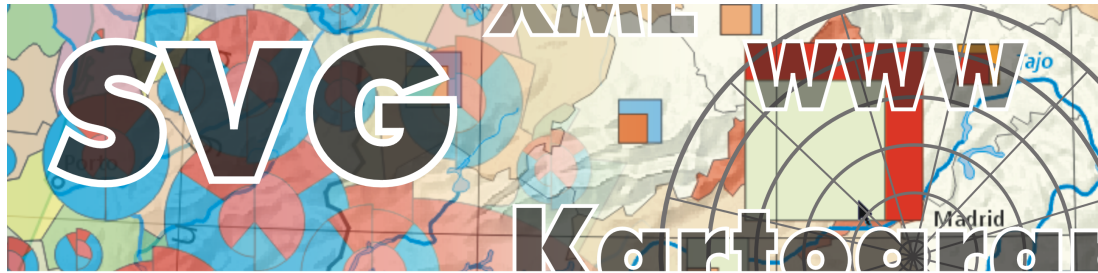


## Hochqualitative Kartographie im Internet auf Vektorbasis – mit Hilfe von SVG nun möglich.

Yes raster is faster, but raster is vaster, and vector just seems more correcter.

[[Tomlin, 1990](#)]



*Neumann Andreas, Institut für Kartographie, ETH Zürich*

([neumann@karto.baug.ethz.ch](mailto:neumann@karto.baug.ethz.ch)) und

*Winter Andréas, Institut für Geographie und Regionalforschung, Universität Wien*

([andre.mw@gmx.net](mailto:andre.mw@gmx.net)).

Das Internet ist ein verteiltes Massenphänomen. Es stellt andere Kommunikations- oder Informationsmedien bezüglich Nutzungsmöglichkeiten, Wachstumsraten und Durchdringungsgrad in den Hintergrund. Auch für die Kartographie waren die verschiedenen Dienste des Internets, allen voran das WWW, stets interessante Verteil- und Publikationsmedien. Lange Zeit jedoch war qualitativ hochstehende Online-Kartographie, aufgrund fehlender standardisierter Vektor-Graphik-Formate stark beeinträchtigt.

Mit Vektoren können wir die Interaktionen und Analysen, sowie die bildschirmbedingten Funktionen des Zooming (Lupe) und Panning (Translation des Ausschnittes) wesentlich einfacher integrieren und die Qualität der graphischen Wiedergabe um ein vielfaches steigern. Bis vor kurzem lief nur die Darstellung von Text und Rasterbildern standardisiert ab. Für die Kartographie notwendige Funktionserweiterungen waren nur mit Hilfe von serverseitigen Scripten oder komplizierten Javascript-Konstrukten und dem redundanten Laden von Rastergraphiken möglich.

Nun gibt es seit Ende 1999 einen neuen, vielversprechenden Standard namens SVG (Scalable Vector Graphics), der die Möglichkeit bietet, WWW-Seiten mit Vektorgraphiken zu versehen. Die Entwicklung seitens der betroffenen Industrien spricht für ein baldiges flächenhaftes Einsetzen dieser Technologie, die es den Kartographen erstmals erlauben wird, auch im Internet höchste graphische Ansprüche zu befriedigen.

Dieser Artikel beschäftigt sich mit der Problematik von Standards im Internet, die den späten Zeitpunkt für das Einsetzen von Vektorgraphik erklären. Es folgt eine Besprechung der zum Teil seit 1996 verfügbaren anderen Möglichkeiten, Vektordarstellungen im Internet zu verwenden. Nach einer Erläuterung der nötigen Voraussetzungen für Vektorgraphik und der wichtigsten komplementären Techniken wird der neue Standard SVG vorgestellt, dessen Eigenschaften kartographiebezogen und im Einzelnen besprochen. Beispiele dazu sind an den jeweils ausgewiesenen Internetseiten mit kommentiertem Quellcode verfügbar.

## **1. Standards im Internet**

Gerade das Internet hat sehr zur Öffnung von Fileformaten, Schnittstellen, Protokollen, Architekturen, ja sogar ganzer Betriebssysteme und Applikationssoftware beigetragen. Diese tragen schliesslich auch den grössten Teil der Internet-Infrastruktur und stellen herkömmliche Strukturen, Hierarchien und Geschäftsmodelle in Frage – dies obwohl einzelne Firmen immer wieder versuchen diese zu unterwandern oder für sich vereinzunehmen. Es ist zu wiederholen, dass im Internet der Begriff "Standard" erst angewandt werden kann, wenn er von einer Mehrheit der Firmen akzeptiert (das heisst umgesetzt) wird und auch tatsächlich von den Betreibern und Erstellern von Internet-Seiten eingesetzt wird. Es gibt hier mehrere Entscheidungsträger und Entscheidungsmechanismen:

### **1.1 De-Jure Standards**

Standards sind dokumentierte Vereinbarungen, die technische Spezifikationen oder

andere exakte Merkmale wie Regeln, Richtlinien oder Definitionen von Eigenschaften liefern um sicherzugehen, dass Materialien, Produkte, Prozesse und Dienstleistungen ihrem Zweck gerecht werden [ISO, 1999]. Diese allgemeine Definition gilt nur im Rahmen der De-Jure-Standards. Sie werden von einem Normierungsinstitut, beispielsweise ISO (International Organization for Standardization) oder IEEE (Institute of Electrical and Electronical Engineers) verabschiedet und sind offiziell gültig. ECMA (European Computer Manufacturer's Association) oder W3C (World Wide Web Consortium) bilden ebenfalls Normen die öffentlich angekündigt werden und in deren Erstellung die Öffentlichkeit miteingebunden wird, allerdings ohne dass diese in nationale Normenwerke übernommen werden. Grundsätzlich geht es im hier besprochenen Bereich neben der Zweckmässigkeit auch um die Harmonisierung der Seitenbeschreibungssprachen im WWW. Es gibt nicht nur HTML zur Beschreibung von Text, sondern auch andere Sprachelemente (wie Formatierung, Animation, Interaktivität, etc.). Die ineinandergreifenden Elemente müssen "kommunizieren" können, und dies ist nur bei einer klar definierten Syntax gewährleistet.

Man sollte sich nicht beirren lassen, diese meist hochwertigen, gut dokumentierten Standards werden in der Regel nicht vollständig eingehalten. Anfangs hinkten die offiziellen Standards der Entwicklung hinterher, so war zum Beispiel HTML 1.0 zum Zeitpunkt seiner Publikation bereits längst umgesetzt und technisch überholt. Zur Zeit ist es aber durchaus so, dass Standards geschaffen werden, die technisch noch gar nicht einsetzbar sind. Diese Standards werden von mehr oder weniger unabhängigen Konsortien oder Arbeitsgruppen ausgearbeitet.

## **1.2 De-Facto Standard**

Standard, der zwar nicht durch eine anerkannte Standardisierungsorganisation, offiziell verabschiedet wurde, der aber durch seine Verbreitung allgemein akzeptiert ist. Es handelt sich hier also nicht um eine Norm, sondern um einen Umstand. Software-Firmen können aus technischer Sicht nicht alle De-jure-Standards einhalten, andererseits möchten sie aus Konkurrenzgründen firmeneigene Standards aufstellen, mit dem Ziel sie zum De-facto-Standard zu erheben.

Eine negative Folge dieser Umstände ist eine Verwässerung der De-jure-Standards. Das heisst konkret, dass Teile von verabschiedeten Paketen (wie zum Beispiel HTML 4.0) nicht umgesetzt werden. Hinzu kommen firmeneigene Standards die unter Umständen positive Eigenschaften aufweisen, aber in Kompatibilitätsproblemen mit Konkurrenzprodukten oder den Umgebungsvariablen münden. Will man unter diesen Umständen sicher gehen, dass alle Anwender (das Internet ist heterogen bestückt), die nach gewissen Standards geschriebene Information verstehen, so muss man auf einen niedrigeren, älteren Standard zurückgreifen. Innerhalb dieses Dilemmas befindet sich jeder, der ein Internetprojekt aufbaut, ganz besonders wenn es darum geht Inhalte aufzubereiten die sich a priori nicht dafür eignen: das ist zum Teil der Fall bei Graphik, oder auch bei Geodaten.

