



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Departement Bau, Umwelt und Geomatik
Studienrichtung Geomatikingenieurwissenschaften

Vertiefungsblock Kartographie WS 2003/04

Interaktive Webkarte vom Katasterplan Morcote (Tessin)

Leitung:

Prof. Dr. L. Hurni, Institut für Kartographie (IKA)

Betreuung:

Olaf Schnabel, Institut für Kartographie (IKA)

Student:

Christian Höhener

6. Februar 2004

I Vorwort

Die Arbeit „Interaktive Webkarte vom Katasterplan Morcote“ entstand im Rahmen eines Vertiefungsblockes in Kartographie am Institut für Kartographie (IKA) der ETH Zürich.

Im Diplomstudiengang Geomatikingenieurwissenschaften am Departement Bau, Umwelt und Geomatik der ETH steht den Studenten im letzten Jahr des Studiums Zeit zur Verfügung, sich im Sinne einer Vertiefung zweimal einer Projektarbeit zu widmen. Dies ist für sie eine wertvolle Gelegenheit, ihr Wissen in die Praxis umzusetzen und eine vorgegebene Problematik von Anfang bis Schluss zu behandeln.

Ich möchte mich an dieser Stelle bei Herrn Prof. Dr. Lorenz Hurni für die Ermöglichung dieses Vertiefungsblockes und bei Olaf Schnabel für seinen unermüdlichen Einsatz und seine Unterstützung bei der Arbeit herzlich bedanken.

Christian Höhener
Gäbrisstr. 4
9056 Gais
[chrigel@student.ethz.ch]

II Zusammenfassung

Mit der enormen Verbreitung des Internet ist die Publikation von Geodaten auf diesem Medium sehr aktuell geworden. Nachdem nahezu alle Städte über einen elektronischen Stadtplan verfügen, gibt es immer mehr Gemeinden, die einen solchen auch anbieten wollen. So auch die Behörden des Kanton Tessins, die den Katasterplan von Morcote auf dem Internet präsentieren wollen. Zusätzlich prüfen sie eine webbasierte Aktualisierung der Daten.

Die vorliegende Arbeit hat zum Ziel, eine interaktive Webkarte zu entwickeln. Auch eine Möglichkeit zur Manipulation der Daten soll geprüft werden. Da Katasterdaten fast tagtäglich ändern, muss die Erstellung dieser Karte möglichst automatisch von sich gehen.

Zur Erreichung dieser Ziele wurde ein zweispuriges Konzept erarbeitet. Die Geometrien werden als SVG Graphik dargestellt, die Sachdaten aber in einer Datenbank abgelegt. Verschiedene Skripte sind für die Interaktivität der Karte sowie für die Kommunikation mit der Datenbank zuständig.

Zur Konvertierung der Ausgangsdaten müssen zwei Tools gesucht werden. Verschiedene, auf dem Internet angebotene, Tools wurden miteinander verglichen. Daraus wurden zwei ausgewählt: „shp2svg“, ein Avenue-Skript zur SVG-Erstellung und „shpmysql“, eine ArcView-Extension, welche einen Export in die Datenbank MySQL ermöglicht.

Das Skript zur SVG-Erstellung wurde auf die genauen Anforderungen abgestimmt. Damit ist eine nahezu vollautomatische Erstellung der Webkarte gewährleistet. So können alle benötigten Dateien generiert werden. Darunter auch die PHP-Skripte, die für die Anbindung der MySQL-Datenbank verantwortlich sind. Mit diesen Skripten, sowie mit JavaScript-Funktionen, konnte die Interaktivität der Karte erfüllt werden.

Mit diesem Aufbau der Karte können Änderungen online vorgenommen werden. Eine komplette Aktualisierung aus den GIS-Daten ist auch nahezu vollautomatisiert. Zudem kann die Architektur für die Erstellung weiterer Webkarten ohne grossen Aufwand weiterverwendet werden.

III Inhaltsverzeichnis

I	Vorwort	2
II	Zusammenfassung	3
III	Inhaltsverzeichnis	4
1	Einleitung	5
2	Zielstellung	6
3	Methodik	7
4	Theorie Webkarten	8
	4.1 Allgemein.....	8
	4.2 Unterscheidung von Webkarten.....	8
5	Interaktionen	10
	5.1 Mögliche Interaktionen.....	10
	5.2 Sinnvolle Interaktionen für eine Karte.....	10
6	Analyse Ausgangsdaten	11
	6.1 Geometrie.....	11
	6.2 Attribute / Sachdaten.....	11
7	Techniken interaktiver Webkarten im Internet	13
	7.1 HTML Hypertext Markup Language.....	13
	7.2 SVG Scalable Vector Graphics.....	14
	7.3 JavaScript.....	15
	7.4 PHP Hypertext Preprocessor.....	15
	7.5 MySQL Datenbank.....	15
	7.6 Gewählte Möglichkeit.....	16
8	Umsetzung	17
	8.1 Untersuchung Konvertierungstools.....	17
	8.2 Auswahl Konvertierungstools.....	18
9	Praktische Umsetzung	19
	9.1 SVG-Karte.....	19
	9.2 Interaktivität.....	20
	9.3 Konvertierung der Shape-Files.....	22
	9.4 Webserver.....	23
10	Anleitung zum Export	25
	10.1 MySQL-Export.....	25
	10.2 SVG-Export.....	26
	10.3 Erstellung Webkarte.....	26
11	Diskussion der Lösung	27
	11.1 Probleme.....	27
	11.2 Kritische Betrachtung.....	28
	11.3 Erweiterungsmöglichkeiten.....	29
12	Schlusswort	31
13	Literaturverzeichnis	32
14	Abbildungs-, Tabellenverzeichnis	33

1 Einleitung

Das Kantonale Katasteramt Tessin erstellt seine Katasterpläne schon seit einigen Jahren mit Hilfe eines Geographischen Informationssystem (GIS). Da diese Geodaten einer breiten Öffentlichkeit zur Verfügung gestellt werden sollen, ist deren Veröffentlichung im Internet beabsichtigt. Mit dieser Möglichkeit kann die interessierte Öffentlichkeit die vollständigen und aktuellen Daten nutzen und sie können sogar über das Internet nachgeführt werden (z.B. vor Ort von Mitarbeitern).

Heute existieren verschiedene Möglichkeiten, um geographische Daten im Internet zu präsentieren. Die Palette reicht von einfachen Rasterbildern, über statische Vektorgraphiken bis zu interaktiven Karten, welche dynamisch geändert werden können.

Den Geodaten können zusätzlich noch mehrere beschreibende Merkmale zugeordnet werden. Mit der Geometrie eines Gebäudes ist zum Beispiel sein Eigentümer, dessen Adresse, aber auch die Strassennummer, die Assekuranznummer, das zugehörige Grundstück verknüpft. Solche Daten werden typischerweise in einem GIS erfasst und verwaltet.

Bei der Veröffentlichung auf dem Internet ist es sehr sinnvoll, wenn alle diese Informationen des GIS auch benutzt werden können. In einer Webapplikation können verschiedene Funktionen des GIS zur Verfügung stehen: Zooming, Panning, Abfragen der Attribute, verschiedene Analysefunktionen, sogar Änderungen und Aktualisierungen der Daten. Damit kann der Nutzer Daten genau nach seinen Bedürfnissen abfragen. Diese Informationen stehen grundsätzlich jedem Internetbenutzer zur Verfügung.

2 Zielstellung

Ziel der Arbeit ist es anhand eines Beispielsausschnitts (digital vorliegender Katasterplan von Morcote (Tessin)) eine möglichst einfache und automatische Überführung von GIS-Daten in eine interaktive Webkarte zu testen. Der Katasterplan wird vom Datenerheber (kantonales Katasteramt Tessin) laufend nachgeführt wird (z.B. Neubau von Strassen, Gebäuden, etc.). Damit auf dem Internet ein aktueller Zustand gewährleistet werden kann, muss die Karte möglichst einfach in relativ kurzen Intervallen aktualisiert werden können.

Die automatische Überführung soll möglichst allgemein gehalten werden, so dass sie später auch für andere Daten (z.B. weitere Katasterpläne oder Projekte auf Gemeindebasis) beliebiger Grösse genutzt werden kann.

Die so erzeugte Webkarte sollte interaktiv sein. Der Benutzer muss intuitiv mit der Karte arbeiten und seine gewünschten Informationen erhalten können. Zudem sollte die Karte über das Internet erweiterbar sein. Damit könnte ein Mitarbeiter ortsunabhängig Änderungen am Datensatz vornehmen.

3 Methodik

Zur Erfüllung der oben genannten Zielsetzung ist ein methodischer Aufbau der Arbeit erforderlich.

