

Aktuelle Webtechniken und deren Anwendung in der thematischen Kartographie und der Hochgebirgskartographie

Lorenz HURNI, Andreas NEUMANN und Andréas M. WINTER

Institut für Kartographie, ETH Zurich
[e-mail:hurni.neumann@karto.baug.ethz.ch](mailto:hurni.neumann@karto.baug.ethz.ch)
<http://www.karto.ethz.ch/>, <http://www.carto.net/>

Freytag & Berndt KG
[e-mail:winter@freytagberndt.at](mailto:winter@freytagberndt.at)
<http://www.freytagberndt.at/>, <http://www.carto.net/>

1 Einleitung

Die Kartographie wurde immer wieder von neuen Medien und Darstellungstechniken beeinflusst. Im Laufe ihrer Geschichte sind mehrmals ihre Informationsträger gewechselt oder ergänzt worden. In den letzten Jahrzehnten ist diese Entwicklung durch die Fortschritte in der Informationsverarbeitung zusätzlich beschleunigt worden. Interessanterweise verdrängten neue Medien kaum ihre Vorgänger, sondern ergänzten diese meist, sodass wir heute eine diversifizierte Palette an verschiedenen Medien und Datenträgern haben, jedes davon mit spezifischen Vor- und Nachteilen.

Eines dieser neuen Medien, das in jüngster Zeit verstärkt Kartographen beschäftigt, ist sicherlich das World Wide Web. Webmapping-Projekte, als weitere Evolutionsstufe der CD-basierten Publikationen, erben auch deren Vorteile: interaktive Präsentation, Verknüpfung mit Datenbanken, Analyse-Möglichkeiten und gute Integration verschiedener Medien. Zu den spezifischen Vorteilen des Webmappings zählen die potentiell kürzeren Aktualisierungszyklen, die weltweite Verfügbarkeit, die kostengünstige Vertriebsmöglichkeit sowie die weitgehend grössere Plattform-Unabhängigkeit. Da Webmapping Systeme oft unter Zuhilfenahme offener Standards implementiert sind, lassen sich Webmapping-Projekte einfach in bereits bestehende Produktions-Systeme und Workflows einbauen. Ist man in der erfreulichen Lage, derartige Systeme von Grund auf neu entwickeln zu können, so sollte auf jeden Fall auf eine Trennung von Inhalt, Datenstruktur und Präsentationsform geachtet werden. Dieser anfängliche Mehraufwand wird sich im Laufe eines Projekts mehrfach auszahlen, besonders wenn man künftig vermehrt in Richtung »Cross-Media-Publishing« arbeiten wird.

2 Web- und Open-Source Philosophie

2.1 Weshalb sich die Einhaltung allgemein gültiger Standards lohnt

Ursprünglich war das WWW als offenes Medium zum Austausch von wissenschaftlichen Informationen gedacht. Die ursprünglich standardisierte Dokumenten-Strukturierungssprache HTML wurde jedoch bald »verunglimpft« durch proprietäre Erweiterungen von Unternehmen, deren Ziel es war, dadurch einen Wettbewerbsvorteil gegenüber den Konkurrenten zu schaffen. Dafür setzte man zudem häufig auf Copyrights und Patente. So war bereits die Einführung der Rasterformate (.gif und .jpeg) innerhalb von HTML-Seiten eine proprietäre Erweiterung seitens Netscape. Im Rahmen der sogenannten »Browserkriege« zwischen Netscape und Microsoft kamen noch unzählige weitere »Verstösse« gegen offizielle Standards hinzu. Erweiterungen sind grundsätzlich zu begrüßen, deren Einführung sollte allerdings wohlüberlegt, standardisiert und in Zusammenarbeit der beteiligten Institutionen und Unternehmen erfolgen.

Glücklicherweise formierte sich im Herbst 1994 das W3C (World Wide Web Konsortium). Es zählt heute ca. 500 Mitglieder mit Vertretern aus universitären und kommerziellen Bereichen. Das W3C verfügt weltweit über Zweigstellen und unterhält eine gut ausgebauten Infrastruktur. Aufgabe des W3C ist es plattformunabhängige und projektübergreifende Basisstandards zu spezifizieren die als Grundlage für Software-Implementationen und als Nachschlagewerk und Empfehlung für Webseiten-Entwickler gelten sollen. Heute gibt es dank standardisierter Datenformate neben Internet-Explorer und Netscape weitere gute Browser-Alternativen (Opera, Mozilla, Konqueror, ...).

Die Philosophie von »Open Source«¹ und dem W3C hat bereits Früchte getragen und immer mehr grosse Firmen dazu bewegen, ihre internen Standards offenzulegen oder zusammen mit anderen, z.T. sogar Konkurrenzfirmen, zu kooperieren. Die Folge ist ein besseres Image bei Kunden und Entwicklern, die Öffnung für grössere Märkte und der Zwang zur besseren Dokumentation der eigenen Produkte. Kunden kaufen erfahrungsgemäss lieber gut dokumentierte, erweiterbare Produkte als sog. »Blackboxes«. Ein klassisches Beispiel einer kommerziellen Firma, welche ihre Datenformate immer offengelegt hat, ist Adobe mit ihren Formaten Postscript, PDF, Adobe Illustrator und SVG. Alle Formate wurden offengelegt und erfreuen sich daher grosser Beliebtheit im Daten- und Dokumentenaustausch. Obwohl die Konkurrenz dadurch tiefere Einblicke in Adobe Produkte gewinnen konnte, ist Adobe heute eine der wenigen Softwarefirmen, welche kontinuierlich ihre Gewinne steigern kann. Solange man technologisch fundierte und benutzerfreundliche Produkte, gemeinsam mit einem durchdachten Service- und Support-Konzept anbieten kann, braucht man sich offensichtlich vor der Konkurrenz trotz Offenlegung wenig zu fürchten.

2.2 Weshalb der Einsatz von »Open Source«-Systemen sinnvoll ist

Kern der Open-Source Philosophie ist, dass der Benutzer den Quellcode eines Programmes, und damit dessen Architektur, einsehen kann. Dadurch ist es für andere möglich, Verbesserungen anzubringen, die wiederum allen Benutzern zugute kommen.

¹ Die Open-Source Philosophie wird in Kapitel 2.2 erklärt.

Jeder stellt sein Wissen und Können anderen zur Verfügung und profitiert gleichzeitig von anderen. Diese Programmierphilosophie führt in der Regel dazu, dass man das »Rad nicht immer wieder neu erfinden« muss. Je nach Art der Open-Source Lizenz wird allerdings im Gegenzug verlangt, dass Folgeprodukte welche auf Open-Source aufbauen, wiederum der Öffentlichkeit zur Verfügung gestellt werden muss. Heute basieren rund zwei Drittel der Internet-Infrastruktur auf Open-Source. Referenzprojekte sind etwa Apache als Webserver, BIND als DNS-Server, Sendmail/Postfix als E-Mail-Server, WUFTPd und PROFTPd als FTP Server, MySQL und PostgreSQL als Datenbankserver. Ebenso halten die drei Open-Source Scripting-Sprachen Perl, PHP und Python einen grossen Marktanteil im Bereich der Scripting-Sprachen. Neu sind auch Open-Source Projekte im Bereich der Kommunikationsserver. Open-Source Betriebssysteme wie GNU/Linux und FreeBSD zählen schon seit mehreren Jahren zu den Server-Plattform die sich am schnellsten verbreiten.

