

# SVG für die Webkartographie - Aktuelles und Zukünftiges

Georg Held, Andreas Neumann, Nicole Ueberschär, André M. Winter

<http://www.carto.net>

## 1 Wo steht SVG?

SVG steht für "Scalable Vector Graphics" und ist ein XML basierter "Webstandard" der 2D Vektorgraphik, Rastergraphik, Text, Animation, Scripting und Multimedia vereint. Ausführlich vorgestellt wurde SVG unter anderem auf den vorangegangenen Webmapping Symposien. Aufgrund der stetigen Veränderungen in der Informationstechnologie soll diese Ausarbeitung die wichtigsten Neuerungen sowie mögliche Szenarien und zukünftige Entwicklungen aufzeigen. Alle Beispiele und diese Dokumentation sind zu finden unter URL[01], die Webseite der Arbeitsgruppe unter URL[02].

### 1.1 Roadmap

Die erste Version der SVG Spezifikation erschien im September 2001 als eine W3C Recommendation (äquivalent zu Webstandard). Version 1.1 folgte im Januar 2003 und ist die derzeit aktuelle. Im Gegensatz zum Vorgänger gliedert sich diese in verschiedene Profile: SVG Standard (für Desktops), SVG Basic (für PDA's und Smartphones), SVG Tiny (für schwächere Mobiltelefone) und SVG Print (gegenwärtig noch im Entwicklungsstadium).

Gegenwärtig arbeitet die SVG Arbeitsgruppe des W3C (World Wide Web Konsortiums) an der Version 1.2. Der aktuell vorliegende Working Draft lässt viel versprechende neue Features erahnen. Dazu gehören beispielsweise Textwrapping in beliebigen Formen, Streaming SVG, verbesserter Animationssupport, das standardisierte Einbetten und Synchronisieren von Audio und Video, Unterstützung mehrseitiger SVG Dokumente, verbesserte Filter und Füllungen, Tooltips und Feedback zum Downloadstatus einer SVG Datei. Eine detaillierte Auflistung würde den Rahmen hier sprengen. Interessantestes und derzeit viel diskutiertes neues Feature ist das Rendering von beliebigen XML Code Fragmenten. Die Fachwelt spricht dabei von **Rendering Custom Content**. RCC erlaubt das Erstellen wiederverwendbarer interaktiver Graphikelemente innerhalb von SVG, wie das etwa GUI Elemente oder kartographische Symbole sind (vgl. Kapitel 3).

Betrachtet man die neuen Möglichkeiten von SVG 1.2, so wird klar, dass die SVG Arbeitsgruppe in Zukunft verstärkt das Segment interaktiver, webfähiger Multimediaanwendungen ansprechen und zudem SVG als graphisches Frontend für Webservices und Webapplikationen propagieren will. Hier wird SVG vermehrt in Konkurrenz zu proprietären Technologien, wie Macromedia Flash, treten. SVG verspricht Offenheit und deswegen Flexibilität sowie optimale Integration in die "XML-Welt".

Genaue Termine sind aufgrund des zu durchlaufenden Annahmeprozesses der in der Arbeitsgruppe Beteiligten schwer auszumachen. Voraussichtlich wird SVG 1.2 im ersten Quartal 2004 finalisiert und freigegeben. Da namhafte Hersteller wie Adobe und Corel bereits an neuen

Implementierungen arbeiten, werden erste SVG 1.2 konforme Viewer nach der Veröffentlichung des Standards sofort verfügbar sein (vgl. Kapitel 3.3).

## 1.2 Übersicht derzeitiger Implementierungen

Wichtig für die Verbreitung eines Standards ist die Quantität und Qualität der Implementierungen, die ihn unterstützen. Unterschieden wird dabei zwischen Möglichkeiten des Betrachtens, Erstellens und Modifizierens von SVG Dateien. Die folgende Auflistung gegenwärtig wichtiger Implementierungen berücksichtigt im Besonderen auch die Erzeugung von SVG aus gebräuchlichen kartographischen Anwendungen. Das bisher etwas unbekanntere Corel Smart Graphics Studio wurde gesondert unter die Lupe, um dessen Einsatzmöglichkeiten zu erläutern.

### • SVG Viewer

Als Viewer gelten Möglichkeiten, SVG durch Softwareapplikationen auf dem Bildschirm darzustellen. Man unterscheidet zwischen Plug-Ins für Browser und Standalone-Anwendungen. Im Folgenden eine Übersicht der wichtigsten Viewer:

Name	Art
Adobe SVG Viewer 3.01	Plug-In
Corel SVG Viewer 2.1	Plug-In
Squiggle SVG Viewer 1.5 (Batik)	standalone
KSVG	Native Browser Unterstützung (Linux/KDE)
Amaya 8.2	Browser/Graphikeditor
Mozilla SVG	Native Browser Unterstützung

Leider ist hinzuzufügen, dass die viel erhoffte, einsetzbare native Browserunterstützung nicht absehbar ist. Vielmehr gefährdet gerade ein Streit den Fortbestand von Plug-Ins aufgrund verschiedener Patentansprüche. Tendenziell werden – unter Vorbehalt – "Standalone Viewer", die verschiedene XML Namensräume unterstützen, erheblich größeren Anteil gewinnen. Dadurch könnten beispielsweise Standards, wie SVG und XHTML, in einem Dokument vermischt werden.

### • Editierung aus Applikationen

Unter Editierung versteht man die Erzeugung, Weiterverarbeitung und Manipulation von SVG Dateien. SVG Dokumente werden zum einen durch Texteditoren bearbeitet. Andererseits dienen Autorensysteme und Desktop Publishing Anwendungen zur komfortablen Erzeugung mittels einer Graphischen Benutzeroberfläche. Weiter lassen sie es begrenzt zu, SVG Interaktivität zuzuführen. Wichtigste aktuelle Vertreter sind:

Name	Hersteller	Kategorie
Illustrator 10	Adobe	DTP-Software (SVG Export)
Corel Draw 12	Corel	DTP-Software (SVG Export)
MayuraDraw 4.1	Open-Source	Graphikeditor (SVG Export)
e-Picture	Beatware	Graphikeditor (SVG Export)
Sodipodi 0.31	Open-Source	Graphikeditor (nativ)
Amaya 8.2	Open-Source	Graphikeditor/Browser (nativ)
WebDraw 1.02	Jasc	Autorensystem (nativ)
Xstudio 2.0	Evolgrafix	Autorensystem (nativ)

Zu Beachten ist, dass vor allem Macromedia noch immer keine Bemühungen unternimmt, SVG zu unterstützen. Diese Tatsache ist beim Einsatz von Freehand bedauerndswert, da deswegen der "Umweg" über Illustrator oder Corel Draw beschritten werden muss.