Zuerst muss die Ausgangslage untersucht werden. Dazu gehört eine Analysierung der heutigen Darstellungstechnologien von Webkarten (vgl. Kap. 4) und der Datenformate im Bereich des Internets (vgl. Kap. 7). Danach kann die zur Erfüllung der Ziele zweckmässigste Variante ausgewählt werden (vgl. Kap. 8). Schliesslich wird das Projekt dementsprechend umgesetzt (vgl. Kap. 9) und abschliessend die Lösung hinsichtlich ihrer Stärken und Schwächen diskutiert (vgl. Kap. 11).

Diese Arbeit fand im Rahmen eines Vertiefungsblockes statt. Der Zeitrahmen war demzufolge, wie dies auch bei einem realen Projekt der Fall ist, klar eingegrenzt. Zur Durchführung der Arbeit wurde ein Zeitplan aufgestellt, in welchem alle Arbeitsschritte sowie ihre Dauer in einem tabellarischen Rahmen aufgelistet sind. Nicht zu vergessen ist die Verfassung eines Berichtes mit der Erläuterung aller Arbeitsschritte.

4 Theorie Webkarten

4.1 Allgemein

Grundsätzlich kann jede Karte, die auf dem Internet angeboten wird, als Webkarte bezeichnet werden.

Hinsichtlich ihrer Gestaltung müssen die Eigenschaften des Ausgabemediums beachtet werden. Dies wird primär ein Computerbildschirm sein. Die Auflösung eines solchen liegt typischerweise bei 72 dpi, diejenige einer gedruckten Karte bei mindestens 300 dpi. Die Grösse der Karte ist auch in ihrem Ausmass begrenzt. Ein Monitor eines Webkartenbetrachters misst ungefähr 30 x 22 cm, eine analoge Karte kann um ein Vielfaches grösser sein. Dieser eingeschränkten Betrachtungsweise ist beim Entwerfen der Karte zwingend Beachtung zu schenken.

Eine Webkarte muss zudem von einem Server an den gewünschten Client übermittelt werden (Client-Server Architektur des Internets). Da die Übertragungsgeschwindigkeit begrenzt ist, sollte die Datenmenge der Karte nicht zu gross sein.

Auf dem Client wird die Internetseite durch einen Browser angezeigt. Heute werden verschiedene, nicht alle miteinander kompatible Browser benutzt. Bei der Erstellung einer Webseite ist daher darauf zu achten, dass diese mit möglichst allen üblichen Browsern dargestellt werden kann.

4.2 Unterscheidung von Webkarten

Webkarten können in folgende Kategorien aufgeteilt werden:

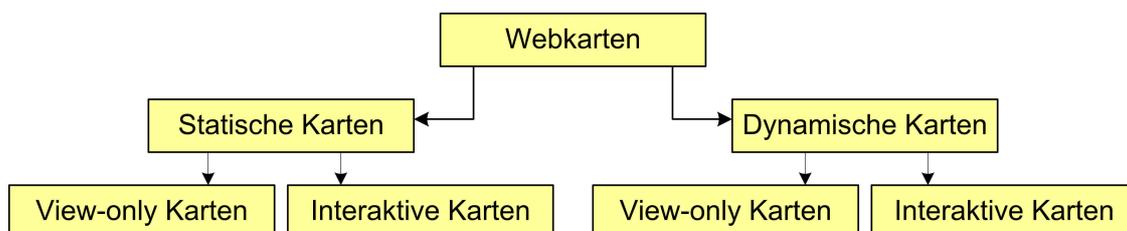


Abb. 1: Kategorien von Webkarten

Wie aus Abb. 1 ersichtlich ist, kann eine erste Unterteilung in statische und dynamische Karten gemacht werden. Diese werden weiter in view-only und interaktive Karten aufgeteilt.

4.2.1 Statische Karten

Statische Karten zeichnen sich dadurch aus, dass sie nur einmal erstellt werden. Eine Möglichkeit, die Karte über das Internet zu aktualisieren besteht nicht. Sie muss mit der entsprechenden Software vollständig neu erstellt werden.

4.2.2 Dynamische Karten

Dynamische Karten werden bei jedem Aufruf auf dem Server automatisch neu erstellt. Damit hat der Benutzer immer die aktuellsten Daten zur Verfügung. Er kann die Karte zudem auf seine persönlichen Bedürfnisse abstimmen. Als Nachteil erweist sich, dass die Karte bei jeder Änderung auf dem Server neu aufbereitet wird und diese Daten über das Internet an den Client gesendet werden müssen. Dieser Transfer erfordert natürlich eine zusätzliche Übertragungszeit.

4.2.3 View-only Karten

View-only Karten können nur betrachtet und nicht geändert werden. Sie liegen meist im Rasterformat vor (z.B. GIF, JPEG).

4.2.4 Interaktive Karten

Interaktive Karte bieten Möglichkeiten, welche weder analoge noch view-only Karten bieten. Sie erlauben Interaktionen zwischen dem Benutzer und der Karte. Dies sind Manipulation der Karte im Bereich der Visualisierung (Änderung der Graphik/des Bildes), Navigation (Rollover Events, Klickable Maps, etc.) oder der Analyse (Zooming, Panning, Rotation, etc.).

Beispielsweise können Informationen eines Objektes abgefragt werden oder Ebenen ein- und ausgeschaltet werden. Weiter können in sogenannt analytischen Karten Farben geändert oder sogar Analysen durchgeführt werden (Internet-GIS).

Für vorliegendes Projekt wird eine interaktive Karte benutzt. Durch die Interaktivität kann der Benutzer die Karte zu einem gewissen Grad verändern und seine gesuchten Informationen besser finden.

5 Interaktionen

Jeglicher Informationsfluss zwischen Mensch und Maschine wird als Interaktion bezeichnet. Das System kann auf einen Input des Benutzer reagieren und stellt die gewünschten Daten dar.

5.1 Mögliche Interaktionen

Es gibt sehr viele verschiedene Interaktionsarten. Man kann sie nach den Arten des Inputs vom Menschen ordnen. Eingaben an die Maschine werden bei einem Computer normalerweise über die Tastatur oder mit der Maus gemacht. In Zukunft wird es weitere Möglichkeiten wie Touch-Screen, Sprachsteuerung, etc. geben.

Tastatureingaben können entweder geschriebene Befehle sein (urtümliche Interaktion mit einem Computer) oder aus Tastenkombinationen (z.B. „Ctrl + C“ für Kopieren) bestehen.

Heute werden Interaktionen vor allem mit der Maus vorgenommen. Mit ihr kann auf Elemente geklickt oder doppelgeklickt werden. Weiter kann eine Funktion beim Überfahren eines Elementes aktiviert werden.

Die Reaktionen des Computers sind mehrheitlich als Änderungen des Bildes, als Öffnen eines neuen Fensters, etc. bemerkbar. Es können aber auch Tonsignale, Ausdrücke, etc. sein.

5.2 Sinnvolle Interaktionen für eine Karte

Für eine Karte sind vor allem Interaktionen mit der Maus nützlich. Mit ihr kann auf der Karte durch anklicken oder überfahren eines beliebigen Objekts eine Aktion ausgelöst werden.

Mit der Interaktion wird meist eine Änderung des Kartenbildes angestrebt. Typischerweise können Kartenebenen ein- und ausgeschaltet werden, die Karte vergrößert, verkleinert oder verschoben werden.

Für die Webkarte vom Katasterplan werden neben den üblichen Karteninteraktionen (Zoom, Pan, Ebenen wechseln) folgende zwei weitere Interaktivitäten ausgewählt:

- Beim Überfahren eines Objektes werden in einem Textfeld Informationen dazu angezeigt
- Beim Klick auf ein Objekt öffnet sich ein zweites Browserfenster, indem alle Informationen zum Objekt in einer Tabelle angezeigt werden.

6 Analyse Ausgangsdaten

Der Katasterplan Morcote liegt digital im weit verbreiteten Shape-File Format vor.

6.1 Geometrie

Das Shape-File ist ein digitales Vektor-Datenformat für die Speicherung von Geometrien und den damit verknüpften Merkmalen. Es wurde von ESRI entwickelt und kann mit ArcView, ARC/INFO, ArcGIS und weiteren GIS-Systemen bearbeitet werden.

Dieses Geodatenformat ist aus drei einzelnen Files aufgebaut:

- *.shp: Die Datei mit den Geometrien der einzelnen Features.
- *.shx: In diesem File ist der Index der Geometrie gespeichert.
- *.dbf: Hier sind die Attribute der Geometrien gespeichert (dBASE-File).

Da das Shape-File nicht topologisch ist, unterstützt es keine räumlichen Beziehungen wie Nachbarschaft oder Areadefinitionen. Dies macht das Format einfacher, schränkt es aber bei komplexen räumlichen Analysen ein.