Es sind jedoch nicht nur universitäre und Hackerkreise, die sich für Open-Source Projekte interessieren. Unterstützung gibt es mittlerweile auch von Grossfirmen wie IBM, HP, Oracle und Compaq. Offene Projekte werden aus verschiedenen Gründen geschätzt: man löst die Abhängigkeit gegenüber einzelnen Firmen, spart z.T. Lizenzkosten, man kann die Projekte leichter auf eigene Bedürfnisse anpassen und erhält tiefere Einblicke in den inneren Aufbau eines Projekts. In der Regel können Kunden und Entwickler sich mit offenen Produkten stärker identifizieren, da sie zur Weiterentwicklung des Projektes beitragen können. Open-Source Entwickler suchen in der Regel den Kontakt zu den Benutzern und versuchen diese aktiv miteinzubinden. Dies steht im Gegensatz zu gewissen kommerziellen Firmen, die ihre Entwickler von den Benutzern teilweise »hermetisch« abschotten, und entsprechend ihrer Firmenphilosophie keinerlei Hintergrundinformationen über ihre eigenen Produkte veröffentlichen. Der Druck der Open-Source Bewegung auf kommerzielle Firmen nimmt jedoch zu: Heute beginnt selbst Microsoft den Quellcode mancher Produkte unter einer sog. »shared license«² zu veröffentlichen.

Patches und Fixes werden bei Open-Source Projekten schneller publiziert als bei kommerziellen Produkten. Nicht zuletzt haben Open-Source Programmierer in der Regel ein sehr hohes Verantwortungsbewusstsein gegenüber ihren Benutzern und stehen konstruktiver Kritik und Wünsche ihrer Anwender sehr offen gegenüber. Anpassungen und Verbesserungen von Drittentwicklern fliessen häufig wieder in das Hauptprodukt ein. Studien und Statistiken zeigen, dass dermassen »gereifte« offene Software in der Regel zuverlässiger und sicherer arbeitet als geschlossene.

Die weitverbreitete Meinung, dass man mit Open-Source Projekten kein Geld verdienen kann, ist meist falsch. Es haben sich jedoch vielfach die Vertriebsarten und die Verdienstmöglichkeiten geändert: Verdiente eine klassische Software-Firma v.a. am Vertrieb der Software selbst, so werden Open-Source Unternehmen immer mehr zu Dienstleistern, die kundenspezifische Anpassungen vornehmen und Support für Projekte leisten. Dabei ist der Kunde bei Verwendung von standardisierter Open-Source-Software, und speziell durch die gute Dokumentation, nicht mehr an einen einzelnen Dienstleister gebunden. Es kommt nicht von ungefähr, dass immer mehr Softwareriesen

² eine stark eingeschränkte »read only« Quellcode-Lizenz

die Bedeutung von firmenübergreifenden Open-Source Projekten erkennen, diese aktiv fördern und z.T. als Geschäftsphilosophie übernehmen.

Auch in der Kartographie wird offene Software in Zukunft eine grössere Rolle spielen. Aktuelle Projekte wie der Open-Source »Internet-Mapserver«, GRASS GIS (von der Uni Hannover weiterentwickelt), GIMP, PovRay und viele mehr, repräsentieren erst den Anfang eines neuen Trends in der Graphik- und Kartographie-Software [freegis.org; 2001]. Die Symbiose zwischen Vernetzung, offenen Standards und Software, wird hoffentlich auch in der Kartographie zu einer vermehrten und verbesserten Zusammenarbeit zwischen Kartographie-Firmen, Universitäten, Software-Entwicklern und dem Kartenbenutzer führen.

3. WWW-Basistechnologien

3.1 XML als Basis aller künftigen Web- und »Cross-Media-Publishing-Strategien«

XML³ ist eine textbasierte Markup-Language⁴ und wurde auf Basis ein vereinfachter Nachfolger der SGML⁵-Sprache entwickelt. XML wurde vom W3C und vielen Softwarefirmen als künftiges Basis-Datenformat für Webpublikationen, für die Speicherung von Unternehmensdaten, für Konfigurationsfiles und GUI-Ressourcenbeschreibungen, sowie für den Datenaustausch u.v.m. ausgewählt. XML ist eigentlich eine »Metasprache«, die dazu dient, weitere Sprachen und Datenmodelle zu beschreiben. Sie ist als solche sehr flexibel und ausbaubar. Alle XML-basierten Sprachen können daher auf die gleichen Basistechnologien zurückgreifen. Bereits erworbenes XML-KnowHow kann damit vielfältig weitergenutzt werden. Zu den wichtigsten Vorteilen von XML zählt die strikte Trennung von Inhalt, Datenstruktur und Präsentation/Formatierung. XML ist ein plattformunabhängiger Standard. Da er auf Textelementen basiert, ist er auch für Nicht-Experten gut lesbar und für die langfristige Archivierung geeignet. Es können aber auch binäre Daten wie Rastergraphiken, Videos und Sound-Elemente in einem XML-Dokument referenziert werden.

Durch die bereits erwähnte Trennung ist XML sehr gut als Basis für »Cross-Media-Publishing« geeignet. Aus den gleichen Daten können textbasierte Berichte, Tabellen, Graphiken und Diagramme oder auch thematische Karten generiert werden. Zur Formatierung und Transformation in andere textbasierte File-Formate dienen XSLFO (Formatting Objects) und XSLT (Transformation)⁶. XML Parser können XML-Basisdaten mittels der »DTD2 (Document Type Definition) und dem »Schema« auf

³ XML ist ein Acronym für »Extensible Markup Language«

⁴ eine »Markup Language« dient zum Strukturieren und Auszeichnen von Datensätzen

⁵ SGML ist ein Acronym »Standard Generalized Markup Language« und ist der Urvater von HTML und XML. SGML ist heute wesentlich komplexer und kräftiger als HTML/XML und ist als Auszeichnungs- und Dokumenten-Strukturierungssprache bereits jahrelang im Einsatz

⁶ XSL ist ein Acronym für Extensible Stylesheet-Language; XSLFO ist für die formatierte Ausgabe zuständig, XSLT für die Transformation von einem textbasierten strukturierten Format in ein anderes.

korrekte Syntax und auf die Verwendung erlaubter Datentypen und Wertebereiche testen. XML erfährt derzeit breite Unterstützung durch Industrie, Wirtschaft und Open-Source Projekte. Import/Export-Filter und Parser⁷ existieren bereits für die meisten Produkte und Programmiersprachen.

3.2 DOM – das »Document Object Model« als hierarchisches Zugriffsmodell zur Kontrolle aller Elemente und deren Attribute

Das DOM ist eine plattform- und sprachneutrale Schnittstelle, die Programmen und Scripten dynamischen Zugriff auf Inhalt, Struktur und Präsentation/Formatierung von Objekten und Dokumenten erlaubt. Das Dokument und dessen Struktur kann zur Laufzeit eingelesen werden. Berechnungen oder Umformatierung können damit vorgenommen und das Ergebnis wieder zur Laufzeit in die aktuelle Seite eingearbeitet werden. Offene Standards wie SVG⁸ und SMIL⁹ erlauben einen transparenten Zugriff für Scripting- und Programmiersprachen via DOM. Dabei ist eine Webseite hierarchisch, d.h. in einer Art »Baumstruktur« organisiert. Interaktive und offene Webkartographie-Systeme bauen in der Regel auf Scriptingsprachen und DOM-Zugriffen auf.

3.4 Scripting-Sprachen auf dem Vormarsch

Der Erfolg des WWW und der Bedarf der Entwicklung interaktiver Webseiten machte es notwendig, einfache Scripting-Sprachen zur Verfügung zu stellen, welche auch »Nicht-Programmierern« einen schnellen und einfachen Einstieg in die Programmierung solcher Webseiten erlauben. Scriptingsprachen werden im Quelltext übertragen und zur Laufzeit interpretiert und ausgeführt. Dies unterscheidet sie von kompilierten Sprachen wie C++, Pascal oder teilweise auch Java. Scripting-Sprachen sind üblicherweise hoch portabel¹⁰, zur Ausführung der Programme wird jedoch auf der ausführenden Maschine ein Interpreter benötigt. Man unterscheidet zwischen serverseitigen und clientseitigen Scriptingsprachen. Zu ersteren zählen v.a. Perl, PHP, Python und Javascript, Vertreter der letzteren sind v.a. Javascript, VBScript, in jüngster Zeit aber auch Perl und Python. Obwohl Scriptingsprachen meist einfach und schnell zu erlernen sind, erlauben sie die Erstellung von komplexen Projekten. Serverseitige Skripte dienen v.a. dem Datenbankzugriff, der Konvertierung, der Auswertung von Formulareingaben und der Anbindung von scriptingfähigen Applikationen. Clientseitige Skripte dienen v.a. der Interaktivität, der Animation und der Ausführung von einfachem Programmcode.