Es müssen aber nicht immer grosse Firmen diese Standards etablieren. Auch eine Forschergruppe kann eine Idee umsetzen und sie zur Verfügung stellen, die Verbreitung findet alleine durch die Qualität der Idee statt. Beispiele sind hier das CERN (European Organization for Nuclear Research), das überhaupt für den ersten

graphischen Browser verantwortlich zeichnet. Ein neueres Beispiel wäre das Fraunhofer Institut mit dem MP3–Audio–Kompressionsverfahren. Ähnlich zu den De–Jure–Standards können, bei geklärter rechtlicher Lage, diese Ideen im kommerziellen Bereich aufgefangen, verfeinert und vermarktet werden. Dieser Beitrag von kleineren Gruppen wirkt durchaus förderlich, da er nicht dem Mainstream von Firmen und Konsortien folgt.

## 2. Vektor–Formate

In diesem Rahmen also muss ein Vektor–Standard etabliert werden. Es war von Anbeginn klar, dass er notwendig ist, und sehr früh kam es zu ersten Versuchen Vektordaten im WWW darzustellen. Neben dem schon klassischen Konkurrenzkampf der Internet–Industrie gesellen sich hier die Software–Giganten der Graphik–Industrie hinzu, was eine Einigung noch schwieriger macht. Ebenfalls erschwerend ist der Umstand, dass es in der Graphik–Welt kein zuverlässiges Vektoraustauschformat gibt. Bei Rastergraphik konnte man sich schneller einigen. Die Folge sind eigenständige Entwicklungen unterschiedlichster Art, die man nicht ohne Relativierung nebeneinanderstellen kann. Hier eine Auswahl von Ansätzen, die prinzipiell kartographisch nutzbar wären (viele davon sind mittlerweile technisch überholt und werden nicht mehr weiterentwickelt):

### 2.1 Bestehende Ansätze zu Vektorformaten im WWW

**SVF:** Simple Vector Format: Ursprünglich für das Anzeigen von CAD–Zeichnungen im Internet gedacht. Ein Plugin, das 1996 die ersten Ansätze zur Darstellung von Vektoren im WWW machte, ab 1997 jedoch keine Weiterentwicklung mehr. Drucken ist nur mit einer kostenpflichtigen Variante möglich. Es bestehen Konverter für HPGL und DXF. Funktionen: gestuftes zoom in, gestuftes zoom out, zoom window, zoom reset, panning, layer on/off. Die Interaktivität ist relativ beschränkt, Linien und Flächen können mit einem Extra–Programm mit Hyperlinks versehen werden. Kein Anti–Aliasing in der Darstellung, keine Animation. ([www.softsource.com/svf](http://www.softsource.com/svf))

**DWF:** Drawing Web Format, visualisierbar mit dem Plugin WHIP4 von AutoDesk ist eines der wenigen Plugins das über JavaScript ansprechbar ist. Erstellung durch Konvertierung von DWG–Dateien. Funktionen: Pan, stufenloses zoom in/out, zoom Window, zoom reset, zoom auf vordefinierte Ansichten (Named Views), Layer ein/aus, Anzeigen von Hyperlinks (nicht zufriedenstellend in der Gesamtansicht). Einbinden von Bildern und Text möglich. Keine Animation, und dadurch ist die Ansprechbarkeit durch JavaScript nur bedingt nützlich. Es gibt davon auch eine Applet–Version. ([www.autodesk.com/whip](http://www.autodesk.com/whip))

**Flash:** das zur Zeit am weitesten verbreitete Vektor–Format mit Interaktion, läuft als Plugin von Macromedia im Browser. Es ist aus kartographischer Sicht der aktuelle Standard für Vektordarstellung. Es entspricht keinem De–Jure–Standard, war lange schlecht dokumentiert und widerspricht als proprietäres, binäres Format der Open–Source–Idee. Da das Format aber auf die Werbe– und Multimediabranche massgeschneidert war, konnte es sich durchsetzen und nahezu 70% der Browser unterstützten Anfang 2000 das Format ohne Zusatzinstallation [[StatMarket, o.J.](#)].

Flash kann allerdings mehr als nur Vektorgraphik anzeigen: Filme sowie Audio lassen sich ohne grosse Schwierigkeiten integrieren bzw. generieren. Graphik kann entweder aus den Programmen Macromedia Freehand, Adobe Illustrator oder auch CorelDRAW exportiert werden und/oder mit der Editier-Software Macromedia Flash bearbeitet werden. Es steht eine breite Palette an Funktionen zur Animation und Einbindung graphischer Effekte zur Verfügung. Viele Anwendungen des Formats zeigen, dass es sinnvoll im Internet einsetzbar ist und eingesetzt wird, da es u.a. auch das progressive Übertragen der Daten ermöglicht (Seitenaufbau ohne dass die gesamte Datei geladen sein muss). Problematischer ist für Kartographen das Einbinden externer Daten und das Nachbearbeiten oder Kreieren eigener, notwendiger Funktionen. Die Schnittstellen werden weiter ausgebaut, sind aber in keiner Weise standardisiert. ([www.macromedia.com/software/flash](http://www.macromedia.com/software/flash))

**PGML:** Precision Graphics Markup Language. Proprietäre, textbasierte 2D-Vektorformatspezifikationen vorgeschlagen von Adobe auf der Basis von PostScript und PDF. Generierung durch Konvertierung aus PDF. Hohe Ansprüche und hohes Interaktivitätsniveau, jedoch keine Umsetzung. ([www.w3.org/TR/1998/NOTE-PGML](http://www.w3.org/TR/1998/NOTE-PGML))

**WebCGM:** ein Versuch den ISO-Standard Computer Graphic Metafile (CGM) WWW-fähig zu machen. Die Version für das WWW ist binär und eine W3C-Empfehlung, also als ausgereifter Standard zu betrachten. WebCGM erlaubt Interaktionen mit reichhaltiger Link-Syntax, ist aber primär der Visualisierung technischer und wissenschaftlicher Zeichnungen gewidmet [[cgmOPEN.org](http://cgmOPEN.org), 1999]. ([www.w3.org/Graphics/WebCGM](http://www.w3.org/Graphics/WebCGM))

**VML:** Vector Markup Language. Ein proprietäres, textbasiertes 2D-Vektor-Format von Microsoft. Wurde für den Microsoft InternetExplorer umgesetzt, aber seit September 1998 nicht mehr weiterentwickelt. Ansätze einer interessanten Syntax, jedoch ohne sinnvolle Erweiterungen. Vor SVG der erste vielversprechende Ansatz, jedoch nur auf eine Plattform beschränkt. ([www.w3.org/TR/NOTE-VML](http://www.w3.org/TR/NOTE-VML))

**PDF:** Portable Document Format. EPS-Syntax. Mittlerweile zum nativen Format von Adobe Illustrator 9 geworden. Ersetzt in der Druckvorstufe mehr und mehr Postscript und EPS. Nicht primär für das WWW gedacht, aber über Plugins einsetzbar. Es existieren Exportfilter aus den meisten Textverarbeitungs-, Graphik- und DTP-Programmen. Der Viewer funktioniert auch unabhängig von einem Browser und wird von Adobe frei zur Verfügung gestellt. Die Dokumente können über Passwörter lese-, druck- oder kopiergeschützt werden. Das binäre Format verbietet ein Erfassen durch WWW-Suchmaschinen, die Text-Variante erlaubt dies, ist aber in bezug auf das Datenvolumen nicht besonders WWW-freundlich. ([partners.adobe.com/asn/developer/acrosdk/DOCS/pdfspec.pdf](http://partners.adobe.com/asn/developer/acrosdk/DOCS/pdfspec.pdf))