Der Katasterplan Morcote bildet einen Auszug aus dem Datensatz der Amtlichen Vermessung. Dieser basiert auf acht Hauptebenen (vgl. Tab. 1). Im vorliegenden Datensatz sind die Ebenen „Bodenbedeckung“ und „Einzelobjekte und Linienelemente“ vorhanden, wobei letztere in „flächige“ und „linienförmige“ aufgeteilt sind.

Ebenen
Fixpunkte
Bodenbedeckung
Einzelobjekte und Linienelemente
Höhen
Nomenklatur
Liegenschaften
Leitungen
Administrative Einteilung

Tab. 1: Ebenen der Amtlichen Vermessung (AV93)

6.2 Attribute / Sachdaten

Die Geometrien besitzen verschiedene Sachdaten (vgl. Tab. 2). Das Merkmal „Art“ wurde zusätzlich eingefügt und bezeichnet die Übersetzung des Attributes „Genere“ ins Deutsche.

Attribut	Beschreibung
Shape	Polygon, PolyLine
Sys_obid	Eindeutige ID
Origine	-1
Qualita	NULL
Genere	Art (Integerzahl als ID für „Genere_w“)
Provenie	NULL
Genere_w	Art (String, italienisch)
Provenie_w	NULL
Art	Art (String, deutsch), Übersetzung von „Genere_w“

Tab. 2: Attribute der Shape-Files

Das Attribut „Sys_obid“ ist über den ganzen Datensatz, über alle Ebenen, eindeutig. Daher eignet es sich sehr gut zur Verwendung als ID in einer Datenbank.

Die Merkmale „Qualita“, „Provenie“ und „Provenie_w“ sind bei allen Objekten leer. Deren Bedeutung ist unklar.

Zwischen den Attributen „Genere“ und „Genere_w“ besteht ein Zusammenhang: „Genere“ (vom Typ Integer) bildet eine ID für „Genere_w“ (vom Typ String).

7 Techniken interaktiver Webkarten im Internet

Das Internet besteht aus unzähligen, zu einem Netzwerk verbundenen Computern. Grundsätzlich kann man zwei Typen Rechner unterscheiden: Client und Server. Client sind Computer, welche Anfragen an einen (Web-)Server senden und von diesem eine Antwort bekommen (vgl. Abb. 2). Als Client wird ein normaler Arbeitscomputer eingesetzt. Server sind Rechner, auf denen spezielle Applikation laufen (Webserver wie Apache, MS Internet Information Server). Diese Anwendungen bearbeiten und beantworten die Anfragen, die sie von den Clients erhalten.

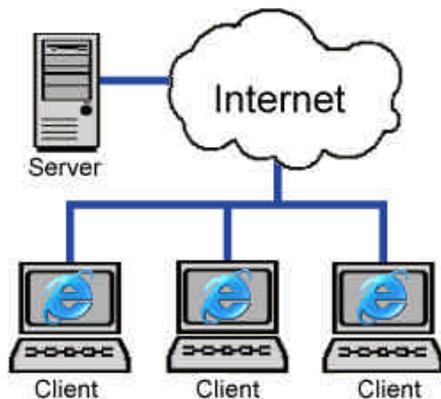


Abb. 2: Client-Server Architektur des Internets

Web-Technologien können hauptsächlich in zwei Gruppen aufgeteilt werden: Client-basierte und server-basierte Konzepte.

Bei client-basierten Lösungen werden Daten vom Server geladen und auf dem Client dargestellt. Manipulationen (Interaktionen) werden auf dem Client berechnet und angezeigt.

Im Gegensatz dazu werden bei server-basierten Konzepten alle Manipulationen auf dem Server berechnet und die Lösung an den Client übermittelt. Daraus resultiert vor allem bei gleichzeitigen Anfragen mehrerer Clients eine hohe Serverbelastung. Zudem ist diese Lösung stark abhängig von der Übertragungsrate des Netzes. Client-basierte Lösungen wiegen diesen Nachteil auf, jedoch sind dem Client Grenzen in der Funktionalität gesetzt.

7.1 HTML Hypertext Markup Language

HTML (Hypertext Markup Language) ist eine codierte Formatierungssprache. Diese wird standardmässig zum Erstellen von Dokumenten für das Internet, sowie für deren Erscheinen, verwendet. Sie erstellt aus einem einfachen Text ein vernetztes Gebilde, in dem sich Anwender durch Mausclicks bewegen können.

7.2 SVG Scalable Vector Graphics

Scalable Vector Graphics ist ein XML-basiertes Datenformat zur Beschreibung zweidimensionaler Vektorgraphiken. SVG-Graphiken können in jedem Browser dargestellt werden, vorausgesetzt das notwendige Plug-In¹ wurde installiert. Dieses wird mit den heutigen Browsern noch nicht standardmäßig mitgeliefert und muss zusätzlich heruntergeladen und installiert werden.

Im Jahre 2001 wurde SVG ein offizieller Standard des W3C². In nächster Zeit ist daher mit einer zunehmenden Verbreitung von SVG zu rechnen.

SVG basiert auf der weit verbreiteten und zukunftsweisenden Formatbeschreibungssprache XML. Die Graphiken werden als ASCII-Dateien abgespeichert und können einfach interpretiert werden. Die Bearbeitung von SVG kann mit einem normalen Texteditor erfolgen. Vorzugsweise wählt man einen spezialisierten XML-Editor (z.B. Altova XMLSpy³). In diesem kann anhand der SVG-DTD (Document Type Definition) gleichzeitig auch die Gültigkeit des Dokumentenaufbaus geprüft werden. Zusätzlich werden die entsprechenden Schlüsselwörter (Tags) speziell hervorgehoben. Alternativ kann auch ein umfangreiches Vektorgraphikprogramm (z.B. Jasc Webdraw oder Adobe Illustrator) verwendet werden.

7.2.1 Möglichkeiten

SVG bietet folgende Möglichkeiten:

- Grundlegende geometrische Objekte (Linien, Rechtecke, Kreise, Ellipsen, Polygone)
- Pfade (Linien aus verschiedenen Segmenten wie Gerade, Kreis- und Ellipsenbogen oder Splines)
- Text, für Suchmaschinen indexierbar
- Graphische Symbole für mehrmaligen Gebrauch definierbar
- Rasterobjekte einbinden
- Farben, Farbverläufe und Muster als Füllung für Objekte
- Filtereffekte für Objekte
- Interaktiv auf Objekte zugreifen und manipulieren
- Hyperlinks für Objekte definieren
- Animationen darstellen
- Clipping und Masking Funktionen
- Trennung Geometrie und Layout

¹ SVG-Viewer sind von verschiedenen Herstellern erhältlich z.B. von:

Adobe: <http://www.adobe.com/svg/main.html> (Zugriff: 27.01.2004) oder

Corel: http://www.smartgraphics.com/Viewer_prod_info.shtml (Zugriff: 27.01.2004)

² W3C: World Wide Web Consortium, <http://www.w3c.org> (Zugriff: 30.01.2004)

³ <http://www.altova.com> (Zugriff: 27.01.2004)

7.2.2 DOM Document Object Model

Das Document Object Model (DOM) ist ein plattform- und sprachunabhängiges Interface, welches Programmen und Skripten den dynamischen Zugriff auf den Inhalt, die Struktur und die Erscheinung eines Dokumentes ermöglicht. Ein Betrachter kann so eine Änderung an irgendeinem Objekt des Dokumentes bewirken und die aktualisierte Version wird ihm wieder präsentiert. In SVG kann mit JavaScript oder einer anderen Skriptsprache auf das DOM zugegriffen werden.

7.3 JavaScript

JavaScript ist eine client-seitige Programmiersprache. Da es eine Skript-Sprache ist, muss sie nicht kompiliert, sondern lediglich vom Browser interpretiert werden. Im Gegensatz zu reinem HTML bietet sich mit JavaScript die Möglichkeit, dynamische Webseiten zu generieren. In Verbindung mit SVG ergibt dies viele Möglichkeiten: Bei einem Klick auf ein Objekt in der Karte kann etwa eine JavaScript-Funktion, welche ihm eine andere Farbe zuweist, aufgerufen werden.

Spricht man von JavaScript im Allgemeinen, meint man eigentlich ECMAScript. Dies entspricht dem von der europäischen Organisation ECMA⁴ verabschiedeten Standard. Eine Standardisierung wurde notwendig, weil sich die JavaScript-Versionen von Netscape und Microsoft immer mehr unterschieden. Zwar sind sich die beiden Sprachen in weiten Teilen sehr ähnlich, trotzdem ergeben sich vor allem beim DOM-Zugriff grosse Differenzen.