⁷ Ein Parser hilft dem Programmierer beim Einlesen von strukturierten Dateien und gibt ihm die Möglichkeit, einfach und selektiv auf die eingelesene Objekthierarchie zuzugreifen.

⁸ SVG ist das Acronym für »Scalable Vector Graphics« und dient zur Darstellung interaktiver Vektorgraphik. Es wird später im Text darauf eingegangen.

⁹ SMIL ist das Acronym für »Synchronized Multimedia Language« und dient zur Präsentation multimedialer Inhalte.

¹⁰ Unter verschiedenen Betriebssystemen lauffähig.

4 Qualitätskriterien jenseits des »Multimedia- und Virtual-Reality Rummels«

Multimedia- und WWW gehören mit zu den grössten »Modewörtern« des vergangenen Jahrzehnts. So gut sich diese neuen Medien zur zeitgemässen und informativen Präsentation geographischer Inhalte auch eignen, so findet man trotzdem vor allem viele Negativ-Beispiele. Sehr viele Portalseiten sind überfrachtet – die eigentliche Information ist versteckt hinter Werbe-Bannern und blinkenden, hüpfenden Logos und Icons. Das Layout vieler Webportale wechselt häufig oder es ist innerhalb des gleichen Projekts inkonsistent und in den wenigsten Fällen benutzerfreundlich gehalten.

Auch in der Webkartographie gilt der Grundsatz »weniger ist mehr«. Layouts und Farbwahl sollten schlicht gehalten und Legende wie Benutzerinterface klar gegliedert sein. Animation wie auch Multimedia sollte man nur dort einsetzen, wo sie wirklich zur Unterstützung der Präsentation und dem besseren Verständnis dienen. Aufgezwungene Eingangsanimationen einer Webseite oder Multimedia-Präsentation wirken störend und sind meist beim zweiten Besuch bereits langweilig. Dass sich Web-Anwender durch derartige Animationen und versteckte Werbebotschaften belästigt fühlen, bestätigt auch der Erfolg gängiger Filtersoftware. Derartige Software hilft, unerwünschte Werbebanner und Animationen zu unterdrücken oder abzuschalten.

Auch in der Kartographie hat der »Multimedia-Rummel« stark zugenommen und die Verkaufsregale werden überschwemmt von billigen, schlecht funktionierenden und oft nur wenig informativen CD-ROM Produkten. (Ähnliches gilt natürlich analog für Webseiten). Einfach gehaltene Informationssysteme, mit sparsam eingesetzten Multimediakomponenten sind effizienter für die Informationsvermittlung als »Multimedia-überfrachtete« Produkte. Multimedia-Elemente können nur sehr kurzfristig von geringer graphischer Qualität, nicht eingehaltenen kartographischen Grundsätzen und einem geringem Informationsgehalt ablenken.

Zum gleichen Thema ein Zitat von Stefan Münz, einem der führenden HTML-Profis im deutschsprachigen Raum: *»Webdesign ist Kommunikationsdesign, kein Werbe-Design. Leider bemühen große Firmen die gleichen Agenturen für ihren Webauftritt, die sie auch für ihre Hochglanzprospekte bemühen. Das ist ein fataler Fehler. Und Interaktivität, so wie es meist verstanden wird, ist ein Schlagwort, bei dem ich jedesmal gähnen muß. Entweder suche ich nach Informationen im Web, dann brauche ich kein sich als interaktiv ausgebendes Gezappel. Oder ich möchte aktiv an etwas teilnehmen, etwa an einem Diskussionsforum, an einer Umfrage – auch dann brauche ich dieses Gezappel nicht. Die meisten echten interaktiven Anwendungen im Web sind serverseitig mit Hilfe der klassischen CGI-Schnittstelle realisiert – ohne Gezappel. Java hat in genau zwei Bereichen im Web Erfolg: beim Online-Banking und bei web-basierten Chats. DHTML dümpelt bislang vor sich hin, weil es unausgereift und zu uneinheitlich implementiert ist. Und Flash mag einiges können, aber bislang kenne ich es nur von nervenden Trailern, Hirngeburten web-fremder Marketing-Strategen. Daher gilt: wer Informationen zu bieten hat, braucht ein ansprechendes, sympathisches Layout und vielleicht auch eine Datenbank, aber er braucht keinen »Rolloverkill« – so etwas ist keine echte Interaktivität. Die Anwender werden dankbar sein, wenn sie nicht abgelenkt werden und sich auf das konzentrieren können, was sie suchen.«* [MUENZ, 2001]

5 Vektorformate für das Internet

Obwohl die Definition des HTML-Standards bereits im Jahr 1992 erfolgte, gab es bis zum Jahr 2000 keinen offiziellen Standard für Vektorgraphiken. Es gab jedoch zahlreiche Versuche proprietäre Standards einzuführen, darunter »Macromedia Flash« als prominentesten Vertreter, das »Drawing Web Format« (kurz DWF) von Autodesk oder auch VML¹¹ von Microsoft. Für eine vollständigere Liste siehe [WINTER/NEUMANN, 2000]. Durch den Einsatz von ActiveX-Modulen¹² oder Java2D¹³-Applets kann man alternativ dazu Programmcode schreiben, welcher beliebige Formate lesen und rendern¹⁴ kann. Das W3C hat aktiv nur zwei Ansätze verfolgt: WebCGM, eine weiterentwickelte Form des CGM-Standards, und SVG. WebCGM ist ein binärer Graphikstandard der Graphiken kompakt repräsentieren kann, aber wenig Interaktion und Animation zulässt. SVG¹⁵ ist bezüglich Sprachumfang und Features wesentlich mächtiger und XML basiert.

Sämtliche Vektorgraphikformate benötigen derzeit zur Darstellung Plugins. Diese werden entweder bereits mit dem Webbrowser mitgeliefert oder sie müssen vom Benutzer selbst nachgerüstet werden. Für die Kartographie sind v.a. 2 Vektorformate interessant: SWF (Macromedia Flash) und SVG. Die nachfolgende Tabelle stellt die beiden Formate einander gegenüber. Zu beachten ist, dass sich der Vergleich auf die beiden Spezifikationen von Macromedia und W3C beziehen.¹⁶ Nicht jede Spezifikation wird bereits von allen Playern/Viewern unterstützt. Macromedia macht einige der Mankos im Fileformat durch Funktionen im Autorensystem wett. Dies geht jedoch zu Lasten der Entwickler, welche das Fileformat direkt schreiben wollen. Zielgruppe dieses Vergleiches ist deshalb der ambitionierte und technologisch interessierte Kartograph/GIS-Entwickler, der flexible, hochqualitative Webmapping-Systeme entwickeln will.

Eigenschaft	SWF	SVG
Kompatibilität, Erweiterbarkeit		
Offizieller W3C Standard	Nein	Ja
XML basiert	Nein	Ja
Einbetten von fremden XML-Namespaces	Nicht direkt, es gibt aber zusätzl. XML-Objekte u. XML-sockets	Ja

¹¹ VML ist ein Acronym für »Vektor Markup Language«

¹² ActiveX ist Microsofts Konkurrenz zu Java: ausführbarer, komponentenbasierter Code innerhalb des Webbrowsers und des Betriebssystems. Vorteil von ActiveX ist die höhere Ausführungsgeschwindigkeit, Nachteil das wesentlich höhere Sicherheitsrisiko. ActiveX Komponenten haben die vollen Rechte, die der Benutzer auf Betriebssystem-Ebene hat.

¹³ Java2D ist eine 2D Graphik-Bibliothek mit grossem Sprachumfang und hochwertiger Rendering-Qualität. Java2D wurde v.a. von Adobe und Sun entwickelt.

¹⁴ Als »rendern« bezeichnet man die Umwandlung einer objekt-basierten Graphikbeschreibung in ein Bild.

