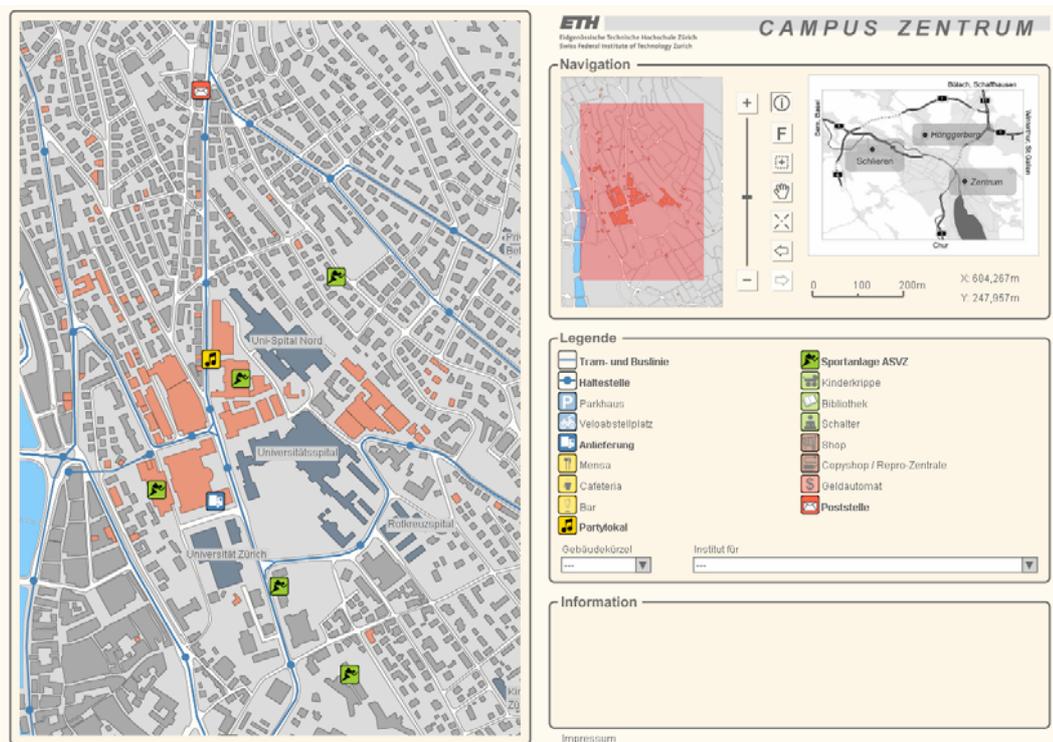


Vertiefungsblock Kartografie WS 2006/07**Interaktives Campusinformationssystem
für ETH Zentrum**

Autorin: Kathrin Sigrist

Leitung: Prof. Dr. Lorenz Hurni, Institut für Kartografie
Betreuung: Dr. Olaf Schnabel, Institut für Kartografie

Zürich, Februar 2007

I. Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei Herrn Prof. Dr. Lorenz Hurni für die Ermöglichung dieser Semesterarbeit bedanken.

Ein ganz besonderer Dank gilt meinem Betreuer Olaf Schnabel, an den ich mich jederzeit mit Fragen wenden konnte und der mir immer mit Rat und Tat zur Seite stand. Durch seine Kompetenz und seinen Ideenreichtum durfte ich eine sehr lehrreiche Zeit erleben.

II. Zusammenfassung

Diese Semesterarbeit befasst sich mit der Entwicklung einer interaktiven Webkarte. Diese sollte bei Änderungen rasch und ohne grossen Aufwand aktualisiert werden können. Um dieses Ziel zu erreichen, wurden Sach- und Geometriedaten separiert. Die Sachdaten wurden in einer Datenbank abgelegt und die Geometrie als SVG-Grafik dargestellt.

Die Arbeit gliedert sich in die zwei Teile Datenaufbereitung und Umsetzung der Webkarte. Da für dieses Gebiet noch keine geeigneten Datensätze vorhanden waren, musste in die Datenaufarbeitung viel Arbeit investiert werden. Die digitalisierten Kartengrundlagen wurden in SVG konvertiert und in eine grafische Benutzeroberfläche eingebettet. Mit Hilfe von JavaScripts wurde die Karte mit Interaktionen und Zusatzinformationen aufgewertet. Dem Nutzer werden verschiedene Werkzeuge geboten, um sich in der Karte zu bewegen und bestimmte Informationen ein- und auszublenden.

Die Kommunikation zwischen der SVG-Datei und der Datenbank wurde über PHP-Scripts realisiert. Diese Scripts ermöglichen die Abfrage der Datenbanken und die Lieferung der Abfrageresultate an die SVG-Karte. Dadurch müssen nicht alle Daten in die SVG-Datei eingebettet werden, sondern können extern in einer Datenbank gespeichert werden.

Auf Grund dieses Aufbaus der interaktiven Webkarte ist eine Aktualisierung der Webkarte ohne grossen Aufwand möglich.