7.4 PHP Hypertext Preprocessor

PHP ist eine verbreitete server-seitige Skriptsprache, welche speziell für Webseiten entwickelt wurde. PHP-Code wird in speziellen Start- und Endtags (`<?php`, bzw. `?>`) eingeschlossen und kann so inmitten eines HTML-Dokumentes platziert werden. Der Unterschied zu client-seitigen Skriptsprachen (z.B. JavaScript) besteht darin, dass der Code auf dem Server interpretiert wird. Dort werden alle PHP-Statements ausgeführt und erst das Ergebnis an den Client übermittelt.

Unter anderem eignet sich PHP auch zur Kommunikation mit einer server-seitigen Datenbank.

7.5 MySQL Datenbank

MySQL ist wohl die bekannteste und verbreitetste Internet-Datenbank. Sie kann auf verschiedenen Plattformen (v.a. Linux, Windows) einfach installiert werden und erreicht eine hohe Performance. MySQL ist eine freie Software, sie obliegt der GNU General Public License (GPL)⁵.

Aus Webapplikationen kann mit server-seitigen Skriptsprachen (z.B. PHP) auf MySQL zugegriffen und so ein effizienter Datenzugriff gewährleistet werden.

⁴ <http://www.ecma-international.org> (Zugriff: 27.01.2004)

⁵ GNU General Public License (GPL): siehe auch <http://www.gnu.org> (Zugriff: 27.01.2004)

7.6 Gewählte Möglichkeit

Für den Aufbau der Webkarte Morcote wurde das Konzept einer statisch-interaktiven Karte gewählt.

In einer HTML Seite wird die Karte als SVG-Graphik eingebettet. Neben SVG gibt es weitere Varianten, um Karten in Webseiten darzustellen (z.B. GIF, JPEG, CGM, Flash). Hier wurde die Technik SVG aus folgenden Gründen gewählt (vgl. auch Kap. 7.2):

- SVG ist ein Vektordatenformat und kann daher ohne Qualitätsverlust skaliert werden.
- Pfade ermöglichen die Darstellung von grundlegenden geometrischen Elementen und ermöglichen so eine geeignete automatisierte Generierung.
- SVG unterstützt Skriptsprachen (JavaScript) und lässt sich in HTML einbinden.
- Via DOM kann interaktiv auf Objekte zugegriffen werden und es sind Hyperlinks definierbar.
- SVG basiert auf XML, ist daher ein ASCII-File und kann einfach editiert werden.
- SVG als W3C-Standard für Vektorgraphiken ist ein zukunftsorientiertes Dateiformat.

Das proprietäre Vektorgraphikformat FLASH⁶ der Firma Macromedia figuriert als geeignete Alternative zu SVG. Das dazu notwendige Shockwave-PlugIn ist kleiner zum Herunterladen und heutzutage weiter verbreitet wie SVG. FLASH birgt jedoch einige tragende Nachteile:

- Binäres Dateiformat, welches nur mit der Autorensoftware „Flash“ von Macromedia editiert werden kann.
- Abhängigkeit von einer Firma (Macromedia).
- FLASH-Dateien (*.swf) sind grösser als SVG-Dateien.

Die Legende wird in HTML definiert und ist über JavaScript-Funktionen mit der Karte verbunden, was das Ein- und Ausblenden von Ebenen erlaubt.

Die Interaktivität kann durch JavaScript-Befehle (onClick, onMouseOver) gewährleistet werden.

Die Attribute der Shape-Files werden in eine MySQL Datenbank exportiert. PHP-Skripte bewerkstelligen den Datentransfer zwischen HTML und MySQL.

⁶ <http://www.macromedia.com>

8 Umsetzung

8.1 Untersuchung Konvertierungstools

Damit eine grösstmögliche Automatisierung erreicht werden kann, muss die Konvertierung der GIS-Daten in Webkartenformate möglichst einfach und automatisch ablaufen.

Gemäss dem zweigeteilten Kartenaufbau (vgl. Kap. 7.6) werden zwei Konvertierungstools benötigt. Eines zur Erstellung der SVG-Datei mit den Geometrien, ein zweites mit den Sachdaten zum Import in die Datenbank.

Die im Shape-File vorliegenden Daten können in allen üblichen GIS-System betrachtet werden. Für diese Untersuchung wurde das in der Praxis weit verbreitete ESRI ArcView als Ausgangs-GIS gewählt.

8.1.1 GIS zu SVG

Für den Export der Geometrien aus ArcView wurde ein geeignetes Skript gesucht. Auf der ESRI-Homepage finden sich im Downloadbereich viele verschiedene Tools⁷. Aus diesen wurde eines ausgewählt, welches folgende Bedürfnisse möglichst gut erfüllt:

- Es muss ein SVG-File erstellt werden. In diesem müssen die verschiedenen Ebenen aufgeteilt sein, so dass in der Webkarte Ebenen ein- und ausgeschaltet werden können. Dazu müssen die einzelnen Objekte (path-Elemente) in Gruppen zusammengefasst werden.
- Die Interaktivität muss gewährleistet sein. Zu jedem Objekt muss eine JavaScript-Funktion aufgerufen werden können.
- Das Tool sollte auf die besonderen Bedürfnisse angepasst werden können.

Die Resultate sind in Tab. 3 zusammengefasst.

Tool	SVG	Interaktivität	Anpassbarkeit	Weiteres
Shp2svg	✓	✓	✓	<ul style="list-style-type: none"> • Avenue-Skript • Attribute in JavaScript • OpenSource
MapViewSVG	✓	✓	✗	<ul style="list-style-type: none"> • ArcView-Extension • Attribute in JavaScript
Neten'Map	✓	✓	✗	<ul style="list-style-type: none"> • ArcView-Extension • Attribute in JavaScript

Tab. 3: Vergleich verschiedener SVG-Export Tools

8.1.2 GIS zu Datenbank

Neben den Geometriedaten müssen auch die Sachdaten exportiert werden. Diese sollten in einer MySQL-Datenbank verfügbar sein. Analog zum Export zu SVG gibt es im Internet auch für den Export in MySQL verschiedene Tools. Folgende Anforderungen wurden an das zu wählende Tool gestellt:

⁷ <http://arcscripts.esri.com> (Zugriff: 02.02.2004)

- Die Sachdaten müssen in der Datenbank derart vorliegen, dass später mit den PHP-Skripts die gewünschten Informationen abgefragt werden können.
- Die Geometriedaten müssen nicht in der Datenbank vorliegen.

Die Resultate sind in Tab. 4 zusammengefasst.

Tool	Details
MySQL Connection	<ul style="list-style-type: none"> • MySQL als Datenquelle in ArcView verfügbar • „shpmysql“ zum Export
OpenSVGMapServer	<ul style="list-style-type: none"> • „tomysql“ zum Export • Dynamische SVG-Generierung aus MySQL • Attribute in JavaScript

Tab. 4: Vergleich verschiedener SQL-Export Tools

8.2 Auswahl Konvertierungstools

8.2.1 GIS zu SVG

Die drei in die nähere Auswahl einbezogenen Tools (vgl. Tab. 3) erfüllen alle die ersten beiden Kriterien. Als einziges kann jedoch das Tool „Shp2svg“ auf die speziellen Bedürfnisse angepasst werden. Es ist ein Avenue-Skript und kann problemlos editiert werden. Die zwei anderen Skripte sind ArcView-Extensions. Der Quellcode dieser kann nicht eingesehen werden und so auch nicht bearbeitet werden.

Mit Änderungen am „Shp2svg“-Skript können alle gestellten Forderungen erfüllt und ein automatischer Export gewährleistet werden.

8.2.2 GIS zu Datenbank

Aus den beiden Tools in Tab. 4 wurde „MySQL Connection“ ausgewählt. Die Sachdaten werden gemäss den Ebenen im ArcView in Tabellen abgelegt und können über SQL aus den PHP-Skripts einfach abgefragt werden. Das Tool „OpenSVGMapServer“ legt die Attribute nicht in der Datenbank aber, sondern in JavaScript-Dateien. In der Datenbank befinden sich aber die Geometriedaten. Der OpenSVGMapServer erzeugt die SVG-Dateien aus den Geometrien der MySQL-Datenbank.

„MySQL Connection“ eignet sich für den Datenexport in diesem Projekt. Es exportiert zwar auch die Geometriedaten. Dies wäre allerdings nicht erforderlich. Aufgrund der geringen Speichergrosse kann dies aber gut vernachlässigt werden. Da das Tool eine ArcView-Extension ist und der Quellcode nicht vorhanden ist, kann dieser Punkt auch nicht selber ausgemerzt werden.