¹⁵ SVG ist ein Acronym für »Scalable Vector Graphics«

¹⁶ SVG File-Format Spec: <http://www.w3.org/TR/SVG/index.html>, Macromedia File-Format Spec.: <http://www.macromedia.com/software/flash/open/licensing/fileformat/>

Eigenschaft	SWF	SVG
Einbetten eigener Attribute	Ja, eigene Tag-Types f. Entwickler reserviert	Ja
Verwenden v. Metadaten, RDF	Ja, nur proprietär	Ja, RDF kompatibel
Binär/Textbasiert	binär	textbasiert
Format kann einfach gelesen und editiert werden	nein, da binär	ja, da XML-basiert
Accessibility (für Behinderte)	?	ja, siehe unter ¹⁷
Filegrösse	z.T. etw. kompakter als SVG, da binär	rel. kompakt, kann zusätzl. gezippt werden
DOM transparent	Hat ein internes DOM, nicht W3C konform	Ja
CSS (Stylesheet) und XSL kompatibel	teilweise, nur proprietäre styles	Ja
XSLT Transformationen	Nein	Ja
Kann v. Suchmaschine indiziert werden	Nein	Ja
Verwenden von Entities	Nein	Ja
Syntax und Validitätprüfung	Nein	Ja, DTD, Schema, vom Plugin abhängig
Basisfunktionalität des Plugins		
Braucht Plugin	Ja	Ja, derzeit noch
Pluggingrösse	klein, ca. 200k	gross, ca. 3 MB
Zoomen und Pannen	Ja	Ja
Anhalten von Animationen	Ja	Ja
Antialiasing	Ja	Ja
Suchen nach Textstrings	Nein	Ja
Quelltext einsehen	Nein	Ja
Datenschutz, Verschlüsselung	Nein, jedoch etw. schwieriger zu konvertieren als SVG	Nein, ev. Abhilfe durch XML Encryption oder externe Reader
Einsatzmöglichkeit auf Handhelds	Ja, limitiert	Ja, limitiert, siehe ¹⁸
Graphik-Primitive, Fill, Stroke, ...		

¹⁷ <http://www.w3.org/TR/SVG-access/> (features) und <http://www.w3.org/TR/SVG/access.html> (Richtlinien)

¹⁸ Es gibt eine W3C Arbeitsgruppe für SVG für mobile Devices – Informationen werden auf W3C-Seite publiziert. Der Status ist ähnlich experimentell wie Flash für Embedded Devices.

Eigenschaft	SWF	SVG
Graphische Primitive	nur Path-Element, inkl. quadratische Bezierkurven.	Path-Element, inkl. quadrat. u. cubische Bezier- und elliptische Kurven, Rechtecke, Kreise, Ellipsen, Linien, Polylinien, Polygone
Erstellen eigener Graphik-Primitive	Nein	Ja, mit DEFS und XLINK, eigenen Symbolen und Entities
Füllungen	Solid Fill, Farbverläufe, Bitmapfüllungen	Solid Fill, Farbverläufe, Bitmapfüllungen, Vektorfüllungen
Transparenz	Ja, Füllungen und Striche	Ja, Füllungen und Striche
Linientypen	nur Strichstärke und Farbe	Strichstärke, Farbe, Strichlierungen, Marker, Pfeile, versch. Linienenden, transparente Linien
Einbetten v. Farbprofilen	Ja, nur proprietär	Ja, ICC/SRG
Graphikformate	Ja, Füllung, Linie, Text – proprietär	Ja, Füllung, Linie, Text, CSS und XSL kompatibel
Linienstyles für jedes Liniensegment extra	Ja	Nein
Marker	Nein	Ja
Symbole	Nein	Ja
Verwenden v. Rasterbildern	Ja, eingebettet	Ja, eingebettet (i.d. »data-section«) u. verknüpft, gif, jpg, png
Verwenden von Filtern	Nein	Ja, Raster u. Vektor
Gruppierungen, Layer	Ja, Layer	Ja, Gruppen
Koordinatensysteme, Transformationen		
Separates Device und Viewport-Koordinatensystem (ermöglicht Verwendung v. Echtkoordinaten)	Nein	Ja

Eigenschaft	SWF	SVG
Transformationen	Translation, Rotation, Skalierung, Skewing, Matrix	Translation, Rotation, Skalierung, Skewing, Matrix, Spiegelung
Verschachtelte Transformationen	Nein	Ja
Verschiedene Längeneinheiten	Nein	Ja (mm, px, pt, pc, in, em, etc.)
Clipping, Masking, Compositing	Ja, Clipping und Masking, nur gesamte Layer	Clipping, Masking, Compositing, praktisch alle Elemente können geclippt werden od. als clipping-path dienen
Text		
Einbetten von Fonts	Ja, nur einzelne Glyphen	Ja, einzelne Glyphen und ganze Schriftarten mit Hilfe von W3C WebFonts
Kann Text weiterverwendet werden? (selektiert, kopiert, gesprochen, indiziert, etc.)	Nein?	Ja
Text Kerning	Nein	nur mit Adobe CEF-Files
Rotation von Einzelglyphen	Nein	Ja
Text von rechts nach links, resp. oben nach unten	Nein	Ja
Text entlang von Pfaden	Nein	Ja
Animation, Interaktivität		
Sound	Ja, eingebettet	Nur Teil der SMIL-Spezifikation, kann in Adobe SVG-Viewer verwendet werden.
Scriptingsprache, Programmierung	proprietär, Macromedia Actionscript (ähnlich ECMA-Script) – kann gemixt werden mit ECMA-Script	ECMA-Script, Java Bindings, ActiveX control
Verzweigungen, Fallunterscheidungen im Fileformat	Nein	Ja (Switches und required features check)
Hinzufügen von neuen Objekten via Script	Ja	Ja

Eigenschaft	SWF	SVG
Veränderung des Szenegraphen zur Laufzeit	Ja	Ja
Identifizier für Graphikobjekt	Ja	Ja
Animation	Ja, zeitlinienbasiert, framebasiert	Ja, SMIL kompatibel: zeitlinienbasiert/framebasiert u. interpolationsbasiert
Hyperlinking	eingeschränkt, mit ActionGetURL	voll XLINK kompatibel, innerhalb SVG, auf Viewports und zu externen Entities
Elemente die animiert werden können	Farbe, Position, Transformation, Opazität, Geometrie, Position entlang von Pfaden	Farbe, Position, Transformation, Opazität, Geometrie, Position und Orientierung entlang von Pfaden
Morping von Elementen	Ja	Ja
Unterstützte Events	Mouse-Events, Keyboard Events, Frame/Timeline-Events, onload/onunload, getbyte-events, etc.	Mouse-Events, Time-Events (SMIL), DOMTree-Events, Scroll/Zoom/Resize, onload/onunload; siehe auch ¹⁹
Benutzerdefinierte Cursor	Ja	Ja, siehe ²⁰
Workflow, Konvertierung		
Export aus Graphiksoftware	Macromedia Freehand, CorelDraw, Adobe Illustrator, Macromedia Flash, Adobe Livemotion, u.v.m.	CorelDraw, Adobe Illustrator, Sketch, Kontour, Mayura Draw, Sphinx, Jasc WebDraw, u.v.m.
Export aus GIS-Systemen	Nein	SiCAD, GE Smallworld
Datenbankanbindung	Ja, Macromedia Generator, PerlDBI, Servlets, ASP, FlashSQL, etc.	Ja, PerlDBI, Servlets, ASP, etc.

¹⁹ Keyboard-Events sind derz. nicht Teil der SVG-Spezifikation. Adobe SVG-Viewer hat diese jedoch als Extension implementiert.

²⁰ war während des Schreibens des Artikels noch nicht implementiert in Adobe SVG-Viewer.

Eigenschaft	SWF	SVG
Konvertierungsskripte aus anderen Graphik- und GIS-Formaten vorhanden	Ja	Ja
Schreiben von Konvertern	schwierig, da binär, jedoch SDK zur Verfügung	einfach, Vielzahl an XML Parsern erhältlich, textbasiert

Es ist zu beachten, dass diese Tabelle laufend aktualisiert und korrigiert wird. Für die aktuelle Version siehe http://www.carto.net/papers/svg/comparison_flash_svg.html.