**SVG:** Scalable Vector Graphics, ist ein neuer, vollständig offener Standard, der vom World Wide Web Consortium (W3C) empfohlen und entwickelt wird und in der Entwicklung zahlreiche wichtige Softwarekonzerne und Forschungsanstalten miteinschliesst. Der SVG-Standard bietet alle Vorteile des derzeitigen De-Facto-Standards Flash (siehe weiter [oben](#)), und stellt folgende zusätzliche Eigenschaften zur Verfügung: eingebettete Schriften, Extensible Markup Language (XML), Stylesheets (CSS), Interaktivität und Animation; das Ganze bei voller Einbindung in

die vorhandene HTML–Struktur über das [DOM](#). Mehr dazu im [Hauptteil](#) dieses Artikels. ([www.w3.org/Graphics/SVG/Overview.htm](http://www.w3.org/Graphics/SVG/Overview.htm))

**VRML:** Virtual Reality Modeling Language, dient der 3D–Darstellung, grundsätzlich ist es aber in 2D einsetzbar. Bietet viele nötige Voraussetzungen, stellt aber einen Overkill dar, wenn es um einfache Darstellungen geht; des weiteren befindet sich VRML zur Zeit in einer Phase der Umstrukturierung. Plugin nötig. ([www.web3d.org](http://www.web3d.org))

**HGML:** Hyper Graphics Markup Language, Juni 1998, ist als kompaktes Format gedacht, spezialisiert auf geringe Bandbreiten, keine Umsetzung. ([www.w3.org/TR/NOTE–HGML](http://www.w3.org/TR/NOTE–HGML))

**DrawML:** Dezember 1998, Format für das einfache Abbilden und Aktualisieren von technischen Zeichnungen. ([www.w3.org/TR/1998/NOTE–drawml–19981203](http://www.w3.org/TR/1998/NOTE–drawml–19981203))

## 2.2 2D–API's im Zusammenhang mit Applets oder ActiveX–Objekten

**ActiveX:** ein Ansatz von Microsoft, über die proprietäre Scripting–Sprache Visual Basic, Objekte mit Microsoft–spezifischer Kodierung anzusprechen. Keine sinnvolle Anwendbarkeit, da Browser– und Betriebssystemabhängig, sowie kaum dokumentiert. Erstellung mit sog. "Development Kits". Interaktivität möglich. Es handelt sich um kein Plugin im eigentlichen Sinne. ActiveX's sind in andere Dokumente eingebettete und ansprechbare Objekte. ActiveX kann man somit auch in Microsoft Word– und PowerPoint–Dokumenten finden.

**Java2D:** als binäres Programm mit graphischer 2D–Ausgabe in Internetseiten eingebundener Bereich. Frei gestaltbar und sehr flexibel und plattformunabhängig einsetzbar. Läuft in einer "Virtual Machine" auf dem Client ab und belastet diesen unter Umständen relativ stark. Java2D wurde von Sun und Adobe designt um plattformunabhängige, qualitativ hochwertige und einfache Graphikprogrammierung zu erlauben. Im Prinzip gibt es die gleichen graphischen Möglichkeiten wie unter SVG – die meisten SVG Viewer und Plugins sind in Java2 geschrieben – man braucht aber wesentlich mehr Programmier–Know How um einen ähnlichen Grad an Interaktion zu erreichen wie unter SVG. Dafür können Java2 Programmierer von einer Vielzahl hochwertiger Bibliotheken aus den Bereichen Datenbanken, User–Interface, 2D–Grafik, 3D–Graphik, Multimedia, Netzwerk und Telephonie profitieren. Java2–Applets können SVG Plugins steuern und sind durchaus sinnvoll mit diesen kombinierbar. Ohne grossen Aufwand können aus den gleichen Quellen auch stand–alone Programme erzeugt werden.

## 3. Document Object Model (DOM)

Die meisten der oben erwähnten Vektorformate für das WWW werden in der Regel mittels Plugin im Browser dargestellt, das heisst, es ist ein zusätzliches Programm im Browser notwendig, das nicht ursprünglich in der Browsersoftware integriert ist. Des weiteren handelt es sich vielfach um proprietäre, schlecht dokumentierte Datenformate, die nur durch Export aus einem bestimmten Graphikprogramm heraus

generiert werden können und somit dem Internetgrundsatz des Open–Source widersprechen. Weitere Nachteile sind die meist binären Formate, d.h. sie sind ausserhalb der Erstellungssoftware nicht mehr editier– und korrigierbar. Dies ist allerdings bei (fast) jeder sinnvollen Internet–Karte notwendig, da Interaktionen und Anzeigemechanismen "händisch" massgeschneidert und angepasst werden müssen und eine spätere Manipulation entsprechend der Sachdaten, etwaiger Analysen, Selektionen und Symbolisierung vorgenommen werden muss.

Das Schlagwort "Interaktion" bedingt Ansprechbarkeit der einzelnen Elemente einer WWW–Seite, so auch der abzubildenden Vektor–Objekte. Diesen Grundsatz erfüllt keiner der obigen proprietären Ansätze. Um einzelne Objekte anzusprechen bedarf es einer klaren Objekthierarchie, deren oberstes Glied die WWW–Seite ist. Alle in dieser Seite enthaltenen Objekte können mittels einer hierarchischen Struktur angesprochen werden. So können in einer herkömmlichen WWW–Seite zum Beispiel folgende Hierarchien herrschen (vereinfacht):

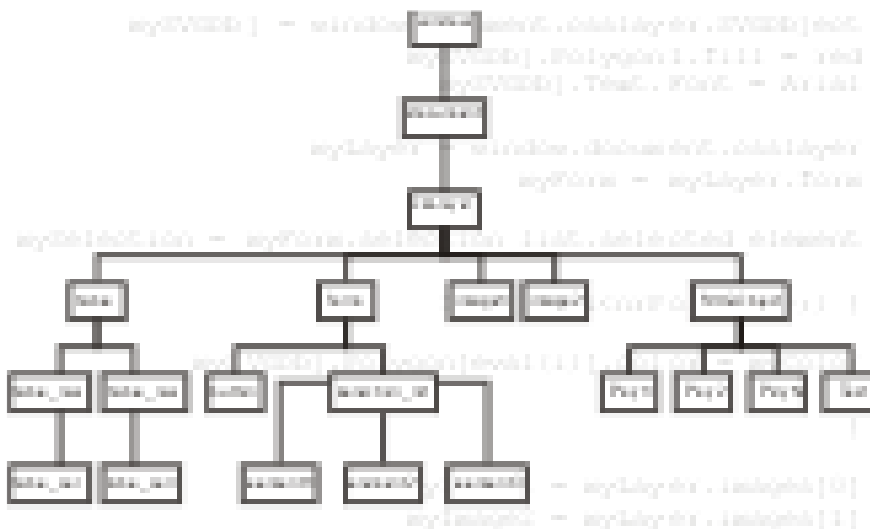


Abbildung 1: Beispiel einer einfachen DOM Objekthierarchie

Diese Objekthierarchie nennt man Document Object Model (DOM), und ist der Grundstein für effizientes Arbeiten mit Elementen einer WWW–Seite wie es auch mit kartographischen Inhalten der Fall ist. Ein neuer Vektor–Standard muss also DOM–konform sein. Im Prinzip lassen sich auch Plugins in das DOM einbinden. Da aber die Hierarchie unterhalb des Plugins meist für das reibungslose Funktionieren innerhalb des Plugins und nicht für die Kommunikation des Plugins mit seiner gesamten Umgebung geschaffen wurde, treten an dieser für die Interaktion sehr wichtigen Stelle immer wieder Probleme auf.

## 4. Extensible Markup Language (XML) – Basis aller künftigen WWW–Dialekte

Ursprünglich war die Idee von HTML (Hypertext Markup Language) Inhalt, Struktur und deren Formatierung für verschiedene Ausgabemedien strikte zu trennen. CSS (Cascading Style Sheets) wären in diesem Fall für das zentrale

Layoutieren von WWW-Seiten-Elementen zuständig, sowie auch für das pixelgenaue Positionieren. Spätestens mit dem "Browserkrieg" zwischen Netscape und Microsoft war dieser Ansatz im Zuge der im "Wildwuchs" eintretenden Einführung eigener, proprietärer Standards zum Scheitern verurteilt. Dazu kam, dass Web-Autoren auch noch diese proprietären Standards verwendeten und Formatierungen im strukturierten Text verwendeten, anstatt die oben vorgeschlagene Trennung vorzunehmen (z.B. fett oder kursiv für Zitat, anstelle Erstellung eines eigenen Formates für "Zitat") – ein Fehler, den heute immer noch sehr viele Anwender beispielsweise im Gebrauch von Textverarbeitungen begehen.