9 Praktische Umsetzung

9.1 SVG-Karte

Es bestehen verschiedene Möglichkeiten eine SVG-Graphik zu erzeugen. Sie kann mit einem Texteditor erstellt oder aus einem Graphikprogramm exportiert werden. In Berücksichtigung, dass die Karte später regelmässig mit möglichst minimalem Aufwand aktualisierbar sein soll, wird auf deren automatische Erstellung grossen Wert gelegt.

Auf der Karte selber werden die drei Ebenen Bodenbedeckung, flächige Einzelobjekte und linienförmige Einzelobjekte dargestellt. Diese werden als thematische Legende, eingefärbt nach dem Attribut „Art“, umgesetzt (vgl. Abb. 3). Dabei ist auf eine gute Lesbarkeit bei jeder möglichen Zoomstufe zu achten. Es wäre sinnvoll die Karte mit adaptivem Zooming zu versehen, d.h. bei verschiedenen Zoomstufen würden verschieden stark generalisierte Karten dargestellt werden. Da dies jedoch einem beträchtlichen Mehraufwand und nicht unbedingt dem Ziel dieser Arbeit entspricht, wurde nur eine Generalisierungsstufe gewählt.

Das SVG-File ist folgendermassen gegliedert (vgl. Codeblock 1). Die einzelnen Ebenen des ArcView-Projekts sind im SVG als Gruppen wiederzuerkennen. Den Elemente einer Gruppe werden über die Style-Eigenschaft die entsprechenden Farben, Linienstärken, etc. zugewiesen. Die Gruppen sind nach dem Painter-Algorithmus definiert. Im SVG-File kommen an erster Stelle die Elemente der Ebenen, die zuunterst liegen. Die einzelnen Objekte sind als Pfade abgespeichert, welche eine ID besitzen, sowie eine onclick- und onmouseover-Funktion (vgl. Kap. 9.2.1) bieten.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN"
"http://www.w3.org/2000/svg10-20000303-stylable" [
<!ENTITY styleAkker_wiese_weide "fill:rgb(220,206,0);fill-
rule:evenodd;stroke:rgb(0,0,0);stroke-width:0.10000000pt;stroke-antialiasing:true;">
[...

<!-- Generator: ArcView Avenue script shp2svg angepasst durch cho, vtb ws2003/04 -->
<svg id="map" width="666.00000000px" height="728.00000000px" viewBox="0 0
6660000.0000000 7280000.0000000">
<desc>Morcote</desc>
<rect x="1" y="1" width="6660000.00000000" height="7280000.00000000"
fill="#ffffff" stroke="black" stroke-width="2" onclick="info(99)" onmouse-
over="overinfo(99, ' ')" />
<g id="Akker_wiese_weide" style="&styleAkker_wiese_weide;">
<path id="81653.00000" onclick="info(81653.00000, 'Akker_wiese_weide')" onmouseo-
ver="overinfo(81653.00000, 'Akker_wiese_weide')" d="
M 6180663 5079048 L 6146814 5055086 L 6192201 4990761 L 6228281 5015765 L 6184159
5077759 L 6180663 5079048 z
" />
```

Codeblock 1: Ausschnitt aus dem SVG-Code der Karte

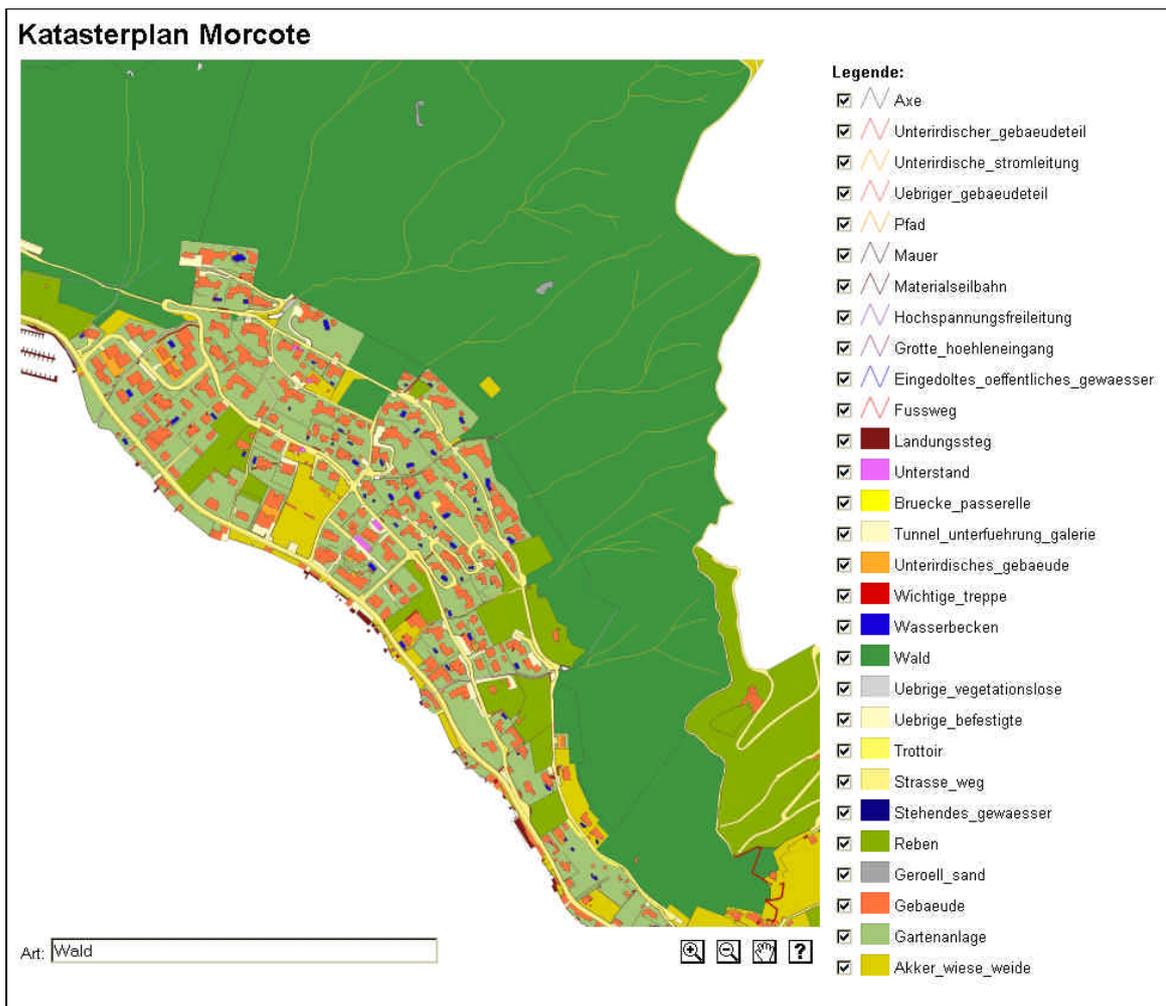


Abb. 3: Webkarte Morcote

9.2 Interaktivität

9.2.1 JavaScript

Beim Überfahren eines Kartenobjektes sollen dazugehörige Informationen in einem Textfeld erscheinen (vgl. Kap. 2). Beim vorliegenden Datensatz eignet sich das Attribut „Art“, um angezeigt zu werden.

Zusätzlich öffnet sich bei einem Klick auf ein Objekt ein neues Browserfenster, in dem alle Attribute angezeigt werden.

Im SVG-File wird dazu jeder Pfad mit einem „onClick“- sowie einem „onmouseover“-Event versehen:

```
<path id="81713" onclick="info(81713,'Acker_wiese_weide')" onmouseover="overinfo(81713,'Acker_wiese_weide')" d="M 5537194 5807235 L 5542347 5804486 L 5539834 5798808 L 5558565 5784937 L 5571823 5802561 L 5550564 5819807 L 5548975 5821362 L 5538739 5809090 L 5537194 5807235 z" />
```

Codeblock 2: Code-Beispiel eines Paths des SVG-Files

Beim Klick auf das Objekt („onclick“) wird die JavaScript-Funktion „info()“ aufgerufen. Ihr werden die zwei Parameter „81713“ und „Acker_wiese_weide“ übergeben. Beim Überfahren („onmouseover“) wird die Funktion „overinfo()“ mit den gleichen Parametern aufgerufen.

Die Funktion „info()“ (vgl. Codeblock 3) übernimmt diese beiden Parameter und öffnet mit „window.open“ ein neues Fenster, in welches die Datei „atttable.php“ geladen wird. Die beiden Parameter werden der URL angehängt und auf diesem Weg der PHP-Datei übergeben. Zudem überprüft die Funktion, ob auf ein Objekt geklickt wurde. Falls dem nicht so ist, beträgt der Wert des ID-Parameters „99“ (Beim Klick auf das Hintergrundrechteck). Dann wird ein eventuell offenes Attributfenster geschlossen.