#### • **SVG für kartographische Anwendungen**

Im Folgenden aktuelle Werkzeuge zum Erzeugen von SVG aus Geoinformationssystemen oder Mapserver Produkten, die von teilweise kommerziellen Anbietern für verschiedene GIS-Implementierungen entwickelt wurden.

Name	GIS	Kategorie
MapViewSVG	ArcView3+8	GIS-Exportwerkzeug
SVGMapper	ArcView3	GIS-Exportwerkzeug
Shp2svg	ArcView3	GIS-Exportwerkzeug
Map2svg 1.3.3	MapInfo	GIS-Exportwerkzeug
SVGMapMaker	MapInfo	GIS-Exportwerkzeug
MicroImages TNT	TNT	GIS-Export (built-in)
SICAD Internet Suite	SICAD	Serverimplementierung
Map Extreme 4.7	MapXtreme	Serverimplementierung

Die Entwicklung der letzten Jahre zeigt zunehmende Möglichkeiten zur Generierung von SVG direkt aus Geoinformationssystemen. Zudem ermöglichen neue Exportwerkzeug-Generationen verbesserte Interaktivität der resultierenden SVG-Karten, ohne zusätzliches Editieren. Auch die Anstrengungen SVG serverseitig zu erzeugen, nehmen gegenwärtig von Release zu Release zu. Zudem kann durch die im GIS implementierte "Scriptingmanipulation" SVG auf sehr einfache Art und Weise erzeugt werden.

#### • **SVG Publishing mit Corel Smart Graphics Studio**

Corel Smart Graphics Studio ist ein SVG Autorensystem, das das effiziente Erstellen großer SVG Webprojekte erlaubt [URL03]. Die Zielgruppe besteht weniger aus Webdesignern, die primär graphisch orientiert arbeiten, sondern vielmehr aus Entwicklern und Systemarchitekten, die SVG in existierende Arbeitsabläufe integrieren müssen. So lassen sich sehr einfach dynamische Graphiken aus XML Datenquellen und Datenbanken heraus generieren. Das Produkt besteht aus 3 Komponenten: Developer, Process Builder und Server. Im Developer können existierende SVG Graphiken aus anderen Graphiktools (etwa CorelDRAW oder Adobe Illustrator) importiert und dort mit dynamischen Eigenschaften wie Maus- und Keyboard Events versehen werden.

Darüber hinaus stellt die Developer-Komponente zahlreiche User-Interface Komponenten (Buttons, Slider, Auswahllisten, etc.) zur Verfügung. Hier können zudem alle Eigenschaften von SVG-Elementen angepasst werden und neue Graphikprimitive hinzugefügt werden. Die Developer-Komponente erlaubt auch das Anlegen von graphischen Templates. Schließlich geschieht hier auch das Mapping zu XML Datenquellen. Das Programm generiert automatisch XSLT Code, der auf beliebigen Plattformen (inkl. Linux/Unix/MacOSX) publiziert werden kann. Der Process-Builder erlaubt das Verbinden der Datenquellen mit Datenbanken und Webservices (via SOAP). Sämtliche Verbindungen können graphisch eingerichtet werden. Zudem offeriert diese Komponente zahlreiche statistische und mathematische Operationen sowie Analysen, bevor die Daten an die Visualisierung (Developer Komponente) weitergeleitet werden. Der Workflow wird vollständig graphisch festgelegt (mit Icons zu Datenquellen, Operatoren und Datenfiltern, Konnektoren, Logikkomponenten und Datenausgaben). Zudem enthält der Process-Builder zahlreiche Bildverarbeitungsmodule zur Manipulation von Raster und Vektordaten. Dieses Modul scheint sehr vielversprechend und bietet weitreichende Möglichkeiten in der Datenaufbereitung und Manipulation.

Das Resultat kann auf Windows und Java Application Servern publiziert werden, wobei es derzeit den Anschein hat, dass die Windows Welt noch wesentlich besser unterstützt wird. Die Serverkomponente erlaubt zudem auch das "Parsing" von GML Daten und das Konvertieren verschiedener Dateiformate. Komplettiert wird das Smartgraphics Studio durch einen eigenen SVG Viewer, der derzeit vollen Scriptingsupport beinhaltet, jedoch noch keine Patterns, Filter und Animationen unterstützt. Natürlich werden der Adobe SVG Viewer und andere standard-kompatible Viewer ebenso unterstützt.

Insgesamt scheint das Corel Smartgraphics Studio eine kostengünstige Alternative zu recht kostspieligen bestehenden Konkurrenzprodukten (wie etwa der Adobe Graphics Server). Fehlende Funktionen im Process Builder und in der Serverkomponente werden auf Wunsch relativ rasch bereitgestellt. Leider setzt das Produkt bisher lediglich auf graphische Administrierbarkeit und unterstützt im Moment noch kein JavaScript Debugging oder das direkte Editieren von Quellcode. Ein Manko, das insbesondere bei der Erstellung von interaktiven clientseitigen Webmappingapplikationen relativ schwerwiegend ist. Die serverseitige Anbindung, inkl. der Unterstützung von Webservices und Datenbanken, sowie die Flexibilität in der Datenaufbereitung und Datenmanipulation scheint jedoch sehr gelungen und viel versprechend. Das Tool wird derzeit primär von größeren Firmen und erfahreneren Entwicklern für größere Projekte eingesetzt, da die Einarbeitungszeit für designorientierte Webentwickler im Moment noch relativ hoch ist. Ohne zumindest grundlegende Kenntnisse von Webstandards und -architekturen ist das Produkt nicht zu empfehlen.

Abschließend aus aktuellem Anlass ist zu erwähnen, dass die Firmen Corel und Oracle eine Kooperation angekündigt haben. Dabei sollen die Datenbank Oracle10g und Corel Smart

Graphics zusammen als SVG Lösung auftreten URL[04].

Eine Übersicht zu weiteren Implementierungen mit Links sind auf der "Implementation Page" des W3C's URL[05] und im SVG-Wiki URL[06] zu finden.

### **1.3 SVG.Open 2003**

Vom 13.-18. Juli 2003 fand in Vancouver die zweite SVG.Open Developers Conference statt. Sie dient als eine Diskussions- und Präsentationsplattform für SVG-Anwender, -Entwickler, das W3C Konsortium und Firmen. Ihren Ursprung hat die Konferenz im Juli 2002, wo sie das erste Mal am kartographischen Institut in Zürich stattfand. In Vancouver nahmen mit ca. 250 Teilnehmern aus ca. 30 Ländern ein wenig mehr Leute als in Zürich teil. Trotz dem relativ geringen Zuwachs an Teilnehmern, war das Programm mit ca. 100 Vorträgen äußerst reichhaltig.