Ansprechbarkeit mittels DOM kann nur mit einer standardisierten Schnittstelle und Sprachsyntax erfolgen. XML, ist der universelle Standard für netzwerktransparente und plattformunabhängige, strukturierte Dokumente für das WWW und darüber hinaus als Austausch zwischen Applikationen aller Art. XML ist erweiterbar und bietet daher die Basis für alle weiteren "Dialekte" (Spezialisierungen), wie z.B. SMIL (Multimedia), SVG (Vektorgraphik), MathML (Mathematische Sonderzeichen und Formatierungen), X3D (3D-Grafik), XHTML (HTML Nachfolger), Xform (Formulargenerierung), GML (Geography Markup Language – erst im Anfangsstadium, siehe OpenGIS Consortium) uvm. Erweiterungen können von Fachverbänden oder Einzelpersonen festgelegt werden, sofern auch die dazugehörige DTD (siehe unten) mitgeliefert wird. Einzelne gebräuchliche Erweiterungen werden vom W3C begutachtet, standardisiert und freigegeben.

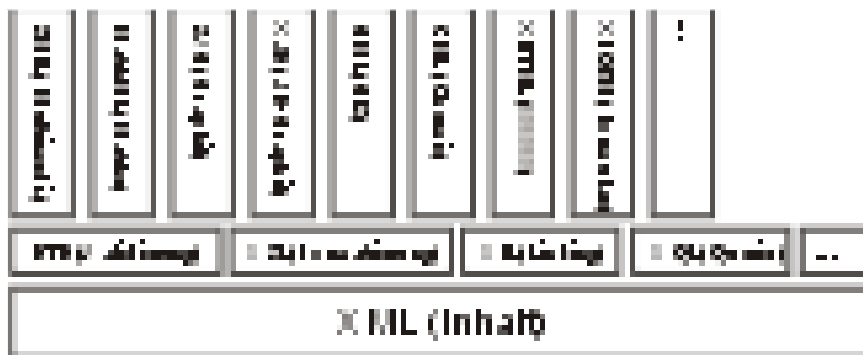


Abbildung 1: XML Architektur und Dialekte

Die eigentlichen Daten werden im XML-File abgelegt, das im Groben aus Elementen (entities, in "tags" formuliert) und Attributen (attributes) besteht. Die Struktur- und Syntaxprüfung erfolgt in der DTD (Document Type Definition). Hier werden die erlaubten Datentypen, die Namespaces (erlaubte tags), die Hierarchie und Verschachtelungen festgelegt. Ferner wird definiert welche Elemente und deren Attribute verpflichtend oder optional sind. DTDs können von Parsern, die es mittlerweile in allen Programmiersprachen und in den neuesten Browserversionen gibt, verwendet werden, XML-Files zu validieren und Fehler zu detektieren.

Weitere XML verwandte Technologien betreffen XSL (XstyleSheets, Nachfolger von CSS – verantwortlich für das Formatieren von Inhalten), XLL

(XlinkingLanguage, verantwortlich für das Verlinken von Elementen und Medien) und XQL (XqueryLanguage, für das strukturierte Abfragen von XML-Daten). Diese Liste wird laufend erweitert und aktualisiert.

Vektoren sollten also auch XML-konform geschrieben sein, das sind die weiter oben geschilderten Formate nur zu einem geringen Teil. Zur Animation und Interaktion erweckt werden die Elemente der WWW-Seite, wenn sie einmal korrekt angesprochen werden können mittels einer Scriptsprache, wobei hier JavaScript die weiteste Verbreitung hat. Voraussetzung ist, dass der Browser diese Sprache interpretieren kann.

## **5. Scalable Vector Graphics (SVG)**

### **5.1 Kurzbeschreibung**

SVG ist abgenommener Standard des W3C, und dort weiter in Entwicklung. Seit 2. November 2000 existiert die offizielle 1.0 Version der SVG-Spezifikation (Candidate Recommendation). SVG ist eine standardisierte in XML formulierte Sprache zur Beschreibung von 2D-Graphik mittels Vektorgraphik, Text und Rastergraphiken. Graphikformate wie man sie aus Graphikprogrammen kennt, können mittels CSS definiert werden um gleich eine ganze Gruppe von graphischen Objekten mit einem einheitlichen Aussehen zu versehen. Auf diese Art und Weise können auch mehrere gleichartige Objekte effizient in ihrem Aussehen verändert werden. Aufgrund der Kompatibilität der Styles (Formate) können SVG-Dateien sehr einfach in bestehende Web-Projekte integriert werden. Durch die Integration von SVG in das bestehende DOM können SVG Elemente mit den üblichen Javascript/Java-Schnittstellen gesteuert und verändert werden.

Hinsichtlich Interaktivität stellt SVG eine reiche Palette an Events (Erkennen von durch den User verursachten Ereignissen wie on mouse-over, on click, on keypress, etc.) zur Verfügung, die in der gleichen Art und Weise zu verwenden sind wie man sie bisher von traditionellen HTML-Elementen kennt. Jedes Element kann in seinen Ausprägungen (zum Beispiel Farbe, Form, Transparenz, Position, etc.) über "key frames" animiert werden. Die diesbezügliche Syntax ist SMIL-konform.

Entwickelt und unterstützt (v.a. in ihren Produkten) wird/wurde der SVG-Standard von allen namhaften Graphik- und EDV-Firmen und Organisationen, die im WWW-Bereich tätig sind: Adobe, Apple, Autodesk, Bit-Flash, Corel, HP, IBM, ILOG, Inso, Kodak, Macromedia, Microsoft, Netscape, Oasis, Open Text, Oxford University, Quark, RAL, Sun-Microsystems, W3C und Xerox. Dies wird in Zukunft eine breite Unterstützung hinsichtlich Import- und Exportfilter und der Entwicklung von Konvertern und Viewern garantieren. Da SVG ein sehr gut dokumentierter, offener XML-Standard ist, kann es aber auch leicht aus eigenen Scripten oder Programmen heraus generiert und konvertiert werden. Es ist zu erwarten, dass SVG auch von GIS-Softwarefirmen unterstützt wird und SVG über seinen Einsatz in WWW-Projekten als generelles Graphik-Austauschformat Bedeutung erlangen wird. Es ist neben Adobe Illustrator und PDF eines der wenigen gut dokumentierten ASCII-Grafikformate und bietet im Vergleich zu ersteren beiden noch wesentlich

mehr Möglichkeiten. Zur Darstellung von SVG-Dokumenten benötigt der Client im Moment noch ein Plugin (von Adobe), das vom Benutzer installiert werden muss. In zukünftigen Versionen von WWW-Browsern werden SVG-Interpreter jedoch standardmässig mitgeliefert werden. Es ist auch die Einbindung mittels Java-Applet von CSIRO möglich.

Dass Vektoren "skalierbar" sind, mag für Kartographen selbstverständlich sein. Laut W3C geht es hier aber mehr darum, dass das Vektor-Bild sich leicht den Umständen anpassen kann, also zum Beispiel beim Ausdrucken nicht starr der Anzeige des Bildschirms entspricht, sondern eine eigene Seitenaufbereitung stattfindet. Dieser Umstand ist bei anderen Plugins nicht gegeben. Ebenso wenig ist es möglich, Seiten die auf der herkömmlichen Raster/Layer-Methode aufbauen und die per JavaScript veränderbar sind, in einem anderen als dem ursprünglichen beim Laden angezeigten Zustand auszudrucken.