Obwohl Macromedia Flash sehr viele Gemeinsamkeiten mit SVG aufweist und bereits sehr viel länger für Web-Entwickler zur Verfügung steht, ist SVG für die interaktive Kartographie das wesentlich besser geeignete Datenformat. Dies gilt insbesondere für die Konvertierung aus GIS-Daten, XML und Datenbanken. Beide Formate haben jedoch das Manko, nicht primär für GIS und Kartographie entwickelt worden zu sein. Deshalb fehlen Eigenschaften wie Topologie und Projektionsangaben. Einige Mankos können durch das Einführen eigener XML-Tags wettgemacht werden. SVG kann aber auch als »Rendering-Engine« für XML-basierte GIS-Formate wie GML²¹ [OPENGIS, 2001] verwendet werden. Transformationen von GML zu SVG können via Skripte, XSLT und XSLFO, sowie unter Zuhilfenahme von XML-Parsern erfolgen. Beide Formate, SWF und SVG, lassen bezüglich Interaktivität und Animation kaum Wünsche offen.

Die Autoren, aber auch immer mehr Kartographen und GIS-Entwickler haben sich für SVG entschieden, da sie Open-Source, offizielle Standards und die solide, zukunftsweisende XML-Basis schätzen. Auch durch den Einsatz von SVG werden nicht alle technologischen Problempunkte, wie z.B. der Datenschutz, geklärt. SVG kommt jedoch einer offenen und technologisch transparenten, leicht erweiterbaren Architektur wesentlich näher als die Macromedia Flash-Technologie. Zudem ist SVG ein Gemeinschaftsprojekt aller bekannten Hard- und Softwarehersteller. Support und schnelle Weiterentwicklung des Standards und der Werkzeuge sind daher garantiert. Insbesondere im Open-Source Bereich und im Austausch zwischen verschiedenen Graphik-Programmen wird SVG eine zunehmend wichtigere Rolle spielen. Die neue OpenGIS²² Infrastruktur baut zunehmend auf XML und sieht SVG und X3D/VRML2000²³ als »Rendering-Backend«. Die W3C Standards bilden ein ineinander verzahntes XML-Framework (z.B. XLINK, SCHEMA, XPOINTER, Namespaces, etc.) – ein Einsatz von Flash in diesem Bereich würde also in eine technologische Sackgasse führen. Zu »Workflow«-Aspekten für den Einsatz von SVG siehe [NEUMANN/WINTER, 2001].

²¹ GML (Geography Markup Language) ist ein vom OpenGIS-Konsortium definiertes XML-basiertes GIS-Daten Austauschformat

²² Das OpenGIS-Konsortium ist eine Gruppe bestehend aus GIS-Firmen und Hochschulen, die sich für die Standardisierung von GIS-Technologien und GIS-Datenformaten einsetzt.

²³ VRML und X3D sind 3D-Graphikformate – sie werden in Kapitel 6 eingehender erklärt.

6 3D-Formate für das Internet

Bei den 3D-Formaten kam es infolge der relativ frühen Einführung des VRML-Standards²⁴ nicht zu einem derartigen »Wildwuchs« wie bei den 2D Vektorgraphik-Formaten. VRML 1.0 wurde bereits 1995/96 als weitgehend an OpenInventor²⁵ angelehntes, szenegraph-basiertes Graphikformat eingeführt. 1997 folgte die Version 2.0 oder VRML97 als ISO/IEC Standard. Neu war dabei v.a. die Definition von Sensoren (interaktive Elemente), Interpolatoren und Animation, sowie die Definition des sog. »External Authoring Interface« (EAI) als Schnittstelle eines VRML-Plugins zur »Aussenwelt«.

VRML97, der heute immer noch gängige VR-Standard fürs Web, stellt folgende Knotentypen²⁶ zur Verfügung:

- Geometrie-Knoten
 - Regelmässige Geometrie: Box, Kugel, Kegel, Zylinder
 - Unregelmässige Geometrie: Punkte, Linien, Indexed-Faces, Extrusion, Elevation-Grids
- Material-Knoten: Farben, Glanz, Texturen
- Lichtquellen: Punktlicht, Spot-Licht, Direkt. Licht, Nebel
- Sensoren: Touch-Sensor, Timer, Visibility-Sensor, Plane-Sensor, Cylinder-Sensor, Sphere-Sensor, Proximity-Sensor, Collision-Detector
- Multimedia-Knoten: Text, Image, Movie, Sound, Billboards, Hyperlink-Knoten
- Interpolatoren: Farb-Interpolator, Koordinaten-Interpolator, Normalvektoren-Interpolator, Positions-Interpolator, Orientierungs-Interpolator, Scalar-Interpolator
- Kontrollstrukturen: Skript-Knoten, Switch-Knoten, Levels of Detail
- Zusätzliche Knoten: Info-Knoten, Gruppen-Knoten, Transform-Knoten, Prototypen

Durch die Interaktionsmöglichkeiten (Sensoren) und Kontrollstrukturen können analog zu SVG/Flash (2D) hoch-interaktive und datenbankgestützte 3D-Welten generiert werden. Einige Hersteller von VRML-Plugins arbeiten zudem an sog. »Multiuser-Browsern«, mittels derer die Benutzer in einer gemeinsamen, vernetzten virtuellen Welt agieren können.

Beim Arbeiten mit EAI (External Authoring Interface) – der Einsatz von EAI ist für komplexere Welten unabdingbar – muss jedoch häufig mit Stabilitätsproblemen gerechnet werden. Die Implementationen sind zudem stark browserabhängig. Hinzu kommt, dass viele Graphikkarten bei komplexeren Welten überfordert sind. Schliesslich ist VRML97 weder XML- noch DOM-basiert. Die Web3D Working-Group arbeitet deshalb an der Weiterentwicklung von VRML, dem sogenannten VRML200x/X3D Standard. Er wird neu auch XML und DOM-basiert sein. Die Neuerungen gegenüber VRML 97 sind:

²⁴ VRML, ein Acronym für »Virtual Reality Modeling Language«, ist eine 3D Beschreibungssprache, die neben der 3D Geometrie auch Interaktionen beschreiben kann.

²⁵ OpenInventor ist eine von SGI entwickelte »High-Level« 3D-Graphik-Schnittstelle.

²⁶ Ein VRML-Knoten ist eine Entität des gesamten Szenegraphen, also z.B. eine Geometrie oder Materialbeschreibung

- Funktionsplitting in sog. Profile (erlaubt konforme VRML-Browser mit unterschiedl. Funktionsumfang)
- Konsistenteres Object-Model, DOM
- Unicode-Support
- Neue Syntax (X3D)
- Verbessertes Event-Model
- Verbesserter Script-Support
- GeoVRML support für geographische Daten
- Support für Surfaces
- Support für Nurbs
- Verbessertes Zeitmodell
- Streaming-Support
- Binary-Support

Ob der neue Standard Erfolg haben wird, hängt in erster Linie von den Plugin-Entwicklern und der Qualität der Integration in bestehende Webbrowserprojekte ab.

7 Beispiele für kartographische Anwendungen aktueller Webtechniken

Am Institut für Kartographie der ETH Zürich werden laufend Studentarbeiten (Seminararbeiten und Diplomarbeiten) im Bereich Web-Kartographie durchgeführt. Ein grosses Anliegen ist es, proprietäre Standards zu vermeiden. Deshalb hat sich das Institut auf der Client-Seite auf die Standards SVG und VRML, in Kombination mit Javascript und Java, konzentriert. Serverseitig kommen häufig Perl/CGI/PHP und SQL-Datenbanken zum Einsatz. Nachfolgend sollen ein paar Beispiele solcher Arbeiten vorgestellt werden.