Zahlreiche Beiträge zeigten SVG Applikationen aus den Bereichen Business Graphics, wissenschaftliche Visualisierung, Multimedia, SVG Mobile, E-Learning, GIS und Kartographie. Wie auch im Jahr zuvor waren Kartographen und GIS-Teilnehmer mit Präsentationen stark vertreten. Im Fachpublikum befanden sich zudem einige Experten aus öffentlichen Ämtern (z.B. Statistik und Landesvermessung), welche SVG teilweise bereits erfolgreich einsetzen oder deren zukünftigen Einsatz evaluieren. Die große Anzahl von Webmapping Interessierten gab den Anstoß zu einer öffentlichen Podiumsdiskussion. Während dieser wurden fehlende und ungenügende Features angesprochen, deren Berücksichtigung in zukünftigen SVG Spezifikationen und Implementierungen Kartographen und GIS-Spezialisten das Leben erleichtern sollen. Die Kartographie hat aufgrund ihres frühen Engagements im SVG-Bereich nicht unbeträchtlichen Einfluss auf die Weiterentwicklung von SVG. Dazu müssen Kartographen jedoch bereit sein, ihre Bedürfnisse klar und deutlich gegenüber der W3C Arbeitsgruppe und den Herstellern zu artikulieren. Deswegen verfassten Kartographen aus verschiedenen Ländern eine "Wishlist", die Ergebnisse der Diskussion zusammenfassen URL[07].

Sehr positiv viel auf, dass wesentlich mehr Sponsoren und Aussteller gewonnen wurden. Fast die ganze Palette an namhaften Herstellern von SVG Implementierungen war vertreten. Außerdem waren spürbare Fortschritte im Bereich SVG Mobile zu sehen. So stellte Nokia erste Mobiltelefone mit SVG Support vor. Firmen wie Bitflash und Zoomon zeigten neue Versionen ihrer SVG Mobile Implementierungen. Eine Aufwertung erhielt die Messe durch angeschlossene Posterpräsentationen verschiedener SVG Projekte.

Des weiteren fanden zusätzlich an den ersten beiden Tagen Workshops statt. Neben Techniken/Technologien rund um den Themenkomplex SVG wurden Webmapping Kurse angeboten.

Als Highlights der Konferenz sind besonders die in der Computerfachwelt namhaften Hauptredner zu erwähnen. Neben den XML-Mitbegründern, Paul Prescod und Tim Bray, waren das beispielsweise Gesandte der Firmen Microsoft oder Corel. Zudem stellte Corel seine Smart

Graphics Solutions vor (vgl. oben), Microsoft kündigte die SVG Unterstützung in Microsoft Visio an (SVG Import/Export/Roundtrip). Adobe zeigte den Alpha-Release des Adobe SVG Viewers 6 anhand von beeindruckenden Beispielen. Auch SUN Microsystems, federführend im Batik Projekte, bot interessante Beispiele.

Die zahlreichen Diskussionen um die Verbesserungen in SVG 1.2 und 2.0 waren weitere, sehr positive Aspekte. Aus Mitteln der letztjährigen SVG Konferenz konnte zudem die Reise von Lauris Kaplinski aus Estland, einem sehr engagierten Autor der Open-Source Software Sodipodi zur Erstellung von SVG Graphiken finanziert werden. Er stellte die neuesten Errungenschaften aus seiner weltweiten Entwicklergruppe vor.

Auch wenn der Eindruck nach außen hin vortäuscht, dass SVG sich nur langsam weiterentwickelt, so hat die Konferenz deutlich gemacht, dass doch sehr viel Engagement von vielen bekannten Firmen, Institutionen und Einzelpersonen in SVG eingebracht wird. Im Vergleich zum Vorjahr konnten zahlreiche Fortschritte erzielt werden. Schließlich ist auch Fakt, dass SVG in Zukunft noch mehr "Marketing" braucht, um bei Webdesignern und Entwicklern bekannter zu werden. Noch zu wenige Entscheidungsträger kennen das hohe Potential und die interessanten technischen Möglichkeiten, die SVG über Plattform- und Systemgrenzen hinweg bietet. Derartige Konferenzen fördern die Verbreitung wesentlich. Alle Dokumentationen, Beispiele und natürlich Präsentationen können, wie auch jene des Vorjahres, auf der SVG.Open Seite eingesehen werden [URL\[08\]](#).

## **2 Aktuelle Möglichkeiten mit SVG**

### **2.1 Konkrete Techniken**

Trotz viel versprechender Anwendungen in der Webkartographie ist SVG kein "kartographisches Format", sondern nach wie vor ein Graphikstandard. Deswegen müssen Anforderungen des Webmapping's, wie die Navigation, teilweise mühsam implementiert werden. Derartige, durch relativ komplexes Scripting zu erzeugende Funktionalität, stellt sich gegenwärtig für Kartographen als Schwierigkeit heraus. Im Folgenden dazu wichtige Grundideen sowie Lösungsansätze.

- **Wiederverwendbare Graphiken und Symbole**

SVG erlaubt es, global definierte Elemente mittels `<use>` wieder zu verwenden. Möglichkeiten der Definition gibt es verschiedene. Allgemein können beispielsweise durch das `<defs>` Element Graphiken zur Interfacegestaltung definiert werden. Kartographische Symbole oder Logos dagegen sind auch durch das `<symbol>`-Element möglich.

- **Layout und CSS**

Alle graphischen Effekte, die neue Vektorgraphikprogramme anbieten, sind in SVG darstellbar, es gibt somit keine graphischen Einschränkungen. Beim konsequenten Einsatz von CSS (Cascading Sytle Sheets) können Projekte zentral im Layout abgeändert werden. In einer globalen Stylesheet Datei werden Formate für das Interface, die Karte und z.B. auch die völlig

abgekoppelte Projektbeschreibung geregelt.

- **Generierung von Elementen am Client**

Die technische Grundlage für Interaktivität und deren Auswirkungen ist die Erzeugung und Manipulation von Elementen. Per JavaScript können sämtliche Elemente auch am Client neu kreiert werden. Auslöser für den Aufruf der Scriptroutinen sind immer so genannte Ereignisse, in der Fachwelt als "events" bekannt. Besonders komplexe kartographische Anwendungen haben nur einen geringen Anteil statischer Elemente. Der Großteil wird dynamisch generiert oder bei Bedarf nachgeladen (vgl. nächster Absatz). Aus diesem Grund kann der erste Blick in den Quellcode einer SVG-Anwendung enttäuschend wirken. Blickt man allerdings "hinter die Kulissen" – hier in den JavaScript Code – bekommt man einen Einblick auf die Gesamtapplikation. Kartographische Anwendungen mit Datenaustausch, interaktiver Datenanzeige und Zoomfunktionalität enthalten mitunter 1000 Zeilen Skriptcode für Interaktion und Funktionalität. SVG Viewer müssen dabei den Scripting Standard ECMA URL[09], das DOM2 Interface URL[10] und logischerweise das SVGDom URL[11] unterstützen. Diese Spezifikationen sind die wichtigsten Nachschlagewerke für erfolgreiche Skripterstellung.