Es folgt nun eine nähere Beschreibung der graphischen Eigenschaften und der Fähigkeiten von SVG im Bereich Interaktion und Animation. Im Rahmen dieser Vorstellung wird auf technischen Quellcode verzichtet. Auch können die visuellen Effekte, die SVG erzielt, nicht in gedruckter Form wiedergegeben werden. Aus diesem Grund ist der Artikel an eine Internet-Seite gekoppelt. Bei den jeweiligen Punkten finden Sie eine Internetadresse, die Sie direkt zu einem konkreten, kartographischen Beispiel führt. Dort finden Sie auch weitere Erklärungen, sowie aufbereiteten Quellcode. Nützen Sie dieses Angebot um einen näheren Einblick zu gewinnen. Präzise Anweisungen zur Betrachtung dieser Inhalte finden Sie unter folgender Adresse:

[www.carto.net/papers/svg/first.html](http://www.carto.net/papers/svg/first.html)

Folgende Ausführungen basieren auf der W3C-1.0-Spezifikation, Working Draft, [W3C, 2000] vom 2. November 2000, sowie auf den Spezifikationen des Adobe-Plugins [Adobe, 2000] vom 5. Juni 2000. Im Rahmen der GIS-SIT 2000 in Fribourg wurde SVG im Groben von A. Neumann bereits vorgestellt [Neumann, 2000].

## **5.2 SVG-Renderingkonzepte (Bildschirm Aufbau)**

SVG Viewer simulieren den "painters algorithm", was vereinfacht bedeutet, dass sich überlappende Bereiche "übermalt" werden oder entsprechend der Transparenz-Werte die darunterliegenden Bildschirm-Pixel durchscheinen lassen. Objekte bis 2.5D (also auch Geländemodelle) können mit dem painters-algorithm korrekt abgebildet werden, sofern sie vorher korrekt vorsortiert wurden. Die Reihenfolge beim Rendering wird nach der Reihenfolge der Anordnung der Objekte in der Datei bestimmt. Objekte die zuoberst in der Datei aufgeführt sind werden zuerst gerendert, darauffolgende Objekte übermalen die darunterliegenden entsprechend der gewählten Muster und Opazität. Einzelne Objekte können zu Gruppen zusammengefasst werden. Grafikformate und Transformationen können auch auf ganze Gruppen wirken. Gruppen bekommen eine eigene Zeichenebene (Layer oder Canvas in DTP-Programmen) zugeordnet, diese können sich auch überlappen.

Im Viewer wird durch Tastenkombinationen und Mausführung ein Zoom-Tool (Lupe) und ein Pan-Tool (Hand) aktiviert. Mit dem Kontextmenü der rechten

Maustaste (Windows) lässt sich die ursprüngliche Ansicht wiederherstellen sowie Quellcode oder auch eine Hilfe aufrufen.

*Zeichenebenen: [www.carto.net/papers/svg/canvas.html](http://www.carto.net/papers/svg/canvas.html)*

*Eine Karte Österreichs, anhand der die elementaren Funktionen des Viewers, sowie das Ein- und Ausschalten von Ebenen vorgestellt wird.*

### 5.3 Koordinatensysteme und Transformationen

SVG verwendet ein kartesisches Koordinatensystem, dessen Ursprung sich in der linken oberen Ecke der Zeichenebene befindet. Verwendet man SVG für Kartendarstellungen, so muss die y-Achse invertiert werden: entweder man stellt den y-Werten jeweils ein Minus voran, oder man macht eine Gruppe um die zu transformierenden Objekte und weist ihr eine Invertierung mittels Matrixoperation zu. Mit dieser 3x3-Matrixoperation lassen sich die elementaren geometrischen Transformationen (Translation, Skalierung, Rotation, Scherung) im Einzelnen, oder kombiniert auf beliebige Gruppen, oder auch auf den gesamten SVG-Bereich anwenden. Transformationen lassen sich beliebig verschachteln, wodurch man beliebig viele lokale Koordinatensysteme erzeugen kann.

*Transformationen: [www.carto.net/papers/svg/matrix.html](http://www.carto.net/papers/svg/matrix.html)*

*An einem Objekt werden die Transformationen gezeigt.*

Zur Steuerung der Darstellung wird auch eine Definition der Anzeigefläche (viewbox) innerhalb einer WWW-Seite bzw. des sichtbaren Ausschnitts (viewport) einer SVG-Datei benötigt. Der viewport kann in der Datei fix vordefiniert sein oder auch interaktiv verändert werden, also zooming oder panning ohne User-Eingriff. So lässt sich beispielsweise eine Übersichtskarte in einem canvas mit einem grösseren Kartenausschnitt in einem anderen canvas koppeln, um die Ausschnittswahl und das Einzoomen zu erleichtern, bei gleichzeitiger Bewahrung der Übersicht – etwa wie es auch in der CD-Version des Atlas der Schweiz angewandt wird. Viewboxes erlauben das Verwenden von Echtkoordinaten, die Umrechnungsparameter für die Bildschirm-Koordinaten werden dabei mit angegeben. Es wird daher zwischen dem "viewport-coordinate-system" und dem "user space coordinate system" unterschieden (zum Beispiel Bildschirm oder Druckerkoordinaten). Viewboxes können beliebig verschachtelt werden.

*Anzeigefläche/-ausschnitt: [www.carto.net/papers/svg/viewbox.html](http://www.carto.net/papers/svg/viewbox.html)*

*Interaktives Zoomen auf verschiedene Ausschnitte, ohne die Maussteuerung des Viewers zu benutzen.*

Klarerweise machen bei der Kontrolle vom viewport, bei gleichzeitigem Beeinflussen der Transformationsparameter User-gesteuerte zoom- und pan-Funktionen keinen Sinn, dies kann man ggf. auch sperren.

### 5.4 Formatierungen

Ähnlich den Formatangaben, die man aus gängigen Textverarbeitungs- oder Grafikprogrammen kennt, kann man in HTML Formate definieren, sogenannte Stylesheets (CSS-Objekte). Diese erleichtern dem Kartographen grosse Projekte effizient bezüglich ihrer Darstellung (Farben, Füllungen, Striche, Texteingenschaften,

Position, etc.) zu verwalten und auf Wunsch zu ändern. Alle Objekte, denen Stylesheets zugeordnet sind, werden automatisch geändert, wenn die jeweilige CSS-Definition geändert wird. Das positive an der SVG-Implementation ist, dass es vollständig mit CSS, wie man es von HTML kennt, kompatibel ist und die Formate daher sowohl auf HTML-Elemente als auch SVG-Elemente angewandt werden können. Formate können dabei auf Gruppen, Untergruppen und Unterformate vererbt werden. Stylesheet-Angaben können in externen Dateien abgelegt werden und somit ganze Projekte zentral beeinflussen, oder einfach Gruppen oder einzelnen Objekten als Parameter zugewiesen werden.

*Formatierungen: [www.carto.net/papers/svg/styles.html](http://www.carto.net/papers/svg/styles.html)  
Eine Datei wird zwei Mal mit unterschiedlichen Styles angezeigt.*

## 5.5 Geometrische Grundelemente

Zu den geometrischen Grundelemente gehören: Rechteck, Kreis, Ellipse, Linie, Polylinie, Polygon und schliesslich das Path-Objekt. Einige der Basis-Elemente können zusätzliche Parameter, wie z.B. abgerundete Ecken, entgegennehmen. Das wichtigste Element ist jedoch das Path-Objekt, das gleichzeitig auch am flexibelsten ist. Man kann damit offene oder geschlossene Linienobjekte und Polygone zeichnen, die aus Koordinatenlisten bestehen, die sowohl absolute als auch relative Werte entgegennehmen können. Die Syntax ist mit jener von Postscript oder PDF verwandt: über Kommandos wie moveTo oder lineTo wird der Pfad gelegt. Um allen Anforderungen gerecht zu werden, können auch kubische und quadratische Bezierkurven, sowie elliptische Kurvenelemente einen Pfad bilden.

*Grundelemente: [www.carto.net/papers/svg/shapes.html](http://www.carto.net/papers/svg/shapes.html)  
Vorstellung der geometrischen Grundelemente von SVG.*

## 5.6 Clipping und Masking

Sämtliche SVG-Objekte können beschnitten (clip) und abmaskiert werden, resp. können auch als "clipping-path" dienen, inklusive Rasterbilder und Text. Masken können etwa an Kartenrändern eingesetzt werden, um Bereiche auszugrenzen. Ähnlich wie bei Pfaden muss auch hier bei komplexen Clipping-Polygonen die Regel für innenliegende und/oder sich selbst überschneidende Polygone angegeben werden.