```
function info(id,theme) {
  switch(id) {
    case 99:
      fensterinfo.close();
      break;
    default:
      fensterinfo=window.open("atttable.php?id="+id+"%"+theme+", "Infos",
"width=390,height=740,menuBar=0,scrollBars=1,resizable=1,top=10,left=10" );
      fensterinfo.focus();
      break;
  }
}
```

Codeblock 3: JavaScript Funktion beim Klick auf ein Objekt

Bei einem „Mouseover“ wird der Wert des zweiten Parameters übernommen und via DOM in das Formulartextfeld der HTML-Datei geschrieben (vgl. Codeblock 4). Eine Variante, in welcher die Infos direkt aus MySQL ausgelesen werden, findet sich auskommentiert (//) (vgl. Kap. 11.2.4).

```
function overinfo(id,theme) {
  document.id.label.value=theme;
  //Attribut aus mysql übernehmen:
  //switch(id) {
  //  case 99:
  //    document.id.label.value="";
  //    break;
  //  default:
  //    fensterforminfo=window.open("overinfo.php?id="+id+"%"+theme+", "nix",
"width=390,height=740,menuBar=0,scrollBars=1,resizable=1,top=10,left=10" );
  //    break;
  // }
}
```

Codeblock 4: JavaScript Funktion beim Überfahren eines Objekts

Die JavaScript-Funktionen sind in der externen Datei scripts.js definiert. Diese wird im Header des HTML-Files eingebunden.

9.2.2 PHP

Zur Kommunikation mit der Datenbank wird das server-seitige PHP benutzt. Beim Klick auf ein Objekt wird mit JavaScript eine PHP-Datei (atttable.php) in das neu geöffnete Fenster geladen.

In diesem PHP-Skript wird eine Verbindung mit MySQL hergestellt. Danach werden die übergebenen Parameter ausgewertet und eine SQL-Abfrage an die Datenbank gesandt. Returniert wird ein Array mit den Resultaten. So wird ein HTML-Dokument (vgl. Abb. 4) aufgebaut, in welchem eine Tabelle alle Attribute des aktiven Objektes darstellt. Zudem wird ein Formular erzeugt, mit welchem Attribute geändert, erstellt und gelöscht werden können. Diese Änderungen mit werden weiteren PHP-

Skripten (change.php, neueattr.php, delete.php) direkt in der Datenbank vorgenommen und sind dauerhaft gespeichert.

Attribute von: Wald

FeatureID	23
XMin	713663.02
XMax	714454
YMin	86938.35
YMax	88132.64
Sys_obid	82609.00000
Origine	-1.00000
Qualita	
Genere	17
Provenie	0
Genere_w	boscata.bosco_fitto
Provenie_w	
Art	Wald

Attribute ändern:

Attribut wählen

Neues Attribut hinzufügen:

neues Attribut...

Attribute löschen:

Attribut wählen

[Fenster schliessen](#)

Abb. 4: Attributfenster

9.3 Konvertierung der Shape-Files

9.3.1 Shp2Svg Tool

9.3.1.1 Avenue-Skript

Für das vorliegende Projekt wurde das Tool shp2svg⁸ ausgewählt. Dieses Avenue-Skript unterliegt den Lizenzbedingungen der GNU General Public License⁹ und ist somit kostenlos erhältlich.

Mit einem Avenue-Skript kann auf alle Elemente eines ArcView-Projektes zugegriffen werden. Der Zugriff erfolgt über das DOM-Modell von ArcView, welches vergleichbar mit jenem von SVG (vgl. Kap. 7.3) ist. Das shp2svg-Skript ist relativ leicht zu verstehen und kann den eigenen Bedürfnissen entsprechend angepasst werden.

⁸ <http://www.carto.net/projects/shp2svg> (Zugriff: 30.01.2004)

⁹ <http://www.gnu.org/copyleft/gpl.html> (Zugriff: 30.01.2004)

Nachfolgend einige Angaben zum shp2svg-Skript:

Export der Features:

- Punkte: werden als Kreise mit einem Radius von 2 Pkt, keinem Strich und einer Füllung entsprechend der Vordergrundfarbe des ArcView-Themes exportiert.
- Linien: werden als ungeschlossene SVG-Pfadobjekte mit keiner Füllung, als ausgezogener Strich (Breite 0.5 Pkt) und einer Strichfarbe entsprechend der Vordergrundfarbe des ArcView-Themes exportiert.
- Polygone: werden als geschlossene SVG-Pfadobjekte mit einer Füllung, einer Strichbreite und -farbe entsprechend der Vordergrundfarbe des ArcView-Themes und einem ausgezogenem Strich exportiert.

Limiten:

Die SVG-Formatierung ist einfach gehalten. Es können weder Schraffuren, gestrichelte Linien, Rasterbilder noch Featurebeschriftungen übernommen werden.

Das Hauptaugenmerk bei der Bearbeitung des Skripts besteht darin, dass eine Aktualisierung der Kartengraphik mit möglichst wenig Aufwand vonstatten gehen kann. Im Avenue-Skript können so alle benötigten Dateien für die Webkarte erstellt werden. Hauptsächlich ist dies natürlich die SVG-Graphik, welche direkt aus dem View von ArcView übernommen wird. Aber auch die zugehörige HTML-Datei und die PHP-Skripts werden erstellt.

9.3.2 shpmysql Tool

Die Attribute sollen in die Datenbank MySQL exportiert werden. Dazu wurde eine ArcView Extension gewählt: shpmysql¹⁰. Dieses Tool ist ein Teil der ArcView-Extension „MySQL Connection“ mit welcher MySQL-Daten in ArcView verwendet werden können. Inbegriffen ist eine Möglichkeit, um Daten in ein SQL-File zu exportieren. Da das ganze Tool eine ArcView-Extension ist, kann es leider nicht bearbeitet und auf die eigenen Bedürfnisse angepasst werden. So werden auch die Geometriedaten in MySQL exportiert, was im vorliegenden Fall nicht notwendig ist.

Für jeden Legendeneintrag im View-Fenster wird ein SQL-File erstellt, welches danach in MySQL eingelesen werden kann (vgl. Kap. 10.3).

9.4 Webserver

Als Webserver wurde die freie Software XAMPP¹¹ in der Version 1.2 verwendet. Dies ist eine Distribution von Apache, MySQL, PHP und Perl. Diese Tools können mit diesem Paket einfach miteinander installiert werden und sind automatisch aufeinander abgestimmt.

¹⁰ <http://arcscrips.esri.com/details.asp?dbid=12896> (Zugriff: 02.02.2004)

¹¹ <http://www.apachefriends.org/xampp.html> (Stand: Januar 2004)

Der eigentlich Webserver ist ein Apache 2.0.48 (für Win32). Zusätzlich wird PHP 4.3.4, Perl 5.8.0 und die Datenbank MySQL in Version 4.0.16 installiert. Ein benutzerfreundliches Front-End für MySQL (PHPMyAdmin) wird mitgeliefert.

Natürlich können auch andere Distributionen von Webservern verwendet werden, bzw. bereits installierte benutzt werden.

Für die Webkarte Morcote ist ein Webserver mit PHP-Unterstützung sowie die Datenbank MySQL erforderlich.

10 Anleitung zum Export

Zuerst müssen die gewünschten Shape-Files in ArcView in einer neuen View dargestellt werden. Dabei sind folgende Punkte zu beachten:

- Thematische Legenden können nicht übernommen werden. Jedoch können die Shape-Files anhand von Abfragen in einzelne Dateien aufgeteilt und diese dann als Ebenen im View angezeigt werden.
- Bei der Benennung der Shape-Files ist darauf zu achten, dass keine Leer-, und auch keine Sonderzeichen gewählt werden. Diese werden in MySQL fehlerhaft abgespeichert.
- Die Grösse des View-Fensters bestimmt die Grösse der zu erstellenden SVG-Graphik.
- Ins SVG-File wird nur die Füllfarbe übernommen. Strichstärke, Strichart, Schraffuren, etc. haben keinen Einfluss.

10.1 MySQL-Export

Beim erstmaligen Verwenden des shpmysql-Tools ist dieses in ArcView zu registrieren. Dazu muss die Datei shpmysql.avx in das Extension-Verzeichnis der ArcView-Installation kopiert werden.

Diese Extension (MySQL Connection) muss vor dem Einsatz des Tools in ArcView geladen werden.