- **Nachladen von Daten mit GetURL/ParseXML**

Die noch nicht standardisierten, aber in fast allen Viewern implementierten JavaScript Methoden `getURL()` bzw. `parseXML()` ermöglichen das Nachladen von weiteren Daten in SVG Dateien. Teile von Dateien, also z.B. eine Kartenebene, einzelne Symbole etc. werden als so genannte XML-Fragmente direkt aufgerufen und eingefügt. Dabei spielt es keine Rolle, ob das nachzuladende Fragment in einer Datei vorliegt oder durch ein serverseitiges Script erzeugt wird. Daten und vorgefertigte SVG Elemente können "ondemand" die Karte ergänzen. Beliebte Anwendungsfälle ergeben sich besonders im Bereich der thematischen Kartographie. Aktuelle Daten werden beispielsweise direkt aus einer Datenbank nachgeladen, zu SVG Diagrammen weiterverarbeitet und in die Karte eingebettet. Sämtliche auf diese Art und Weise eingefügten Elemente sind nach dem Ladevorgang in der SVG Datei verfügbar, als wären sie ursprünglich schon vorhanden gewesen.

- **Koordinatensysteme**

Wie oben schon erwähnt, liegt der Ursprung SVGs in der Wissenschaft der Computergraphik und nicht in der Kartographie. Das Koordinatensystem SVGs ist auf die Bildschirmausgabe ausgerichtet - kartesisch mit negativer Y-Achse. Im Gegensatz dazu werden in der Welt der Geoinformation meist Koordinatensysteme mit positiver y-Achse verwendet. Des wegen müssen sämtliche Kartengeometrien auf das SVG-Koordinatensystem transformiert werden. Dies kann zum einen durch das Spiegeln der y-Achse mittels des SVG Attributs "transform" geschehen. Andererseits ist es möglich, bereits bei der Erzeugung der SVG Karte die "Geo-Koordinaten" in das Bildschirmkoordinatensystem zu transformieren. Letztere Methode ist bei komplexeren Projekten vorzuziehen, da sich das Spiegeln der y-Achse auch auf alle anderen

Kartenelemente, wie Texte oder Diagramme, auswirkt. Diese würden auf den "Kopf gestellt" werden.

#### • **Kartennavigation (Zoom und Pan)**

Obwohl SVG Viewer Zoom- und Panfunktionalitäten beinhalten, ergeben sich Probleme beim Umgang mit verschachtelten Bildschirminhalten einer Benutzeroberfläche. In einer Kartenapplikation soll nur die Karte selbst, und nicht die gesamte Benutzeroberfläche verschoben oder skaliert werden. In der Praxis unterstützen Viewer nicht die separate Navigation im Kartenfenster, sondern nur global im ganzen SVG Dokument. Nach derzeitigem Stand bleiben zwei Optionen zur Kartennavigation:

1) "SVG standalone": Die viewerspezifischen Zoom- und Panfunktionalitäten werden gesperrt. Jegliche Navigationsfunktionalität muss durch kompliziertes Scripting erzeugt werden. Theoretisch sind mittels Skripten alle Bewegungen des Users erfassbar und auch einschränkbar. Praktisch stellt sich jedoch oftmals heraus, dass einfach zu viele Ressourcen verschlungen werden und die Anwendung langsam wird. Zusätzliche Komponenten, wie aktive Übersichtskarten, tragen trotzdem zum komfortablen navigieren bei. "SVG standalone" umgeht die fehlerbehaftete HTML-SVG Scripting Kommunikation und ist deswegen konsistenter.

2) SVG eingebettet in HTML: Der User kann die viewerspezifischen Zoom- und Panfunktionalitäten verwenden. Der Umgang mit der Karte erscheint deswegen bequemer. Jedoch ergeben sich daraus erhebliche Probleme in verschiedenen Browsern und PlugIns, da die HTML-SVG Kommunikation verschiedenartig abläuft. Zudem können einige Viewer "per default" nur mit für die Kartennavigation unbrauchbaren Sprungwerten Abbildungen skalieren. Stufenlose Skalierung ist deswegen nicht immer gegeben.

SVG 1.2 und verbesserte Implementierungen werden diesem Manko Abhilfe leisten, indem getrennt für verschachtelte Viewboxen festgelegt werden können.

## **2.2 Realisierte Projekte**

#### • **Tirol Atlas der Universität Innsbruck [URL12]**

Der Tirol Atlas ist die Weiterführung eines klassischen Papieratlases, der 1999 beendet wurde. Die vorliegende Internetlösung ist mehrheitlich datenbankgestützt und verwaltet auf diese Art sowohl Sachdaten als auch Geometrie und Bilddaten. Graphiken, darunter besonders dynamische Karten und Diagramme, werden in SVG bereitgestellt. Seit 2002 sind Teile des neuen Angebots online, eine Generalüberholung des Kartenteils erfolgte schon Mitte 2003. Des Weiteren ist seit Herbst 2003 ein Bereich mit der Zielgruppe Schulen neu eingeführt worden.

Technisches Kernstück des Atlases ist eine PostGres-Datenbank [URL13] die mit dem PostGIS-Modul [URL14] auch räumliche Abfragen bewältigen kann. Aufgrund einiger noch zu lösender Probleme bei der Anzeige von großen Datenmengen, wie sie z.B. bei topographischen Karten anfallen, sind diese noch im Prototypstadium. Jedoch werden die Möglichkeiten von SVG 1.0

bereits stark ausgeschöpft und innerhalb einer Kartenansicht sowohl statische als auch datenbankgestützte Sachdaten und Geometrie eingesetzt. Eine umfassende Beschreibung mit Links zu Beispielen wurde bei der SVG-Open Conference im Juli 2003 vorgestellt [URL15].

- **Thematische Karten im Tirol Atlas [URL16]**

Der Kartenteil des Atlas wird von Redakteuren mittels eines Intranet-Interfaces befüllt. Standen ursprünglich nur Choroplethenkarten, Kreis- und Tortendiagramme zur Verfügung, so wurde diese Palette nun mit weiteren Funktionalitäten ergänzt. Darunter die Möglichkeit, Kartenelemente frei mit Funktionen zu versehen die aus einem anderen Bereich des Atlas Daten abrufen und anzeigen können (z.B. Datenblätter, komplexe Diagramme in eigenen Fenstern, etc.). Damit lassen sich aktuellste Daten abrufen. Karten zu Wahlen wurden beispielsweise im September noch am Abend des Wahltags online gestellt [URL17].