*Clipping und Masking: [www.carto.net/papers/svg/clipmask.html](http://www.carto.net/papers/svg/clipmask.html)  
Bereiche der Österreichkarte werden beschnitten und abmaskiert.*

## 5.7 Text

SVG erlaubt beliebige Parameter bei der Erstellung von Textelementen: Schriftfamilie, Schriftart, Schriftgrösse, Schriftschnitt, Lage, Schriftstärke, Schriftbreite, Laufrichtung, Ausrichtung, Sperrung, etc. – eigentlich alle Parameter, die man aus dem DTP- und Kartographie-Bereich kennt. Insbesondere die Möglichkeit Text an path-Elementen auszurichten dürfte für Kartographen spannend werden. Leider kennt SVG in der jetzigen Spezifikation keinen Fliesstext. Um auch nicht-lateinische Schriften oder selbsterstellte Schriftarten (z.B. für Signaturen oder lineare Elemente wie Böschungskanten) richtig darzustellen, erlaubt SVG

Fonteigenschaften und –beschreibungen in externen Schriftsatzdateien abzulegen, oder aber auch fehlende Unicode–Characters durch eigene Graphiken oder Symbole zu ersetzen. Die Schriften können auch als Ganzes in die SVG–Datei eingebettet werden.

Bezüglich Textformatierung gelten wie bei HTML4 alle CSS–Regeln. Einzelne Zeichen kann man in ihrer Orientierung abdrehen. Die Schriftrichtung kann für nicht–lateinische Lettern bestimmt werden, auch beide Richtungen innerhalb einer Zeichenkette und vertikale Schriften werden korrekt abgebildet. Ebenso wie HTML4 unterstützt SVG Unicode, den Internationalen Standard zur Kodierung aller Schriftzeichen der Welt; westliche Sprachen gehören der Unicode–Familie 8859–1 (Latin1) an.

Ein herausragendes Merkmal von der SVG–Text–Implementierung ist, dass sich Textelemente auch von Suchmaschinen indizieren lassen, da ja die gesamte Graphik in XML abgelegt ist. Auch innerhalb des Plugins kann man Texte suchen und highlighten lassen; liegt der gefundene Text ausserhalb des aktuellen viewports, wird dieser richtig angepasst. Es erfolgt dabei aber keine automatische Zentrierung bei jeder Suche, um belastende viewport–Änderungen zu minimieren.

*Text–Element:* [www.carto.net/papers/svg/text.html](http://www.carto.net/papers/svg/text.html)

*Einige Beispiele.*

## **5.8 Farben, Füll–Muster, Farbverläufe, Transparenzen, Strichtypen**

Farbwerte werden im sRGB–Standard definiert. Dies ist ein eigener Farbraum für das Internet mit anzeigegerätespezifischen Parametern. Die Werte werden wie bei HTML4 in Hexadezimalwerten angegeben. Jeder Füllung, selbst Strichen und Texten, kann man Transparenzwerte zuweisen. Striche können als Attribut die Dicke, einen Linienendtyp, die Art der Darstellung bei Stützpunkten, Strichlierungsart, Strichlierungsoffset, etc. haben. An den Enden können Marker (beispielsweise Pfeile) angebracht werden. Als Spezialität können Marker auch an jedem Stützpunkt angezeigt werden, etwa um Stützpunkte bei geringen Winkeländerungen hervorzuheben.

Bezüglich Farbverläufen werden lineare und radiale Farbverläufe unterstützt, mit beliebig vielen Zwischenwerten und Stopwerten. Auch die Ausbreitungsmethode kann teilweise gesteuert werden. Selbst die Opazität lässt sich mit einem Verlauf versehen. Muster können sowohl aus Rastern als auch aus Vektoren bestehen. Derartige Muster werden gekachelt. Eine Kachel muss einmal definiert werden und wird dann entsprechend der Kachelgrösse wiederholt. Offsetwerte können dabei ebenfalls angegeben werden.

*Farben und Füllungen:* [www.carto.net/papers/svg/fill.html](http://www.carto.net/papers/svg/fill.html)

*Einige Beispiele.*

## **5.9 Interaktion, Scripting**

Zu den Basis–Interaktionsmöglichkeiten, die ein SVG–Viewer zur Verfügung stellt gehören zooming, panning, die Rückkehr zur ursprünglichen Ansicht und eine Ausdruckmöglichkeit im jeweiligen Zustand der Anzeige. Adobes Viewer hat

zusätzlich noch eine Text-Such-Funktion implementiert, die Möglichkeit das Anti-Aliasing ein- oder auszuschalten, sowie zu kopieren, abzuspeichern und den Quellcode anzusehen. Geplant ist auch die Möglichkeit aus einem standardisierten Set von Maus cursoren zu wählen, resp. sogar seine eigenen Bilder als Maus cursor zu verwenden. Zusammen mit den möglichen Events, liessen sich einfache Zeichenfunktionen implementieren, um innerhalb von kartographischen Applikationen z.B. Distanz- oder Flächenmessung anzubieten.

Hyperlinks können ähnlich wie auch in HTML dazu verwendet werden, auf andere Dateien zu verweisen, oder innerhalb des SVG-Dokuments auf einzelne Elemente. Dabei kann auch auf fix eingestellte "view-Elemente" verwiesen werden.

Es stehen in SVG drei Kategorien an Events zur Verfügung: Mouse Events, Keyboard Events, State Change Events (betreffen die Anzeige und den Ladezustand der SVG-Datei). Das Event-Handling geschieht analog zu HTML4. Die Kombination aller möglichen Events lässt eigentlich kaum Wünsche offen. Komplexe Anwendungen können in Kombination mit Scripten und Java-Applets erstellt werden. Mittels Javascript hat der Entwickler vollen Zugang auf das DOM von SVG und aller sonstigen Elemente im WWW-Browser. Hier muss allerdings eingeworfen werden, dass viele dieser Funktionen noch nicht einsetzbar sind, da noch nicht alle DOM-Bereiche innerhalb des Adobe-SVG-Plugins implementiert sind.

Kartographische Anwendungen reichen vom einfachen Ein- und Ausblenden von Elementen und Layern, Ändern derer graphischer Attribute, über das Reagieren auf mouse-events, zum Beispiel das Anzeigen von Objektdaten beim Überfahren mit der Maus, das Kombinieren von verschiedenen Ansichten (linked windows, zum Beispiel Übersichtskarte und Hauptkarte), das interaktive Verschieben, Skalieren und Rotieren von Elementen (zum Beispiel Puzzles für den Geographieunterricht) bis hin zu kleinen Applikationen, die zum Beispiel einem WWW-User kleine Digitalisierschritte erlauben und diese auf dem WWW-Server für weitere Bearbeitung aufzuzeichnen. Schliesslich lässt sich mit serverseitigen CGI-Scripten oder Java-Applets und/oder Java-Servlets auch eine Datenbankverknüpfung realisieren. Die Anwendungen sind unzählige und man darf gespannt sein, wie die Kreativität der Kartographen sich auf zukünftige Web-Projekte, im Hinblick auf die neuen technischen Möglichkeiten, auswirken wird.

