „Adobe SVG Viewer 3.0" gibt es unter http://www.adobe.com/svg.
[23] „What You See Is What You Get" – "Was du siehst, ist was du bekommst"

6 Ausblick

Nachdem erst vor kurzem die Spezifikation 1.1 von SVG veröffentlicht wurde, wird bereits an der Version 1.2 gearbeitet. Für SVG 2.0 werden schon erste Bedarfanalysen erstellt. Aufgrund unterschiedlichster Anforderungen wurde die Spezifikation aufgespaltet. Für mobile Endgeräte wie PDAs oder Handys steht SVG-Mobile bereit. Für die Druckindustrie oder auch für den Heimanwender wird gerade SVG Print, ein für Ausdrucke optimiertes SVG Format, entwickelt. Die W3C Arbeitsgruppe ist vor allem auch bestrebt SVG als Standard-Applikation für die Darstellung von Vektorgrafiken in die gängigen Web-Browsern zu integrieren, so dass eine problemlose Darstellung von SVG-Dateien beim Heimanwender gesichert ist. Als weiterer Schritt ist dann die Etablierung von SVG als Austauschformat für vektorbasierte Bildformate geplant. [NeuWin01]

Macromedia hingegen versucht das Tempo mit Hilfe seiner MX Produktfamilie zu verschärfen. Mit diesem Paket deckt Macromedia sämtliche Produkte ab, um den von ihnen propagierten „Rich Media/Internet Content“ abzudecken. Mit diesem Programmpacket lassen sich Bilder bearbeiten (Fireworks), Vektorgrafiken erstellen (FreeHand), Webseiten bauen (Dreamweaver), interaktive Animationen erzeugen (Flash) und Datenbanken einbinden (ColdFusion). Mit diesen benutzerfreundlichen Programmen lassen sich schnell und komfortabel auf jedem Einsatzgebiet erstklassige Ergebnisse erzielen. Macromedia versucht also durch eine Lösung aus einer Hand eine Vielzahl von Aufgabenbereichen abzudecken, um so (auch) ihre Marktführerschaft im Bereich der vekorbasierten Grafikformate zu sichern. [Macro03]

7 Zusammenfassung

Flash hat mit seinem Format ganz klar die „Marktführerschaft“ inne. Jedoch hat es auch einiges an Zeitvorsprung im Vergleich zu SVG. SVG wurde erst 2001 von W3C spezifiziert, begann dann aber eine schnelle Aufholjagd. Aufgrund der Tatsache, dass es XML basiert ist, und da es mit seiner umfassenden Spezifikation sehr weitreichende Anwendungsgebiete eröffnet, wird es seine Anhänger vor allem bei den professionellen Anwendern finden, die lange Zeit auf ein geeignetes (offenes) Vektorformat gewartet haben, das ihnen ein „mächtiges“ Werkzeug an die Hand gibt, das ihren Anforderungen gerecht wird. Solche Anforderungen könnten z.B. in der technischen Illustration, wissenschaftlichen Datenvisualisierung oder allgemein in der Druckindustrie liegen. Diese Aufgabenfelder hat Flash bis jetzt gar nicht erschlossen. Deshalb glaube ich, dass Flash seine „Monopolstellung“ im Internet erhalten kann, da Flash und SVG auf unterschiedliche Anwender und Anwendungsgebiete ausgelegt sind. SVG hingegen erschließt ganz neue Gebiete, die es bis jetzt noch gar nicht auf diese Weise erschlossen werden konnten.

Literatur

[Blitz01] „ja.na“ Rogge, Pixel und Vektoren, Blitzmerker, 2001
http://www.blitzmerker.i-d.de/pdf/referate/pixel_huehne.pdf

[Carto01] carto:net. Geogrphic geogrphic papers and puplications, 2001
http://www.carto.net/papers/svg/path_animation_e.html

[Fünder03] Guido Fünders. Macromedia Flash, Rich Internet Applications, RWTH-Aachen, 2003. http://www.rz.rwth-aachen.de/mata/downloads/seminar_dv/2003_04/Flash_RIA.pdf

[Gal03] Galileo Computing, Glossar: Pixel http://www.galileocomputing.de/glossar/gp/anzeige-8745

[Hilfe02] Flash-Hilfe zu Flash MX 6.0, 2002

[Kerman02] Phillip Kerman. In 21 Tagen Flash MX, Markt + Technik Verlag, 2002

[Macro03] Die offizielle Homepage von Macromedia http://www.macromedia.com/de

[NetLex03] Akademie.de, Net-Lexikon http://netlexikon.akademie.de/DTD.html

[NeuWin01] Andreas Neumann, Andréas M. Winter. SVG - Scalable Vector Graphics, Ein zukünftiger Eckstein der WWW-Infrastruktur, ETH Zürich, 2001 http://www.carto.net/papers/svg/articles/paper_ugra_zurich_2001.pdf

[Player03] Flash Player 7, 2003 http://www.macromedia.com/software/flashplayer

[Stark01] Benjamin Stark. Flash Weather, Vektorisierung von raum- und zeitbezogenen Daten zur Visualisierung mit Macromedia Flash, Universität Osnabrück, 2001 http://www-lehre.informatik.uni-osnabrueck.de/~fbstark/diplom/arbeit/html/node6.html

[Umgeh02] Karl Umgeher. SVG - Scalable Vector Graphics, Fachhochschule St. Pölten, 2002 http://www.t0.or.at/~mkoerner/diploma/da_umgeher.pdf

[W3C03] W3C. Infoseite zu SVG http://www.w3.org/Graphics/SVG

[Spez03] W3C. Spezifikation von SVG 1.1 http://www.w3.org/TR/SVG11