Clientseitig wird ein Interface zur Verfügung gestellt, das alle Daten zu einem thematischen Kartenset vom Server abrufen und gleichzeitig die nötigen Javascript-Bibliotheken bereitgestellt bekommt, um daraus dynamische Karten zu generieren: die Kartengrundlagen (Schummerung, Flüsse, Orte, politische Einheiten) sind für die thematische Darstellung statisch, die Einfärbung der Choroplethen nach eigenen Farbskalen sowie das Generieren der Diagramme erfolgt clientseitig. Das gilt auch für aufwändigere Diagrammtypen wie Bevölkerungspyramiden, auch diese bekommen "nur" eine Datenliste und Funktionen zum Erstellen der Graphik bereitgestellt.

Ein flexibles räumliches Navigationsinterface mit freier Bewegung in der Hauptkarte mittels der Viewerfunktionalitäten "Zoom und Pan" sowie einer intelligenten Übersichtskarte ergänzt eine fixe und klassische hierarchische thematische Navigation. Ein zuschaltbares und mehrschichtiges Legendenfenster bietet auf den jeweiligen Karteninhalt angepasste Legenden und Zusatzfunktionen sowie Erklärungstexte.

- **Der Schulbereich des Tirol Atlas [URL18]**

Als EU-InterRegIIIA-Projekt kommt der Tirol Atlas auch seinem Bildungsauftrag nach. Der Hauptteil des Atlases hat teilweise Inhalte auf wissenschaftlichem Niveau, die es nötig machen, für jüngere Interessierte ein anders strukturiertes Angebot bereitzustellen. Dies erfolgt in einem eigenen Bereich mit eigenem Layout. Module wie ein Memory, Bilderrätsel (Tirol in Bildern) oder Quiz sind in zentralen Skript-Bibliotheken definiert und Abwandlungen davon kommen mit verschiedenen Inhalten zum Einsatz. Teile davon kommen aus der oben genannten Datenbank, andere sind statisch. Die Gliederung ist bis auf weiteres noch schlicht und wird sobald mehr Inhalte vorliegen auf eine didaktisch-thematische Struktur umgebrochen. Der Jugendbereich ist bereits online, wird in dieser ersten Phase jedoch noch nicht intensiv beworben.

Technisch betrachtet kommen im Schulbereich "SVG standalone" Lösungen und HTML mit DOM-Manipulation und eingebetteten SVG zum Einsatz. Da alle Spiele und Lernmodule eigene Funktionen brauchen, wird in der Regel nur der Datenblock (z.B. eine Liste mit Fragen, Antworten und Bildern für ein Quiz) mit der Datenbank bedient, das Template wird händisch angepasst und im Inhaltsverzeichnis freigeschaltet.

- **Topographische Karten mit SVG**

Das Beispiel "Türlensee" soll die Möglichkeiten von SVG für interaktive topographische Karten demonstrieren [URL19]. Im Prototyp werden die Demo-Daten der schweizerischen Landestopographie (Vektor25, Pixelkarten 1:25.000) verwendet. Der Kartennutzer kann im Gebiet frei navigieren, die Koordinaten inklusive der Z-Koordinate anzeigen, sowie Orte und geographische Features lokalisieren. Bei den Kartenebenen ist es möglich, zusätzlich eine analytische Schattierung einzublenden, deren Lichtrichtung und Vertikalwinkel interaktiv angepasst werden kann. Zusätzlich können GIS-Attribute, gemäss dem ausgewählten Layer abgefragt werden. Eines der neueren Features ist ein interaktives Profiltool, bei dem der Nutzer in der Karte eine Profillinie digitalisieren kann und danach am Client ein Profil berechnet wird. Das resultierende Profil ist mit der Karte gelinkt (mouseover im Profil zeigt Position in der Karte), und der Benutzer kann Analyseparameter wie Distanz vom Ausgangspunkt, Höhenwert

an der jeweiligen Position, sowie Minima und Maxima des Profils abfragen. Schließlich können in der Karte auch vorberechnete Routen zwischen 2 Orten in der Karte animiert dargestellt werden. Dabei ist zu erwähnen, dass die Route statisch in einem SVG File vorliegt und nicht von einem Routing-Algorithmus berechnet wird.

Die ursprünglichen Vektordaten für das Projekt stammen aus ESRI Shapefiles. Im Zuge des Projektes wurden Perlscripts geschrieben, die aus Shapefiles mit Hilfe des "PostGIS Loaders shp2pgsql" SVG Geometrie inklusive Attribute generieren können. Da die Koordinaten in quasi "Weltkoordinaten" konvertiert werden, können so verschiedene Datenquellen leicht kombiniert werden. Die zugehörigen Javascript Routinen, die die Interaktivität der SVG-Applikation steuern, wurden alle manuell geschrieben. Sie befinden sich derzeit in Überarbeitung, damit sie in Zukunft leichter auf andere Projekte portiert werden können, sowie auch in anderen SVG Viewern, als dem Adobe SVG Viewer lauffähig sind.

Dass sich die oben vorgestellte Technologie auch dazu eignet, im großen Stil eingesetzt zu werden, zeigt ein datenbankgestütztes Berichtsinformationssystem, das für die Firma Dr. Heinrich Jäckli AG (Geologie-Consultingbüro in Zürich) entwickelt wurde [URL20]. Hierbei wurden verschiedene GIS-Layer und Basiskartenlayer in die räumliche Datenbank PostGres mit der PostGIS-Erweiterung importiert. Entsprechend des gewählten räumlichen Ausschnittes des Kartennutzers werden die entsprechenden geographischen Objekte aus der räumlichen Datenbank extrahiert und im SVG Format an den Client geschickt. Die Objekte werden entsprechend der Größe des gewählten Ausschnittes in der Symbolisierung (etwa Strichstärken, Symbolgrößen) angepasst geliefert. Der Webclient kann neue Daten, ohne die Applikation neu laden zu müssen, sofort mit einbauen. Auf diese Art und Weise können sämtliche 20.000 Berichte der Firma geographisch lokalisiert werden und Metadaten zu den Berichten abgefragt werden. Zusätzlich zu den Anforderungen der Firma ist es nun möglich, beliebige GIS-Daten auf Wunsch in den Webmapping-Client mit aufzunehmen. Die Applikation wird im Firmenintranet von den 4 Standorten aus von ca. 40 geologischen und umwelttechnischen Fachleuten genutzt.