7.1 Beispiele aus der Thematischen Kartographie

7.1.1 Social Patterns in Vienna

Bei diesem Beispiel handelt es sich unseres Wissens um die erste interaktive Webkarte, die mit der Kombination SVG/HTML/CSS/Javascript erstellt wurde. Mit Hilfe von Choroplethen-Darstellungen werden sozio-ökonomische Strukturen der Stadt Wien visualisiert. Der Benutzer kann dabei eine Variable auswählen, die Anzahl der Klassen nach der Methode der Quantile festlegen, Ebenen ein/ausblenden, zoomen und pannen (Verknüpfung mit einer kleinen Übersichtskarte) und mit »Mouse-Over-Effekten« Werte und Namen abfragen. Ein Mausklick auf eine administrative Einheit öffnet ein kleines Info-Fenster mit detaillierteren Daten. Es lassen sich damit effizient räumliche Verteilungsmuster (Segregationen) visualisieren. Das Projekt wurde bereits Anfang 2000, kurz nach dem Erscheinen des ersten SVG-Plugins (Adobe SVG Viewer 1.0) implementiert, und hat viele weitere kartographische SVG-Projekte am Institut und extern inspiriert. URL: <http://www.karto.ethz.ch/neumann/cartography/vienna/>

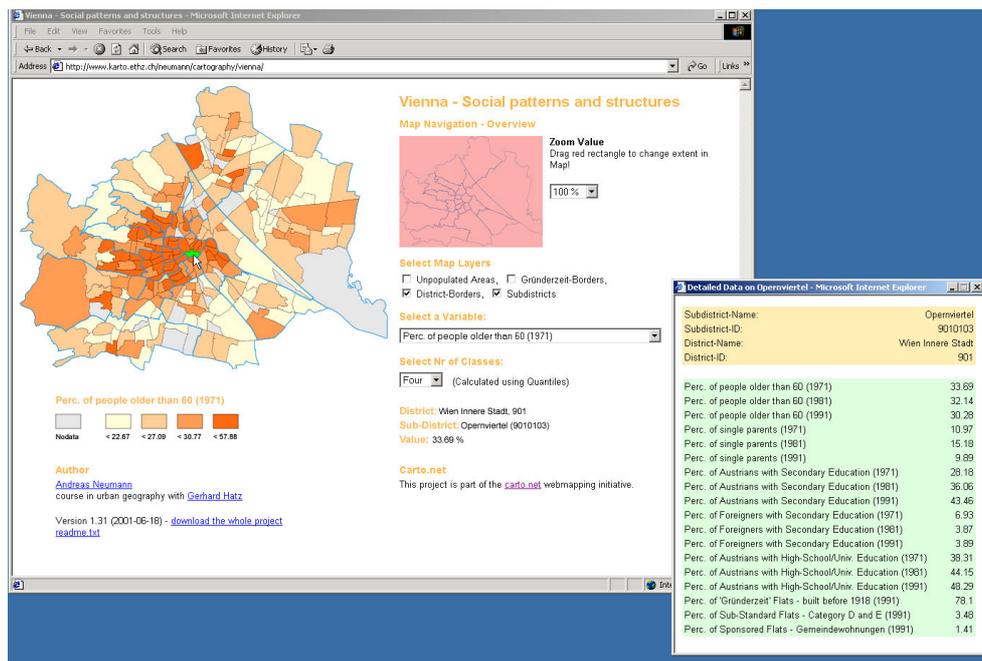


Abbildung 1: SVG Anwendung »Social Patterns in Vienna«

7.1.2 Prototyp eines interaktiven Europa-Atlas

Sehr bald nach dem SVG-Beispiel »Social Patterns« folgte eine Diplomarbeit von Andréas M. Winter, die gemeinsam von der Universität Wien und der ETH Zürich betreut wurde. Mit dem Prototyp wurden erstmals Diagramme dynamisch aus den statistischen Daten generiert. Der Benutzer kann Themen wählen (nicht alle sind implementiert!), Ebenen ein- und ausblenden und die Diagrammparameter einstellen. Dabei können Farben ausgewählt, die Diagrammgröße angepasst, sowie der Diagrammtyp (Kreis- und Flügeldiagramme) gewählt werden. Zusätzlich können sortierbare Tabellen aus den Originaldaten generiert werden. Besonders am Beispiel ist auch, dass auf die Diagramme Filter angewendet werden können, in diesem Fall ein Schatteneffekt, welcher eine bessere Unterscheidung der Diagramme ermöglicht. URL: <http://www.carto.net/papers/svg/eu/>

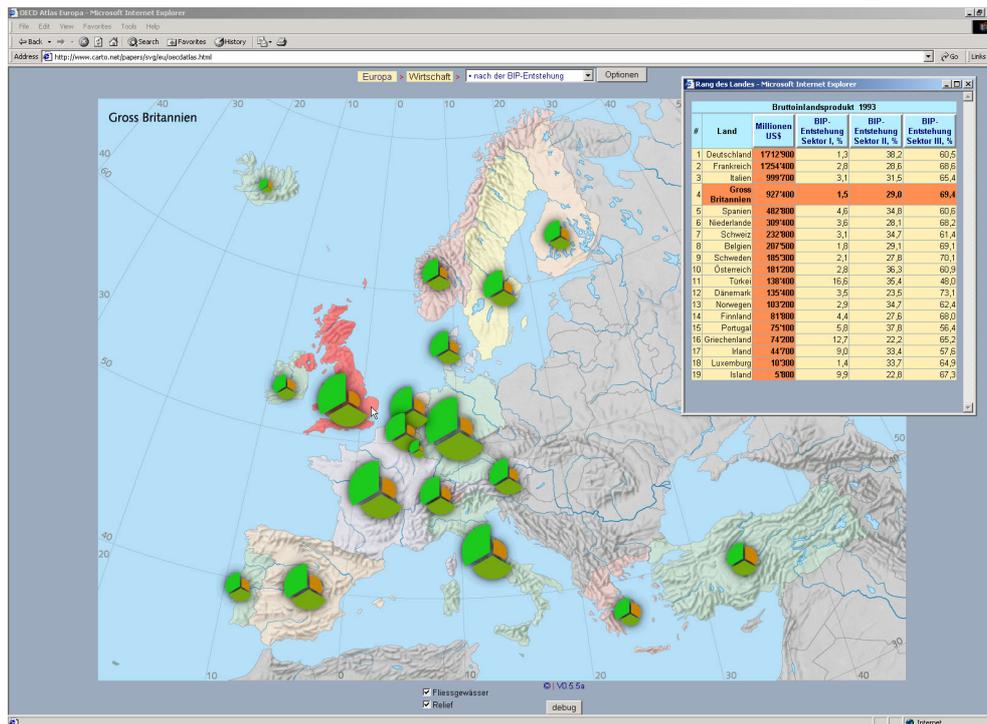


Abbildung 2: SVG Anwendung Prototyp »OECD Atlas Europa«

7.1.3 Migration in Europa

Eine weitere Diplomarbeit (Christian Joray) beschäftigte sich mit der Migration zwischen der Schweiz und dem restlichen Europa. Dabei können der Zeitraum sowie die Wanderungsrichtung (Immigration/Emigration), das Geschlecht, die Anzahl Klassen, die Art der Klassenberechnung (Quantile, gleichabständig und manuell) und eine vordefinierte Farbpalette gewählt werden. Die Karten werden aktiviert, indem man in eine Schweizer Region oder ein Land Europas klickt. Als weitere Features gibt es wiederum Datentabellen, Kreisdiagramme zur geschlechtsspezifischen Verteilung und eine einfache Animationsfunktion. Eine automatisch generierte Verlaufskurve zeigt den Migrationsstrom über die verschiedenen Jahre hinweg. URL: http://www.karto.ethz.ch/teaching/dip_2001_joray/