Nachdem der View erstellt worden ist, kann aus dem neuen Menüpunkt „MySQL“ mit „Export SQL File“ der Export gestartet werden. Über ein Menü kann der zu exportierende View ausgewählt werden. Danach erscheinen alle geladenen Shape-Files dieses Views, aus denen ausgewählt werden kann (vgl. Abb. 5). Weiter ist die Anzahl Dezimalstellen der Koordinaten anzugeben. Dies ist für dieses Projekt nicht obligatorisch, da die in MySQL gespeicherte Geometrie nicht benötigt wird. Es folgen Abfragen über die Version von MySQL und über Tabellendefinitionen. Diese Angaben können wie vorgeschlagen übernommen werden. Dann kann der Filename und Speicherort ausgewählt werden. Bei erfolgreichem Export erscheint eine Bestätigungsmeldung. Diese ganze Prozedur muss für jeden Legendeneintrag wiederholt werden.

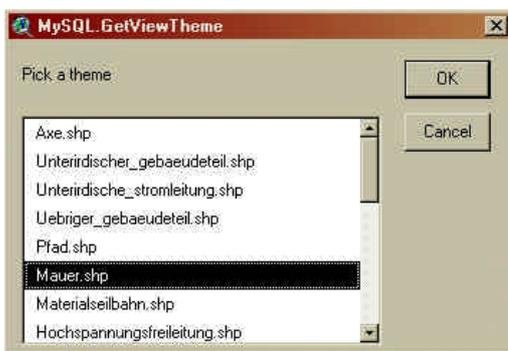


Abb. 5: Menü aus dem MySQL-Export

10.2 SVG-Export

Der SVG-Export erfolgt mit dem Avenue-Skript „shp2svg“, welches speziell auf die Bedürfnis dieser Webkarte angepasst wurde.

Dabei muss das Skript (to_svg.apr) zuerst in das ArcView-Projekt importiert werden. Dies erfolgt über das Menü „Project“ → „Import...“.

Vor dem Start des Exportvorganges muss darauf geachtet werden, dass in der Legende des View-Fensters alle Shape-Files selektiert sind. Der Titel der aktuellen View wird später als Titel der Webkarte übernommen, weshalb er eine sinnvolle Bezeichnung tragen sollte.

Danach wird das Skript im View-Fenster über „View“ → „Convert to SVG“ gestartet. In den nachfolgenden Dialogen können die Namen sowie Verzeichnisse der zu erstellenden Dateien benannt werden. Dabei ist zu beachten, dass alle Dateien im selben Verzeichnis erstellt werden sollten.

Weiter ist eine Funktion eingebaut, mit welcher das SVG-File mit GZIP¹² komprimiert werden kann. Es kann mit einem Komprimierungsfaktor von ca. 60 % gerechnet werden.

Nachfolgend ist zur Erstellung der PHP-Dateien der Name der vorher mit „shpmysql“ erstellten Datenbank einzugeben. Falls diese nicht übereinstimmen, können die PHP-Skripte keine Verbindung mit MySQL aufnehmen.

Anschliessend werden die 4 PHP-Skripte, welcher zur Kommunikation mit MySQL notwendig sind, automatisch erstellt. Darin werden projektspezifische Daten wie der Datenbanknamen gespeichert. Gleichzeitig wird auch die HTML-Hauptdatei, welche die SVG-Graphiken anzeigt, erzeugt. Da dies automatisch geschieht, müssen diese Dateien später nicht mehr weiter editiert werden und sind bereit für den Einsatz im Webprojekt.

10.3 Erstellung Webkarte

Die mit dem SVG-Export erstellten Dateien sind nun in ein Verzeichnis auf dem Webserver zu kopieren.

Zusätzlich zu ihnen werden folgende bereits vorhandene Dateien benötigt:

- Vier Bilddateien: help.gif, panoff.gif, zoominoff.gif und zoomoutoff.gif
- Die JavaScript-Datei: scripts.js
- Die HTML-Datei projektinfo.htm. Diese muss natürlich je nach Projekt abgeändert werden.

Schliesslich müssen die erzeugten SQL-Dateien in die Datenbank importiert werden. Am besten wird dazu die Graphische Benutzerschnittstelle „PHPMyAdmin“ der XAMPP-Distribution verwendet. Alternativ kann auch auf Kommandozeilenbasis gearbeitet werden. Zuerst muss eine neue Datenbank erstellt werden. Der Name hat mit dem im SVG-Export angegebenen übereinzustimmen. Danach können die SQL-Skripte importiert werden. Mit Vorteil erstellt man ein neues SQL-File, in das man alle einzelnen SQL-Dateien hineinkopiert. Dabei ist zu beachten, dass die Tabelle „geometry_columns“ nur einmal erstellt wird.

¹² <http://www.gzip.org/> (Zugriff: 27.01.2004)

11 Diskussion der Lösung

11.1 Probleme

11.1.1 Ausgangsdaten

Die Daten des Katasterplans Morcote sind in italienischer Sprache verfasst. Zuerst wurde ein neues Attribut „Art“ eingeführt, welches die deutsche Übersetzung des italienischen „Genere“ umfasst (vgl. Kap. 6.2). Dies kann ohne grosse Probleme mit ArcView an den Shape-Files vorgenommen werden.

11.1.2 Auswahl des Export-Tools

Der grösste zeitliche Aufwand erforderte die Suche nach einer möglichst geeigneten Konvertierungsmöglichkeit der Daten in das Format SVG sowie MySQL. Es existierten konkrete Anforderungen, welche das Tool beherrschen sollte. Hauptsächlich sollte es ein SVG-File erzeugen können, in welchem die Geometrien und die ID's gespeichert sind. Zudem sollten die OnClick- und OnMouseOver-Elemente enthalten sein. Auf der ESRI-Homepage¹³ sind unzählige Arc-Skripts zu finden. Davon wurden mehrere heruntergeladen, installiert und ein Exporttest vorgenommen. Es erwies sich als recht schwierig die Anforderungen zu erfüllen. Viele Tools beherrschen den reinen Export, aber die konkreten Bedürfnisse dieses Projektes deckt keines lückenlos ab. Als Lösung wurde ein Avenue-Skript ausgewählt, welches einen Grossteil der Bedürfnisse zu erfüllen vermag. Dieses Skript konnte danach manuell editiert werden, sodass möglichst alle Ansprüche erfüllt werden konnten (vgl. Kap. 9.3.1).

Die Anforderungen an das MySQL-Skript waren weniger hoch und ein geeignetes Tool war rasch gefunden (vgl. Kap. 9.3.2). Allerdings werden die Geometrien auch mitexportiert und in MySQL abgespeichert, was eigentlich nicht nötig wäre. Angesichts des geringen Speicherbedarfs kann aber über diesen Nachteil hinweggeschaut werden. Zusätzlich wäre eine Möglichkeit zur Exportierung aller Ebenen auf einmal wünschenswert. Da das Tool eine ArcView Extension ist, wäre zur Änderung die Offenlegung des Quellcodes notwendig, was zur Zeit nicht gegeben ist.

11.1.3 Einarbeitung in neue Internetsprachen

Zur Entwicklung einer Webkarte wurde ein Webserver aufgesetzt, auf welchem PHP installiert wurde. Die Installation dieser Komponenten ging weitgehendst automatisch vor sich und bereitete keine Probleme.

Schwieriger war die Programmierung der Schnittstelle zwischen MySQL und HTML bzw. SVG, weil der Autor ihm bisher unbekannte Techniken anwendete. Hier kam die mächtige server-seitige Skriptsprache PHP zum Einsatz. Bald konnten die wichtigsten Funktionen wie Ausgabe von HTML-Text, Behandlung von Variablen, Verbindung mit MySQL realisiert und nach dem Prinzip „Try and Error“ nach

¹³ <http://arcscrips.esri.com/> (Zugriff: 28.01.2004)

und nach alle Skripte erstellt werden. Auf dem Internet finden sich viele Beispiele und gute Nachschlagewerke¹⁴.

11.2 Kritische Betrachtung

11.2.1 Aktualisierung

Ein Ziel der Webkarte war, dass sie auf möglichst einfache Art und Weise aktualisierbar sein sollte.

In der vorliegenden Arbeit ist dies insbesondere für den SVG-Export gut gelungen. In ArcView können die Shape-Files in den schon bestehenden View geladen werden. Die Anordnung der verschiedenen Ebenen, sowie die Einfärbung bleibt beim Export erhalten. Mit dem Export-Tool „shp2svg“ sind in wenigen Schritten die erforderlichen Dateien erstellt. Diese müssen nur noch auf den Webserver kopiert werden.

Der MySQL-Export erfordert mehr Arbeit: Hier muss jede Ebene einzeln exportiert werden. Falls der Quellcode der ArcView-Extension vorhanden wäre, könnte dieser sicherlich im Sinne einer weiteren Automatisierung entsprechend angepasst werden.

11.2.2 Nutzung für andere Projekte

Weiter bestand die Absicht, dass der ganze Karten-Export und Aufbau der Webkarte auch für andere Daten und Projekte möglich ist. In ArcView können beliebige Daten geladen und diese nachher mit den beschriebenen Tools exportiert werden. An der Webkarte selber sind nachher nur noch Feinheiten anzupassen. Falls ein Projekt mit vollkommen verschiedenem Aufbau exportiert werden sollte, ist auch eine Änderung des Avenue-Skripts denkbar.