## 2.3 Probleme

Neue Technologien haben immer "Kinderkrankheiten", hinzu kommen im Internet auch immer wieder Interessenskonflikte von Konsortien und großen Firmen die negative Einflüsse auf die Verbreitung haben können. Beispielsweise würde eine 1:1 Implementierung von SVG in Microsoft Produkten (z.B. Office 2003 oder Internet Explorer) die Akzeptanz und somit den Einsatz von SVG fördern. Folgende Arten von Hindernissen können auftreten:

- **Clientseitige technische Hindernisse**

- **SVG-Viewer nur mit Administratorenrechten installierbar**

- Es gibt noch nicht für alle Betriebssysteme und Browser ausgereifte Viewer. Hier wird von Zeit zu Zeit aber eine Besserung in Sicht sein. Problematischer dagegen ist, dass zur Installation der Viewer

oftmals Administrator-Rechte benötigt werden. Diese Tatsache stellt sich besonders bei dem aktuell am weitverbreitetsten SVG Viewer dar, dem Adobe SVG Viewer3 unter Microsoft Windows XP/2000.

- **Plugin-Browser Kommunikation**

Viewer liegen von verschiedenen Herstellern in verschiedenen Versionen vor. Es gibt allerdings keine einfachen clientseitigen Abfragemöglichkeiten zum Viewertyp. Viewer werden vorrangig als Plug-Ins in Browsern eingesetzt. Dies erlaubt das Einbetten von SVG in HTML-Seiten. Aber nicht alle Browser erlauben eine skript-basierte Kommunikation zwischen Browser und Viewer (betrifft vor allem SVG-Viewer auf Macintosh). Wie große Projekte bestätigen, ist ein gemischtes Auftreten von HTML und SVG häufig wünschenswert.

Selbstverständlich unterstützen unterschiedliche Viewer verschiedene Teile der Spezifikation. Dies ist nicht mit einem Browserkrieg vergleichbar, da hier der Grund in der Evolution der Spezifikation und im ständigen Ausbau der Viewer liegt; für den Programmierer hat es aber derzeit ähnlich unangenehme Nebenwirkungen.

- **Fehlerbehaftete Viewer**

Der am weitesten verbreitete Adobe SVG-Viewer 3 hat Fehler wie jedes Programmpaket es haben kann, allerdings sind diese gerade bei aufwändigen Projekten, wie es kartographische Projekte häufig sind, störend: Die Reaktionszeit auf Maus-Events (Bewegen, Klick, etc.) in Kombination mit Zoom- und Pan-Events ist u.U. nicht synchronisierbar. So kann es vorkommen, dass Skripte, die Übersichtskärtchen nachführen, schneller ausgeführt werden als die Hauptkarte am Bildschirm. Alle Werte die aufgrund eines Vorgängerwertes verändert werden (also vom Typ  $x=x+z$ ) sind somit potentielle Problemquellen, geometrische Veränderungen lassen sich aber analytisch meist nur über diesen Weg ermitteln.

Teilweise wäre auch die Spezifikation selbst noch ausbaufähig, so gibt es etwa Probleme bei wieder verwendeten Symbolen wenn diese deklarative Animationen beinhalten: sämtliche abgeleiteten Symbole erben dann dieselbe Animation obwohl das in einigen Fällen nicht gewünscht wird.

- **Ladezeiten**

Vektorinhalte können recht groß werden, da macht es Sinn wenn bereits geladene Daten bei Wiederverwendung aus dem Browser-Cache kommen. Das Caching von SVG Daten ist verbesserungswürdig.

- **Anwenderbedingte Probleme**

- **Plug-In-Größe**

Da die meisten Viewer eigene Graphik- und Skripting-Bibliotheken mitliefern um Probleme bei der Browserkommunikation zu umgehen (vgl. auch Punkte oben), werden die Installationspakete recht groß. Der Wille beim unversierten Endanwender, 2-4 MB zur Installation herunterzuladen, ist nicht immer gegeben, besonders wenn er die Qualität des folgenden Angebots nicht kennt.

Es ist jedoch zu erwähnen, dass ein Plug-In im Gegensatz zu beispielsweise Java Applets nur einmal heruntergeladen werden muss. Des Weiteren ist ein Plug-In wesentlich vielseitiger einsetzbar, da er komplette Technologien beinhaltet und nicht nur nutzerspezifische Anwendungen.

- **Usability - Benutzerfreundlichkeit**

Reine HTML-Seiten, auch wenn sie aufwändig gestaltet sind, haben meist einen klassischen Aufbau. Vektorbasierte Seiten, wie SVG und Flash, verhalten sich anders (z.B. beim Verändern der Fenstergröße, beim Auswählen von Text, beim Drucken, etc.) und sind teilweise fundamental anders gestaltet. Dies hat zur Folge, dass in vektorbasierten Anwendungen oft der Zweck und Funktionalität häufig nicht sofort erkannt wird. Häufig kollidiert auch die scheinbar grenzenlose Freiheit der Webdesigner mit den heute meist allgemein akzeptierten GUI-Konzepten, deren striktere Einhaltung häufig die Benutzerfahrung und Einarbeitungszeiten stark verbessern würde. Während in Computerspielen die "Geheimhaltung" von Bedienelementen oder eine rätselhafte Benutzerführung ein durchaus wünschenswertes Feature sein kann, ist dies bei seriösen Webapplikationen und Informationssystemen selten der Fall. In vielen Projekten müsste vermehrt in "Usability" investiert werden.

- **Internetanbindung**

Die derzeit im Einsatz befindlichen Internetanbindungen haben sehr unterschiedliche Übertragungsraten. Kartographische Anwendungen werden immer recht große Datenmengen übertragen müssen, die Zeiten dafür sind aber weder programmtechnisch vorhersehbar, noch können sie wirklich effizient minimiert werden. Dieses Problem ist nicht rein SVG-spezifisch und fällt beim Einsatz von Vektordaten ins Gewicht.

- **Autorensysteme**

Noch immer fehlen ausgereifte und von SVG Entwicklern akzeptierte Autorensysteme, wie beispielsweise Macromedia Flash für .swf. Man muss eingestehen, dass sich dieser Misstand jedoch von Zeit zu Zeit verbessert.

- **Technologische und verwandte Probleme**

- **Datenschutz**

SVG sieht von sich aus keine Schutzmechanismen gegen Datenklau vor. Da SVG XML-basiert ist, kann es von jedem Texteditor gelesen werden. Der Datenschutz in Webanwendungen ist jedoch nicht nur SVG-spezifisch. Jede Client-Server Architektur ermöglicht es, übermittelte Information durch "Sniffer" auszuspionieren. So sind beispielsweise Macromedia Flash Filme durch bestimmte Softwareprodukte auch lesbar. Allein die Möglichkeiten, proprietäre Formate zu interpretieren ist etwas größer. Lediglich durch Paßwortschutz "gesicherte Internetprotokolle" gewährleisten den notwendigen Schutz.