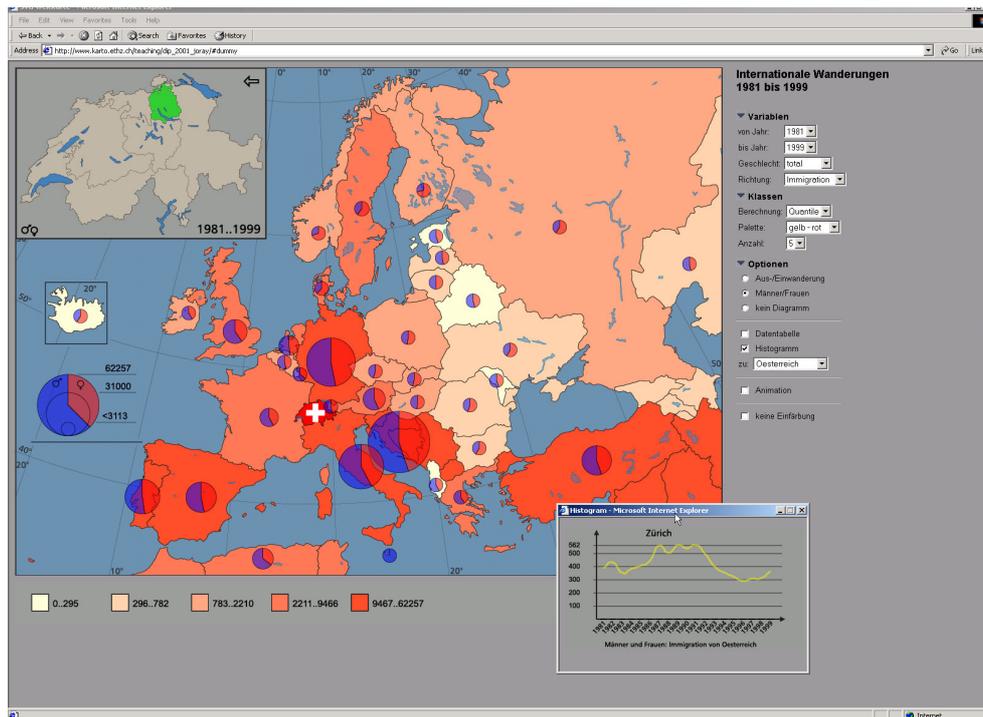


Abbildung 3: SVG Anwendung »Migration zwischen der Schweiz und Europa«

7.1.4 Taxes in Zurich, Schwyz and Zug

Diese Semesterarbeit (Raphael Hilber und Stephan Ziegler) stellt die Steuersituation in den Kantonen Zürich, Schwyz und Zug dar. Neben den üblichen Klassierungs- und Navigationsmöglichkeiten kann man hier die administrative Bezugseinheit, sowie eine Anzahl verschiedener steuer-relevanter Variablen wählen. Vor allem können aber einfache räumliche Abfragen durchgeführt werden. So kann man etwa nach einem Gemeinde-Namen suchen oder nach Attributwerten selektieren. Die Abfragen können auch kombiniert werden. Als Besonderheit können Kreisdiagramme »on mouse-over« ausserhalb der Karte dynamisch generiert werden, um das Kartenbild nicht zu stark zu belasten. URL: http://www.karto.ethz.ch/teaching/vtb_2001_hilber_ziegler/ (läuft allerdings nur unter Internet-Explorer)

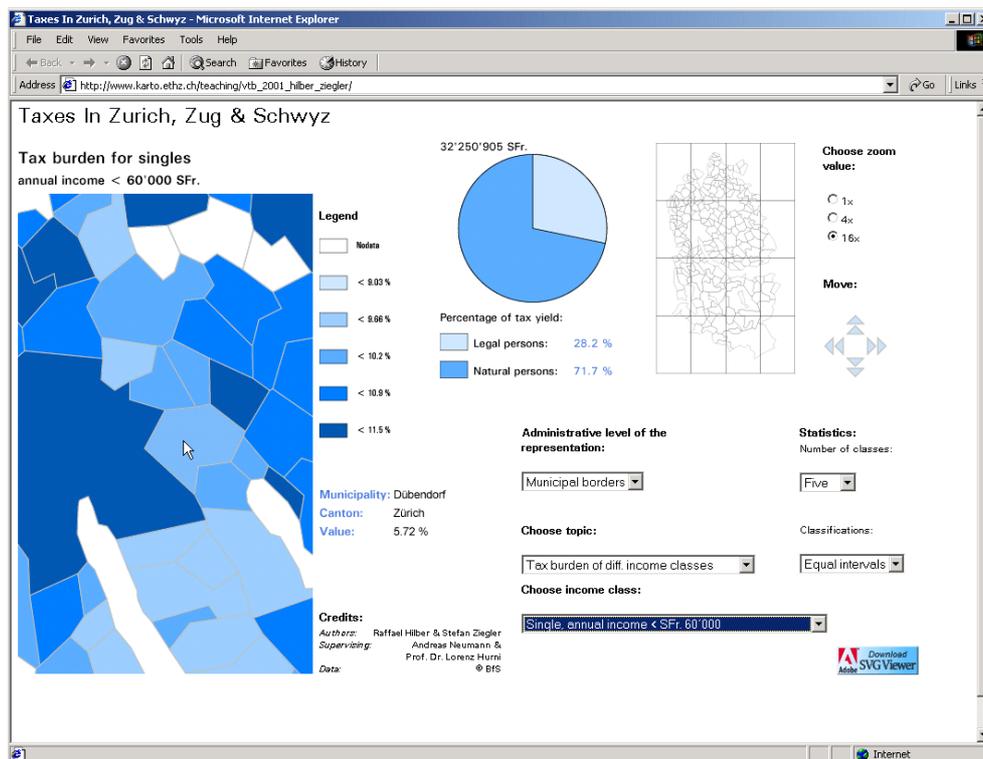


Abbildung 4: SVG Anwendung »Taxes in Zurich, Zug and Schwyz«

7.2 Beispiele aus der topographischen (Gebirgs-)kartographie

7.2.1 SVG und VRML im Einsatz für Höhleninformationssysteme

Ziel einer Gruppe von Höhlenforschern (Kartographen und Geoinformatiker) aus der Schweiz ist es Instrumente und für interaktive und vernetzte Höhleninformationssysteme zu erstellen. SVG und VRML, die beiden oben beschriebenen Standards, bilden dabei wesentliche Elemente zur Plandarstellung resp. 3D-Visualisierung. SVG dient dabei zur verlustfreien Darstellung der 2D Höhlenpläne und als kartographische Basis zur Verortung von Beobachtungen, Fundstücken und Probennahmen, Fotos, verbalen Beschreibungen, etc. VRML kann diese Beobachtungen zusätzlich in der dritten Dimension darstellen. Beide Techniken können mit interaktiven Elementen versehen werden, die dem Benutzer die Navigation erleichtern und ihm ermöglichen, das Aussehen einzelner Elemente zu gestalten. Objekte können ein/ausgeblendet, transformiert und mit weiteren Informationen verknüpft werden. Offene Standards lassen sich zudem sehr gut mit Datenbanken integrieren. Die Arbeiten stützen sich auf die Zusammenarbeit mit Martin Heller, dem Entwickler der Höhlenvermessungssoftware »Toporobot« (<http://www.toporobot.net/>). Sie finden einen Prototypen des Höhleninformationssystems unter <http://www.karto.ethz.ch/neumann/caving/cis/> (Achtung: läuft nur unter Netscape und EAI-fähigen VRML Browsern). Zudem gibt es viele weitere Informationen zur

»virtuellen Höhlenforschung« und Höhlendokumentation unter und
<http://www.karto.ethz.ch/neumann/> (caving section). Vergl. dazu auch
 [HELLER/NEUMANN, 2001].