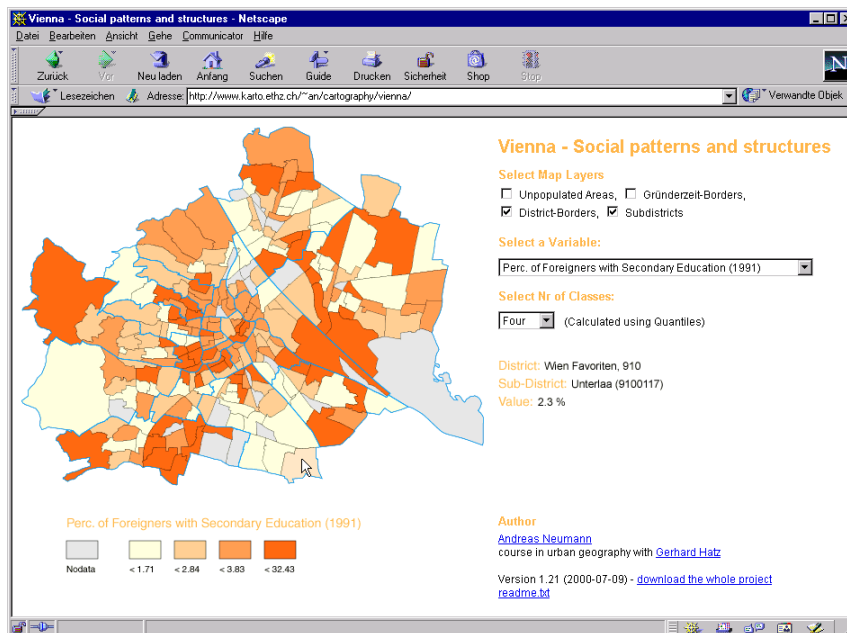


Abbildung 3: Statistische Anwendung: [www.carto.net/papers/svg/wien.html](http://www.carto.net/papers/svg/wien.html)  
Wien auf Zählbezirksebene, mit austauschbaren bevölkerungsstatistischen Werten und variablen Klassengrößen.

## 5.10 Animation

HTML4 erweitert mit CSS und JavaScript erlaubte das Erstellen einfacher Animationen, diese waren aber als solche nicht vorgesehen. Sollte zum Beispiel eine Bewegung eines Objekts von A nach B erfolgen, mussten alle Zwischenschritte dafür vordefiniert werden. SVG bietet hier Interpolationen und vieles mehr an.

Die Syntax und die Möglichkeiten der SVG-Animations-Komponente lehnt sich sehr stark an SMIL an, resp. ist dazu kompatibel. So lassen sich u.a. folgende Animationsvariablen festlegen: Zeitpunkt des Animationsstartes (meist relativ zu einem Event), Animationsende, Dauer, Wiederholung, Anzahl Wiederholungen, etc. Je nach Objekttyp sind unterschiedliche Parameter animierbar, so zum Beispiel: Farbwert, Position, Position relativ zu einem Pfad, Rotation und Skalierung, u.v.a .m.

Die Metapher der Zeitlinien (etwa bekannt aus dem Autorensystem Macromedia-Director, oder dem HTML-Authoring-Tool Dreamweaver) erlaubt das Festlegen von sogenannten "key frames", die zu bestimmten Zeitpunkten fix-vorgegebene Werte angeben. Dazwischen werden entweder diskrete Werte verwendet (nur key frames), oder Interpolationen angewandt. Es stehen schrittweise, lineare, oder Spline-Interpolationen zur Verfügung. Mittels "key time values" können Schlüsselzeitwerte angegeben werden. Fehlen diese so werden die Zeitpunkte für die key frames linear interpoliert. Für Positions-Animationen entlang von Pfad-Elementen besteht eine eigene motion-path-Anweisung, die auch ein distance-from-path Argument entgegennimmt. Dabei kann das Objekt auch automatisch zum Pfad ausgerichtet werden. Animationen können schliesslich beliebig verknüpft (kumulativ und nicht-kumulativ) oder auch anderen Kind-Elementen vererbt

werden.

Animation: [www.carto.net/papers/svg/anim.html](http://www.carto.net/papers/svg/anim.html)  
Einfache Animationen.

### 5.11 Metadaten und Erweiterbarkeit (Extensibility)

Metadaten dienen dazu Informationen über das Dokument in strukturierter Form abzulegen. Angaben werden etwa zum Autor, zum Publikationsdatum, Version, Titel, Kurzbeschreibung, etc. gemacht. Diese können ebenso wie Fremdcode (zum Beispiel JavaScript) in die SVG-Datei eingebettet werden.

Ähnlich wie bei VRML gibt es auch bei SVG die Möglichkeit, bereits beschriebene Elemente, die mittels ID eindeutig identifizierbar sind, später in der Datei wieder zu verwenden. So lassen sich mehr oder weniger komplizierte graphische Objekte und Gruppen zunächst einmal definieren, um dann später in modifizierter Form, etwa transformiert, wieder verwendet zu werden. In der Kartographie ist dies etwa von Relevanz um Symbole oder Piktogramme zu plazieren.

Erweiterbarkeit zählt zu den wichtigsten Eigenschaften von SVG sowie von allen DOM/XML-konformen Spezifikationen. Da SVG ja selbst in XML definiert ist, können beliebige andere in XML definierte Standards, zum Beispiel MathML, XHTML, SMIL, u.v.a.m., innerhalb SVG eingebettet und darauf zugegriffen werden. So könnte man beispielsweise zu Polygonen als Teil einer thematischen Karte deren statistische Daten, in Form von XML, ablegen. Eine Javascript-Funktion könnte die Werte dann jeweils einlesen, verarbeiten und in geeigneter Form, zum Beispiel als Kartodiagramm in SVG darstellen, die parallele Ausgabe der Werte als HTML-Tabelle könnte als weiteres Service angeboten werden.

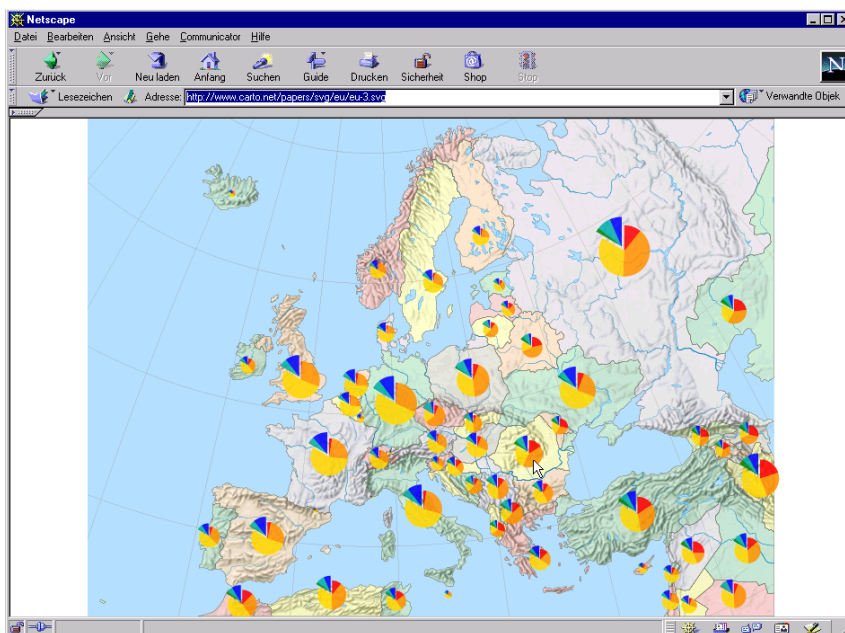


Abbildung 4: Kartodiagramme: [www.carto.net/papers/svg/diagr.html](http://www.carto.net/papers/svg/diagr.html)  
Eine Karte Europas mit Kreisdiagrammen die mittels JavaScript direkt aus den Ursprungsdaten dynamisch generiert werden.

## 5.12 SVG–Export

Die Graphik wird wie oben besprochen in einem SVG–Quellcode abgespeichert und vom Client, dem Browser, interpretiert um korrekt angezeigt zu werden. Um SVG–Code zu generieren gibt es grundsätzlich drei Möglichkeiten – um anspruchsvolle Ergebnisse zu erzielen wird man sie kombinieren müssen: den Export von einem Graphik– oder DTP–Programm, den Export von einem GIS und das Editieren bzw. das Transformieren des Codes mittels Zeichenkettenbearbeitung. Schliesslich ist es auch noch möglich SVG serverseitig dynamisch zu generieren mittels CGI–Scripten, aus Java Servlets [[Sun Microsystems, 2000](#)], Datenbanken und/oder Applikationen.