- **Indizierung von Suchmaschinen**

Aufgrund dessen, dass SVG XML-basiert ist, können SVG Dokumente theoretisch "durchsucht" werden. XML, und somit auch SVG, wird allerdings von namhaften Suchmaschinen (darunter auch Google) noch nicht indiziert. Der Nutzerkreis von SVG Seiten schränkt sich dadurch erheblich ein.

Die oben genannten Probleme im Umgang mit SVG-Viewern klingen zwar recht dramatisch, es können aber alle abgefangen und umgangen werden. Es ist zu betonen, dass diese Aussagen für den Entwicklungsstand Ende 2003. Die im nächsten Kapitel aufgeführten Beispiele sind richtungweisend und können so manchen negativen Punkt relativieren.

### 3 Zukünftige Features

Vordergründig ist zu erwähnen, dass zukünftige Features besonders durch den Working Draft 1.2 der SVG Arbeitsgruppe auszumachen sind. Jedoch ergeben sich neue Möglichkeiten auch aus der Weiterentwicklung der SVG Viewer. So ist es gewöhnlich nicht möglich, alle Inhalte aktueller Spezifikationen in kurzer Zeit zu implementieren (vgl. 3.1 -> Cursor).

#### 3.1 Übersicht der wichtigsten Erneuerungen

- **Cursor**

Die Eigenschaft "cursor" bzw. das Element <cursor> erlauben es, verschiedene Ausprägungen des Mauszeigers zu definieren. Wird einem Element ein Cursor zugewiesen, ändert sich der Mauszeiger beim Überfahren des Elementes entsprechend. Man unterscheidet zwischen vordefinierten Cursorausprägungen, wie beispielsweise ein Fadenkreuz, und benutzerdefinierten Cursorsymbolen. Letzteres kann durch den Gebrauch des Elementes <cursor> definiert werden. Als Symbol müssen PNG Bilder referenziert werden.

Obwohl Cursor schon seit Version 1.0 spezifiziert sind, gelten sie als neues Feature. Das liegt vor allem an der bisher fehlenden Unterstützung der weit verbreiteten SVG Viewer.

- **Multiple Pages**

Der Bedarf, mehrere Seiten innerhalb eines SVG Dokumentes anzulegen, ist vor allem für die sich in Arbeit befindliche SVG Print Spezifikation sehr groß. Dabei zeigt der SVG Viewer eine definierte SVG Seite separat für sich an. Eine SVG Datei kann deswegen aus mehreren Seiten bestehen. Zwischen ihnen kann ohne komplexe Scriptroutinen gewechselt werden.

- **Audio/Video Einbindung**

Audio bzw. Video sind für ein modernes Vektor-/Multimediaformat zwingend erforderlich. Deswegen entschied die Arbeitsgruppe, Möglichkeiten für Ton und Film in den Working Draft 1.2 aufzunehmen. Komponenten des Animationsstandards SMIL 2.0 bieten dafür die Basis. Wichtiges Feature ist die Synchronisation der verschiedenen Sequenzen. Allgemein problematisch sind proprietäre Medienformate, die bei einer Implementierung Lizenzkosten erfordern.

- **Tooltips**

Tooltips sind in der interaktiven Kartographie ein wirkungsvolles Darstellungsmittel. Beim Überfahren eines Objektes mit dem Mauszeiger werden textuelle Informationen direkt neben dem Zeiger eingeblendet. Ist einem Element die "tooltip" Eigenschaft angefügt, wird der Text des als Kind definierten <title> Elementes eingeblendet. Tooltips konnten bisher nur über eigene Scripte realisiert werden.

- **Streaming**

Für den Download komplexer Zeit-basierter SVG Anwendungen ist es vorteilhaft, die Anwendung schon abspielen zu können, bevor das komplette SVG geladen ist. Streaming-Technologien sind in der "Netzwelt" weit verbreitete Methoden, Daten zu übermitteln. Dabei wird eine Datei in einzelne Datenpakete zerlegt. Der Client spielt relevante Inhalte zuerst ab, während spätere Daten hinten an stehen. Der SVG Entwickler muss dabei auf die Reihenfolge zentraler Definitionen, wie beispielsweise Styles oder Symbole, achten. Durch Streaming wird die dem Benutzer zugemutete Wartezeit erheblich verkürzt.

- **Textwrapping**

Eine bisher große Schwäche SVGs, die unzureichenden Möglichkeiten im Umgang mit Textblöcken, werden durch den Working Draft 1.2 ausgemerzt. Zwar konnte bis dato Text in vollen Zügen durch "Styling" graphisch aufbereitet werden. Jedoch war es sehr mühsam, mit Textblöcken zu arbeiten. Die Erneuerungen ermöglichen, Text automatisch in ein vordefiniertes Feld zu platzieren. Die notwendigen Zeilenumbrüche erzeugt der Viewer automatisch. Dabei kann beispielsweise die Ausrichtung des Textblocks zwischen Links, Mitte und Rechts variiert werden. Mühsames Scripting um automatische Zeilenumbrüche zu erzeugen, gehört damit der Vergangenheit an (vgl. 3.2 Beispiele).

- **Rendering Custom Content**

Rendering Custom Content - kurz RCC - benennt eine besondere Möglichkeit, "eigene" Vorlagen zu definieren und als Instanzen wieder zu verwenden. Die Besonderheit zu ähnlichen Techniken älterer Spezifikationen (vgl. 2.1 -> Wiederverwendbare Graphiken und Symbole) besteht darin, dass auch zugehörige Scripte vordefiniert werden können.

Eine Vorlage wird erzeugt, indem die notwendigen Elemente mit verschiedenen Attributen

festgelegt werden. Der daraus resultierende fiktive Elementenbaum wird in der Fachwelt als "shadowTree" bezeichnet. Zusätzlich können in einer Vorlage beliebige Scripte definiert und durch Events ausgelöst werden. Diese bewirken, dass beispielsweise beim Instantiieren, Knoten und Attribute dem "shadowTree" hinzugefügt oder entfernt werden können. Es besteht deswegen die Möglichkeit, vordefinierte Elemente dynamisch an die verschiedenen Dokumente, die die Instanzen enthalten, anzupassen. Ein typisches Anwendungsfeld ist die Definition einer Diagramm-Bibliothek für thematische Karten (vgl. 3.2 Beispiele).