11.2.3 Performance

Bei der Entwicklung einer Internet-Applikation muss stets die beschränkte Übertragungsgeschwindigkeit beachtet werden. Im vorliegenden Fall ist dies insbesondere beim erstmaligen Laden der Seite von Bedeutung. Hier wird das SVG-File vom Server auf den Client übertragen. Das SVG-File vom Katasterplan Morcote ist ca. 2.8 MB, in komprimierter Form immer noch ca. 1 MB gross. Die Ladezeit mit einer zur Zeit noch typischen Übertragungsrate von 64 KiloBit pro Sekunde (ISDN) beträgt damit immerhin ca. 2 Minuten. Im Moment zeichnet sich mit dem rasanten Aufkommen der Breitbandanschlüsse (ADSL, Cablecom) eine schnellere Übertragungsrate ab. Trotzdem ist dieser Aspekt weiterhin bei jeder Webapplikation zu berücksichtigen.

Falls die Geometriedaten in der Datenbank gespeichert werden, wäre das Problem der Übertragungsgeschwindigkeit besser gelöst. Es könnte ein kleineres SVG-File, welches nur den gewünschten Ausschnitt enthält, generiert werden. Dies wäre dann um ein Vielfaches kleiner und dementsprechend schneller übertragen.

¹⁴ Zum Beispiel: <http://www.selfphp.info/> (Zugriff: 28.01.2004)
<http://www.teamone.de/selfhtml> (Zugriff: 28.01.2004)

11.2.4 Update von Attributen

Die entwickelte Webkarte bietet die Möglichkeit, in einem zweiten Browserfenster Attribute des ausgewählten Objektes zu ändern, neue zu erstellen oder bestehende zu löschen. Diese Änderungen wirken sich nur auf der MySQL-Datenbank aus. In der Kartengraphik (SVG) ändert sich aber nichts. Zum Beispiel sollte sich bei einer Änderung des Attributs „Art“ auch die entsprechende Farbe in der SVG-Graphik anpassen.

Beim Überfahren eines Objekts werden die im SVG-File gespeicherten Daten (Themennamen) im Textfeld angezeigt. Eine Variante, welche diese Daten direkt aus der Datenbank ausliest, wurde auch realisiert. In der Datei scripts.js ist in der Funktion „overinfo()“ diese Möglichkeit auskommentiert. Das dazu notwendige PHP-Skript (overinfo.php) wird automatisch erstellt. Diese Variante büsst aber gegenüber der einfacheren, aktivierten Variante an Performance ein. Das begründet sich mit der massiv erhöhten Client-Server-Kommunikation.

11.3 Erweiterungsmöglichkeiten

11.3.1 Kartenfunktionen

Das Hauptaugenmerk dieser Arbeit wurde auf die Möglichkeit der Attributänderungen und eine möglichst einfache Aktualisierung gelegt. Deshalb wurden die Kartenfunktionen auf das Wesentliche beschränkt. Mit den heutigen SVG-Viewern ist eine Möglichkeit des Zooms und des Pannens bereits im Viewer miteingebaut.

Als Erweiterung wäre die Programmierung eigener, auf die speziellen Bedürfnisse dieser Karte abgestimmte Zoom- und Pan-Tools denkbar. Beispiele dazu finden sich in diversen Projekten auf dem Internet.

Auch eine Übersichtskarte wäre wünschenswert. Auf ihr könnten jederzeit die Position und Grösse des Hauptkartenausschnitts erkannt werden.

Eine weitere Möglichkeit wäre eine Funktion, um die Karte auszudrucken. Momentan kann das über die Druckmöglichkeit des Browsers gemacht werden. Sinnvoll wäre aber ein schöner, formatfüllender Ausdruck mit Integration der Legende, Kartenmassstab, Nordpfeil, etc.

Speziell in der mehrsprachigen Schweiz ist eine Möglichkeit zur Umschaltung der Sprache wünschenswert. Dies wäre kein grosses Problem. Alle Dateien könnten im Hinblick auf eine Sprachumschaltung angepasst werden. Eine Möglichkeit zur Umschaltung wäre auf der Hauptseite zu integrieren, wie dies auf vielen Webseiten realisiert ist.

11.3.2 Browserabhängigkeit

Die in dieser Arbeit erstellte Webkarte wurde nur auf dem Microsoft Internet-Explorer getestet.

Dieser Browser ist momentan mit über 90% der am meisten verwendete. Einige Internetnutzer benutzen aber dennoch Alternativprodukte (Netscape/Mozilla, Opera, u.a.). Damit eine Webseite auf al-

len Browsertypen funktioniert, muss sie darauf getestet und mit allfälligen Erweiterungen versehen werden.

11.3.3 Anpassen der Graphik nach Attributänderungen

Wie in Kap. 11.2.4 angesprochen, wird die Kartengraphik nach Attributänderungen nicht geändert.

Mit einer server-seitigen Architektur könnte dieses Problem gelöst werden. Dies würde dem Prinzip einer echt dynamischen Karte (vgl. Kap. 4.2.2) entsprechen. Die Geometrien sind in einer Datenbank abgelegt. Bei einer Anfrage wird das SVG aus diesen Geometriedaten erstellt und übermittelt. Bei einer Attributänderung wird das SVG neu generiert und angezeigt.

11.3.4 Analysemöglichkeiten

Im Hinblick auf ein Web-GIS wären Analysefunktionen wünschenswert. Die dazu notwendige Datengrundlage wäre mit der MySQL-Datenbank geschaffen. Einfache Suchmöglichkeiten nach Strassen, Hausnummern, Eigentümern, etc. könnten ohne allzu grossen Aufwand realisiert werden. Interessant wären auch komplexere GIS-Analysemöglichkeiten wie Bufferzonen oder Nachbarschaftsbeziehungen.

12 Schlusswort

Für mich war die bearbeitete Aufgabenstellung eine echte Herausforderung. Der Reiz bestand darin, eine aktuelle Thematik mit zukunftsweisenden Technologien umzusetzen.

Die Präsentation von Geodaten auf dem Internet fand in den letzten Jahren eine enorme Verbreitung. Die Öffentlichkeit ist daran sehr interessiert und für die Datenerheber ist dies ein weiterer Bestandteil ihrer Arbeit geworden. Bei vielen Webkarten aus der Praxis ist das Problem der automatischen Aktualisierung unbefriedigend gelöst. Zudem gibt es kaum Beispiele, in welchen interaktiv Daten dauerhaft geändert werden können. Diese Arbeit sollte vor allem in diesen Bereichen Ansätze aufzeigen und beweisen, dass dies mit den heutigen Technologien sehr wohl möglich ist.

Ich habe mich in mir bisher unbekanntem Technologien wie PHP, MySQL, Avenue-Skripts eingearbeitet. Dies war anfangs mit Schwierigkeiten und Problemen verbunden. Doch das hatte auch Vorteile, so lernte ich immer besser und schneller die richtigen Informationen aus den verschiedensten Quellen zusammenzusuchen.

Trotz allen aufgetretenen Schwierigkeiten hat mir diese Arbeit sehr viel Spass bereitet und mein Interesse an der Geoinformatik sowie der Kartographie gefestigt. Die letzten 14 Wochen hinterlassen einige wertvolle Erfahrungen und Einsichten, die in der bevorstehenden beruflichen Laufbahn bestimmt von Nutzen sein werden.

13 Literaturverzeichnis

Allgemeine Internetquellen: (Zugriff: 27.01.2004)

<http://selfhtml.teamone.de>

<http://www.selfphp.info>

<http://www.mysql.com>

<http://www.carto.net>

<http://www.svg.org>

<http://www.esri.com>

Weitere Internetquellen sind im Text als Fussnoten vermerkt.

14 Abbildungs-, Tabellenverzeichnis

Abbildungen:

Abb. 1: Kategorien von Webkarten	8
Abb. 2: Client-Server Architektur des Internets.....	13
Abb. 3: Webkarte Morcote	20
Abb. 4: Attributfenster	22
Abb. 5: Menü aus dem MySQL-Export	25

Tabellen:

Tab. 1: Ebenen der Amtlichen Vermessung (AV93)	11
Tab. 2: Attribute der Shape-Files	12
Tab. 3: Vergleich verschiedener SVG-Export Tools.....	17
Tab. 4: Vergleich verschiedener SQL-Export Tools	18