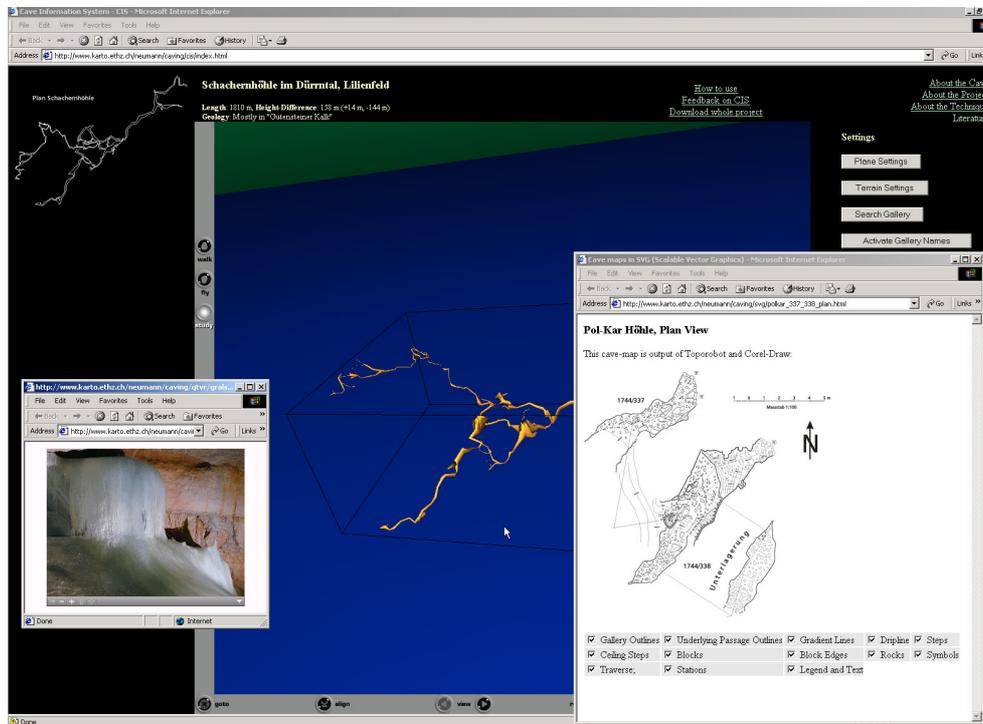


Abbildung 5: Höhleninformationssystem kombiniert mit VRML/SVG und Quicktime

7.2.2 Hochwasserinformationssystem Locarno/Lago Maggiore

Andrea Terribilini und Bernhard Jenny, ebenfalls Mitarbeiter am Institut für Kartographie der ETH Zürich, haben einen Prototyp eines Hochwasserinformationssystems auf SVG-Basis für die Stadt Locarno am Lago Maggiore entwickelt. Dabei kann die Überflutung mit verschiedenen Pegelständen in Ein-Meter-Schritten simuliert werden. Die überfluteten Flächen liegen transparent über dem jeweiligen Stadtplan. Die betroffenen Strassen und Häuser können aufgelistet werden. Gleichzeitig kann man nach Häusern und Strassen suchen. Die Stadtverwaltung könnte ein derartiges System als Zivilschutzinformationssystem einsetzen.

7.2.3 VRML Visualisierung einer archäologischen Ausgrabungsstätte: Palairos/Griechenland

Diese Semesterarbeit von Michael Boller, Martin Fopp und Uta Samaga hatte die dreidimensionale und interaktive Darstellung einer archäologischen Ausgrabungsstätte in Griechenland zur Aufgabe. Die Geometrie wurde vom »Deutschen Institut für Archäologie in Berlin« zur Verfügung gestellt. Sie wurde in 3D-Studio-Max

weiterbearbeitet. Informationen werden mit Hilfe von Touch-Sensoren und Scripten verknüpft. Der Benutzer kann sich im Gelände frei bewegen, oder an einem vordefinierten Rundflug teilnehmen. Die Elemente der einzelnen Zeitepochen können ein/ausgeblendet und hervorgehoben werden. URL: http://www.karto.ethz.ch/teaching/vtb_2001_boller_fopp/ (derzeit nur unter Internet-Explorer und Cortona VRML plugin)

8 Zusammenfassung und Ausblick

Die obigen Beispiele zeigen sehr gut, dass es sich trotz höherem Lernaufwand lohnt, auf offene Standards zu setzen und den Studenten Grundlagenwissen in Computergraphik beizubringen. Dieses Basiswissen wird von sog. Autorensystemen, die oft auf einer abstrakteren Ebene ansetzen, häufig verborgen. Die Unterrichtspraxis hat gezeigt, dass die Studierenden Spass an der Entwicklung interaktiver kartographischer Webseiten haben, und daher auch bereit sind, sich in technologische Neuerungen einzuarbeiten, also in Bereiche, die auch oft für die Assistierenden neu sind. In diesem interessanten Prozess, in dem sowohl Betreuer als auch Studierende gegenseitig geben und nehmen, sind die Strukturen also nicht von vornherein als »Einbahnstrasse« vorgegeben. Viele der Projekte sind jedoch Prototypen, die aus Zeitgründen nicht auf alle Browsertypen angepasst werden konnten und sicherlich noch die einen oder anderen Mankos aufweisen.

Die vielen positiven Erfahrungen mit SVG haben gezeigt, dass es möglich ist, mit verhältnismässig geringem Aufwand interaktive Graphiken in einer flexiblen und standardisierten Architektur ins WWW zu bringen. Zukünftige Arbeiten werden sich dem VRML97 Nachfolger VRML200x/X3D widmen. Dies sind 3D-Standards, welche für 3D-Projekte ähnliche Möglichkeiten wie SVG im 2D-Bereich versprechen. Zukünftige Projekte werden sich vermehrt mit interaktiven topographischen Kartenprojekten beschäftigen. Zudem sollen mit serverseitigen Techniken und Frameworks auch grössere Datenmengen effizient im Web publiziert werden können. Schliesslich sollten die Web-Projekte in ein umfassenderes »Cross-Media-Publishing-System« integriert werden, das sowohl gedruckte Karten, CD-ROMs, als auch Webseiten und mobile Ausgabegeräte unterstützen kann. Voraussetzung dafür ist ein offenes und erweiterbares Datenmodell, ein modulares System an kartographischen Operationen, sowie ein Set von intelligenten Konvertierungsroutinen, um verschiedene Datenformate generieren zu können.

9 Online-Ressourcen zu SVG/VRML/X3D

SVG

<http://www.w3.org/Graphics/SVG/Overview.htm#8> – SVG Spezifikation, Implementationen, News, SVG Testsuite.

<http://www.adobe.com/svg/> – Adobe SVG Viewer, Tutorials, Beispiele, Online-Forum.

<http://groups.yahoo.com/groups/svg-developers/> – Mailingliste/Online-Forum.

<http://www.carto.net/papers/svg/> – Artikel, Beispiele, Tutorials, Mailingliste.

<http://www.kevlindev.com> – Gute SVG Beispiele.

VRML/X3D

<http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm> – VRML97 Specification

<http://www.web3d.org/TaskGroups/x3d/specification/> – VRML 200x/X3D specification

<http://www.blaxxun.com/> – VRML plugin, VRML services.

<http://www.parallelgraphics.com> – VRML plugin, VRML services.

<http://www.openvrml.org> – Open Source VRML development.

<http://www.cai.com/cosmo/> – Good Old Cosmo-Player (nicht mehr in Entwicklung)

10 Literatur

FREEGIS.org (2001): »Free GIS Software and Free Geo-Data«, <http://www.freegis.org/>, Osnabrück, 2001.

HELLER, Martin und Andreas NEUMANN (2001): »Innere Cartographie – From Surveying towards Information Systems«, Proceedings for the 20th International Cartographic Conference, Beijing, 2001.

MUENZ, Stefan (2001): »Interview mit Stefan Münz, Autor des Tutorials SELFHTML«, <http://www.sagmal.de/muenz.htm>, München, 2001.

NEUMANN, Andreas und Andréas M. WINTER (2001): »Time for SVG – Towards high quality interactive Web-Maps«, Proceedings for the 20th International Cartographic Conference, Beijing, 2001.

OPENGIS-Konsortium (2001): »Geography Markup Language, v2.0 Geography Markup Language (GML) 2.0«, <http://www.opengis.net/gml/01-029/GML2.html>, Wayland, MA, USA, 2001.

WEB3D-Konsortium (2001): »VRML200x Specification, ISO/IEC 14772«, <http://www.web3d.org/TaskGroups/x3d/specification/index.html>,

WINTER, Andréas u. Andreas NEUMANN (2000): »Kartographie im Internet auf Vektorbasis, mit Hilfe von SVG nun möglich«, http://www.carto.net/papers/svg/index_d.html, Zürich/Wien, 2000.