Der Export aus einem Graphik– oder DTP–Programm ist jener der unmittelbar anzeigbare Resultate liefert. Voraussetzung dafür ist ein Programm, das den SVG–Export unterstützt; dies sind zur Zeit Adobe Illustrator, CorelDRAW, Mayura Draw und TrajectoryPro in ihren letzten Produktversionen sowie einige Zeichenprogramme aus dem Open–Source Bereich (Sketch, Killustrator, etc.). Adobe plant SVG in seine gesamte Produktpalette zu integrieren. Macromedia mit seinem in der Kartographie weit verbreiteten Freehand ist hier noch nicht vertreten. In diesem Zusammenhang ist aber zu erwähnen, dass Macromedia das besprochene Flash–Format unterstützt, und SVG somit ein Konkurrenzprodukt darstellt. Dennoch ist Macromedia an der SVG–Spezifikation beteiligt, was vermuten lässt, dass hier demnächst Unterstützung angeboten werden wird.

Wie bei jedem Export aus einem Graphik–Programm wird beim Abspeichern die Datei in das Ziel–Format SVG geschrieben. Bei diesem Schritt werden Details zu gewünschten Modi (zum Beispiel Einbetten der Schrift, oder Auflösung des Orientierungsrasters oder Art der Ablage der Style–Informationen) abgefragt. Namen von Zeichenebenen und Objekten werden mit in die SVG–Datei übernommen, womit diese fertig zur Ansicht ist. In Illustrator wird auch ein Belegen von SVG–Objekten mit Links sowie das Einbinden von einfachen Ja vaScript–Funktionen angeboten. Dass dies für anspruchsvolle Anwendungen nicht ausreicht, kannte man schon von ähnlichen Export–Möglichkeiten für Image–Maps – "händisches" Bearbeiten des Codes wird nötig sein.

Oft liegen aber Daten direkt im GIS vor: auch von dort besteht die Möglichkeit SVG–Dateien zu generieren; da aber z.Z. kein Export vorgesehen ist, wird man die Daten (meist die Vektorinformation) in ein lesbares Text–Format exportieren. Dieses muss im Anschluss mittels Zeichenkettenbearbeitung (zum Beispiel mit PERL) in SVG–tauglichen Code umgeschrieben werden. Anschliessend wird rund um die so entstandenen SVG–Objekte noch die restliche Information geschrieben, die eine SVG–Datei ausmachen.

SVG unterstützt wie gesagt Bezier–Kurven. GIS–Anwendungen unterstützen diese häufig nicht. Da SVG aber für das Internet geschaffen ist, gilt es die Dateigrösse gering zu halten. Komplizierte, für SVG überbestimmte, Kurven lassen sich in einem Graphikprogramm relativ bequem zu Bezier–Kurven konvertieren, bzw. kann dabei auch die Anzahl der Stützpunkte reduziert werden. Aus diesem Grund ist der Weg über Graphikprogramme jenen direkt aus den GIS unter Umständen vorzuziehen.

Um noch besser dateigrössenoptimiert arbeiten zu können, ist es möglich die

gesamte SVG-Datei zu komprimieren und sie in dieser komprimierten Version dem WWW-Browser zu übergeben, dafür muss sie aber korrekt referenziert in die HTML-Datei eingebettet werden.

## 6. Zusammenfassung

Bis dato war jeder ernsthafte Versuch Kartographie im Internet zu betreiben an sehr strikte anzeigetechnische Vorgaben gebunden, die graphisch hochqualitative Kartographie unmöglich machten. Das Gros von Kartographen an dieser Front war vorrangig damit beschäftigt Notlösungen (sog. workarounds) zu kreieren, anstatt sich den eigentlichen kartographischen Inhalten widmen zu können. Der neue Vektorstandard für das Internet, SVG, erlaubt es erstmals die Last der Sorgen um die graphische Ausgabe zu verringern, und sich mehr den Interaktionen, die die Bildschirmkartographie noch immer mit sich bringt, zuzuwenden. Der offene Charakter des Internets und der Erfolg des Open-Source Modells zeigt Möglichkeiten und Chancen für die Kartographie auf und sollte wegweisend sein, für weitere Bereiche, wie Datenaufbereitung und Vertrieb.

## 7. Literatur/Quellen

ADOBE SYSTEMS INCORPORATED (2000): SVG, Scalable Vector Graphics, Release notes, [www.adobe.com/svg/indepth/releasenotes.html](http://www.adobe.com/svg/indepth/releasenotes.html) (2000.06.25)

cgmOPEN.org (1999): WebCGM Profile, [www.cgmopen.org/webcgm/webcgm.htm](http://www.cgmopen.org/webcgm/webcgm.htm) (2000.07.27)

D'AMORE, Stefan (2000): SVG von Adobe soll der Konkurrenz Druck machen, Macwelt, Heft 8/2000, S. 100–104.  
(<http://www.tecchannel.de/internet/96/index.html>)

GABORIT, Gaëtan (2000): La cartographie dynamique sur le Net avec SVG, <http://svgmap.free.fr/> (20000926)

ISO, International Organization for Standardization (1999): Introduction to ISO, [www.iso.ch/infoe/intro.htm](http://www.iso.ch/infoe/intro.htm) (2000.03.11)

NEUMANN, Andreas (2000): Social Patterns of Vienna, Statistical Maps with SVG, [www.karto.ethz.ch/~aneumann/cartography/vienna](http://www.karto.ethz.ch/~aneumann/cartography/vienna).

SCHLIMM, Reinhold (1998): Aufbau eines Kartographischen Informationssystems im World Wide Web, [www.geog.fu-berlin.de/~rschlimm/kiv/KN-1-98.html](http://www.geog.fu-berlin.de/~rschlimm/kiv/KN-1-98.html) (2000.06.12). Auch in: Kartographische Nachrichten, 1998. Jg. 48, Heft 1, S. 1–8.

STATMARKET (o.J.): Web SideStory Inc. [www.statmarket.com](http://www.statmarket.com) (1999.09.20 – nunmehr leider nur noch kommerziell abrufbar).

SUN MICROSYSTEMS (2000): JSP and SVG, <http://www.sun.com/software/xml/developers/svg/jsp/>(2000.07.25).

W3C, World Wide Web Consortium (2000): Scalable Vector Graphics (SVG) 1.0 Specification, Candidate Recommendation, 02 August 2000,

[www.w3.org/TR/SVG](http://www.w3.org/TR/SVG) (2000.08.03)

WINTER, Andreas (2000): Dynamically generated statistical Maps of Europe by using SVG, [www.carto.net/papers/svg/diagr.html](http://www.carto.net/papers/svg/diagr.html) (2000.08.12)

WINTER, Andreas u. Andreas NEUMANN (2000 ): Cartographers on the Net – Scalable Vector Graphics, [www.carto.net/papers/svg](http://www.carto.net/papers/svg) (2000.08.12)

DICKMANN, Frank (1997): Kartographien im Internet; in: Kartographische Nachrichten, 1997, Jg. 47, Heft 3, S. 87–96.

GARTNER, Georg (1996): Internet für Kartographen; in: Kartographische Nachrichten, 1996, Jg. 46, Heft 5, S. 185–190.

GARTNER, Georg (1999): Internet–Kartographie: (R)Evolution oder Sackgasse, in: Kartographische Nachrichten, 1999, Jg. 49, Heft 3, S. 98–104.

NEUMANN, Andreas (2000): Hochqualitative Vektorgraphik im Internet mit Hilfe von SVG; in: Tagungsband GIS 2000 SIT – GIS/SIT 2000, SOGI Tagung Fribourg, Workshop 2. 11.4.2000, Internet–Technologien für GIS.

NEUMANN, Andreas (2000): Neue Entwicklungen in der Webkartographie: SVG, VRML und andere ..., in VPK (Vermessung, Photogrammetrie und Kulturtechnik): Nr 10/2000, S. 621–624.

PETERSON, Michael P. (1999): Trends in Internet Map Use; in: Proceedings of the 19th International Cartographic Conference and 11th General Assembly of ICA, Ottawa, Canada, Vol. I, S. 571–580.

TOMLIN, Dana (1990): Geographic Information Systems and Cartographic Modelling, Prentice–Hall, Englewood Cliffs, New Jersey.

---

version 2.0, Oktober 2000.<script language="JavaScript" type="text/javascript"><!--

document.write(document.lastModified.toLowerCase());

// --></script>

[version 1.6](#) von März 2000.

© [andré m. winter](#) & [andreas neumann](#)

<span class="smaller">originalstandort für quellenangabe

<http://www.carto.net/papers/svg/></span> </div>