RCC wurde vor allem aufgrund der großen Nachfrage seitens der SVG-Entwickler nach Möglichkeiten, eigene XML-Namensräume ohne wiederkehrende Scripteinbindung in SVG darzustellen, entwickelt. So lässt RCC beispielsweise die Definition von reinen SVG-Benutzeroberflächen zu. Dabei können einfache Attribute die SVG-GUI beliebig aussehen lassen.

Technisch gesehen unterscheidet man allgemein SVG Inhalt (untere Ebene) und RCC Definition (obere Ebene). Die obere Ebene – RCC – stellt eine Vorlage zur Verfügung die auf der unteren Ebene als Instanz gerendert wird. Derartige Ansätze lassen den Aufbau modularer Syntax zu und erscheinen deswegen als besonders elegant.

RCC hat sicherlich auch seine Grenzen. Besonders bei großen Datenmengen sollte beachtet werden, dass "Custom Content" ausschließlich auf dem Client gerendert wird. SVG Karten, die beispielsweise komplexe Geodaten beinhalten, sollten deswegen nach wie vor serverseitig generiert werden.

## 3.2 Beispiele

Die vorliegenden Beispiele zeigen einige der schon im Adobe SVG Viewer 6 pre-alpha [URL21] implementierten Features des Working Draft 1.2. Die Beispiele sind unter [URL01] zu finden.

### • **Der Einsatz verschiedener Mauszeigerausprägungen**

Die vorliegende SVG Datei zeigt die in der Spezifikation vordefinierten Ausprägungen des Mauszeigers. Wie oben schon erwähnt, erlaubt das <cursor> Element die Definition verschiedener benutzerdefinierter Mauszeigerausprägungen. In der Kartographie kann diese Möglichkeit dazu verwendet werden, um Kartenobjekte durch ein-/ausblendende Symbole genauer zu spezifizieren. In der Abbildung stellen die grauen Rechtecke Kartenobjekte dar, die beim Überfahren mit der Maus durch die Cursorausprägung genauer spezifiziert werden.

### • **Arbeiten mit Textblöcken**

Das vorliegende Beispiel zeigt die neuen Möglichkeiten SVGs im Umgang mit Text. Wie im vorherigen Abschnitt schon erwähnt, bieten die neuen Elemente wesentlich einfachere Optionen bei der Definition von Textblöcken. Wie die Abbildung zeigt, werden verschiedene Textpassagen einer Geometrie zugewiesen und automatisch ausgerichtet (z.B. Links oder Mitte).

## • ChartLibrary mit RCC

Wie im vorherigen Abschnitt bereits erwähnt, lässt die RCC Technologie es auf besonders sinnvolle Art und Weise zu, eine ChartLibrary für thematische Karten zu erstellen. Zuerst muss aufgearbeitet werden, welche Parameter neben den eigentlichen Quantitäten weiteren Einfluss auf die Darstellung haben. Als Folge davon entsteht ein eigener XML-Namensraum für die Erzeugung thematischer Karten. Anhand dessen generieren Scripte die notwendigen Elemente und fügen diese dem "shadowTree" hinzu. Wird dieser benutzerdefinierte Namensraum – im Fachjargon "name space" – dem SVG hinzugefügt, dient er damit als Instanz für die jeweiligen vordefinierten RCC Elemente. Diagramme werden nun entsprechend dargestellt.

Vorteile gegenüber der ursprünglichen Generierung mit "Javascript only" lassen sich besonders leicht ausmachen. Der Code ist wesentlich übersichtlicher, da ein einmalig eingebundenes RCC Modul wiederkehrend instantiiert werden kann. Das "Chartmodul" ist zudem auf sehr einfache Art und Weise in anderen Projekten wiederverwendbar.

Es wird darauf hingewiesen, dass RCC keineswegs eine Lösung für kartographische Probleme, wie beispielsweise die Diagrammplatzierung/-skalierung, darstellt.

## 4 Quellen

URL01: Carto.net Seite zum Web-Mapping Beitrag, <http://www.carto.net/papers/svg/webmapping/>

URL02: World Wide Web Consortium – SVG Arbeitsgruppe, <http://www.w3.org/Graphics/SVG/>

URL03: Corel Smart Graphics, <http://www.smartgraphics.com/>

URL04: Corel Press Room - Oracle and Corel to Provide Open Standard Content Solutions,

<http://www.corel.com/servlet/Satellite?pagename=Corel/Company/PressRoom/PressRelease&id=1044393442588&prid=1047022432211>

URL05: SVG Implementation Page, <http://www.w3.org/Graphics/SVG/SVG-Implementations.htm#8>

URL06: SVG Wiki, <http://www.protocol7.com/svg-wiki/>

URL07: Cartographic Wishlist, [http://www.carto.net/papers/svg/wishlist/wishlist\\_sep\\_18\\_2003.pdf](http://www.carto.net/papers/svg/wishlist/wishlist_sep_18_2003.pdf)

URL08: SVG.Open Homepage, <http://www.svgopen.org/>

URL09: Standard ECMA 262 – ECMAScript Language Definition, 3<sup>rd</sup> edition, <ftp://ftp.ecma.ch/ecma-st/Ecma-2662.pdf>

URL10: DOM2 Interface, <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/core.html>

URL11: ECMAScript Language Binding for SVG, <http://www.w3.org/TR/SVG11/ecmascript-binding.html>

URL12: Tirol Atlas der Universität Innsbruck, <http://tirolatlas.uibk.ac.at/>

URL13: PostGres, <http://www.postgresql.org/>

URL14: PostGIS, <http://postgis.refrains.net/>

URL15: Tirol Atlas, An SVG based online Atlas Information System, SVG Open 2003 Proceedings,

<http://www.svgopen.org/2003/papers/TirolAtlasOnlineWebMapping/>

URL16: Thematische Karten im Tirol Atlas , <http://tirolatlas.uibk.ac.at/maps/thematic/>

URL17: Tirol Atlas Wahlen, <http://tirolatlas.uibk.ac.at/maps/wahlen/>

URL18: Der Schulbereich des Tirol Atlas, <http://tirolatlas.uibk.ac.at/kids/>

URL19: SVG Beispiel Türlersee, <http://www.carto.net/papers/svg/tuerlersee/>

URL20: SVG Open 2003 Proceedings,

[http://www.svgopen.org/2003/papers/abstract\\_neumann\\_topographic\\_maps\\_served\\_by\\_spatial\\_databases/index.html](http://www.svgopen.org/2003/papers/abstract_neumann_topographic_maps_served_by_spatial_databases/index.html)

URL21: Adobe SVG Viewer 6 pre-alpha , <http://www.adobe.com/svg/viewer/install/beta.html>