III. Inhaltsverzeichnis

1	EINLEITUNG.....	1
1.1	Ausgangslage.....	1
1.2	Situation.....	1
1.2.1	Bisherige Karten	1
1.2.2	Prototyp für den Campus Hönggerberg	3
1.3	Zielsetzung	4
2	KONZEPTION EINES CAMPUSINFORMATIONSSYSTEMS	5
2.1	Inhalt.....	5
2.2	Layout.....	6
2.3	Farb- und Schriftkonzept.....	7
2.4	Technik	9
3	TECHNISCHE GRUNDLAGEN	10
3.1	ArcGIS.....	10
3.2	SVG	10
3.3	JavaScript	10
3.4	PHP.....	11
3.5	PostgreSQL.....	11
4	UMSETZUNG	12
4.1	Datengrundlagen.....	12
4.2	Datenaufbereitung	12
4.2.1	Digitalisierung im ArcGIS	12
4.2.2	Datenhaltung im PostgreSQL	13
4.3	Konvertierung der GIS-Daten in SVG	14
4.4	Grafische Benutzeroberfläche	15
4.5	Symbolisierung.....	16
4.6	Interaktivität	16

4.6.1	Räumliche Navigation.....	16
4.6.2	Thematische Navigation	17
4.6.3	Onmouseover	18
4.6.4	Tooltip.....	18
4.6.5	Datenbankabfrage mittels PHP	19
4.6.6	Fahrplanauskunft.....	19
5	ANLEITUNG ZUR AKTUALISIERUNG	22
5.1	Änderung von Informationen zu den Gebäuden oder Instituten	22
5.2	Änderung von Gebäudestandorten	22
5.2.1	Aufhebung eines Standorts	22
5.2.2	Neuer Standort	22
5.3	Änderung von Service-Standorten oder -informationen	23
5.4	Änderung der Haltestellen oder der öV-Linien	23
5.5	Erweiterung der Datenbank.....	23
5.5.1	Erweiterung einer Tabelle.....	23
5.5.2	Erweiterung der Datenbank um eine Tabelle.....	24
6	DISKUSSION DES RESULTATS.....	25
6.1	Beurteilung	25
6.2	Erweiterungsmöglichkeiten.....	25
7	SCHLUSSWORT	27
8	LITERATURVERZEICHNIS	28
9	ABBILDUNGS- UND TABELLENVERZEICHNIS	29

1 Einleitung

1.1 Ausgangslage

Die Eidgenössisch Technische Hochschule Zürich (ETH Zürich) ist eine Schweizer Hochschule in Zürich. 18'000 Personen aus 80 Nationen studieren, forschen und arbeiten in unterschiedlichen Instituten und Gebäuden an der ETH Zürich [ETH-Homepage, 2007].

Die ETH Zürich wurde im Jahr 1854 von der Schweizer Eidgenossenschaft gegründet und 1855 in Zürich eröffnet. Die ETH Zürich wuchs rasch an und jedes Jahrzehnt kam ein neues Gebäude dazu. In den sechziger Jahren wurde auf dem Hönggerberg ein neuer Campus aufgebaut [wikipedia, 2007]. Im Laufe der Jahre haben sich die ETH-Gebäude somit über die ganze Stadt verteilt.

Die hier vorliegende Arbeit befasst sich nun mit der Visualisierung von Informationen (Services, Gebäudesuche, Institutssuche), die für die Studierenden und die Beschäftigten der ETH von Interesse sein können. Das Endprodukt soll als interaktive Webkarte ins Internet gestellt werden.

1.2 Situation

1.2.1 Bisherige Karten

Es sind bereits Übersichtspläne der ETH Zürich, Campus Zentrum in analoger und digitaler Form verfügbar. Sie liegen als Freehand-, Illustrator, PDF- und TIFF-Datensatz vor. In Abbildung 1 ist ein solcher Übersichtsplan zu sehen.

Die Karten bieten einen generellen Überblick über den Campus Zentrum. Man sieht, welche Gebäude zur ETH gehören und auch die Gebäudekürzel sind der Karte zu entnehmen.

Viele Vorlesungen des Grundstudiums finden im Zentrum statt. Der Campus im Zentrum umfasst ein sehr grosses Gebiet mit vielen Gebäuden. In den ersten Semestern ist ein Student häufig auf der Suche nach einem bestimmten Gebäude. Mit den bereits existierenden Übersichtsplänen ist zwar eine Suche nach einem Gebäude möglich, doch diese kann unter Umständen sehr mühsam sein, da die ganze Karte selber nach diesem bestimmten Gebäude abgesucht werden muss. Eine Erleichterung einer solchen Suche wäre sicherlich wünschenswert.

Für Besucher oder Gastdozenten der ETH wäre auch eine Institutssuche von Vorteil. Eine Übersicht über die Einrichtungen bzw. Services käme den Studenten zugute. Doch diese beiden Nutzungen sind in den bestehenden Übersichtsplänen nicht integriert.

1.2.2 Prototyp für den Campus Höggerberg

Vor zwei Jahren wurde für den ETH-Campus Höggerberg ein Prototyp für ein Campusinformationssystem ausgearbeitet. Ein Screenshot dieses Campusinformationssystems ist in der Abbildung 2 zu sehen. Dieser Prototyp ist das Resultat der Projektarbeit von Kerstin Schreiner, welche am Institut für Kartografie ihr Praktikum absolviert hat.

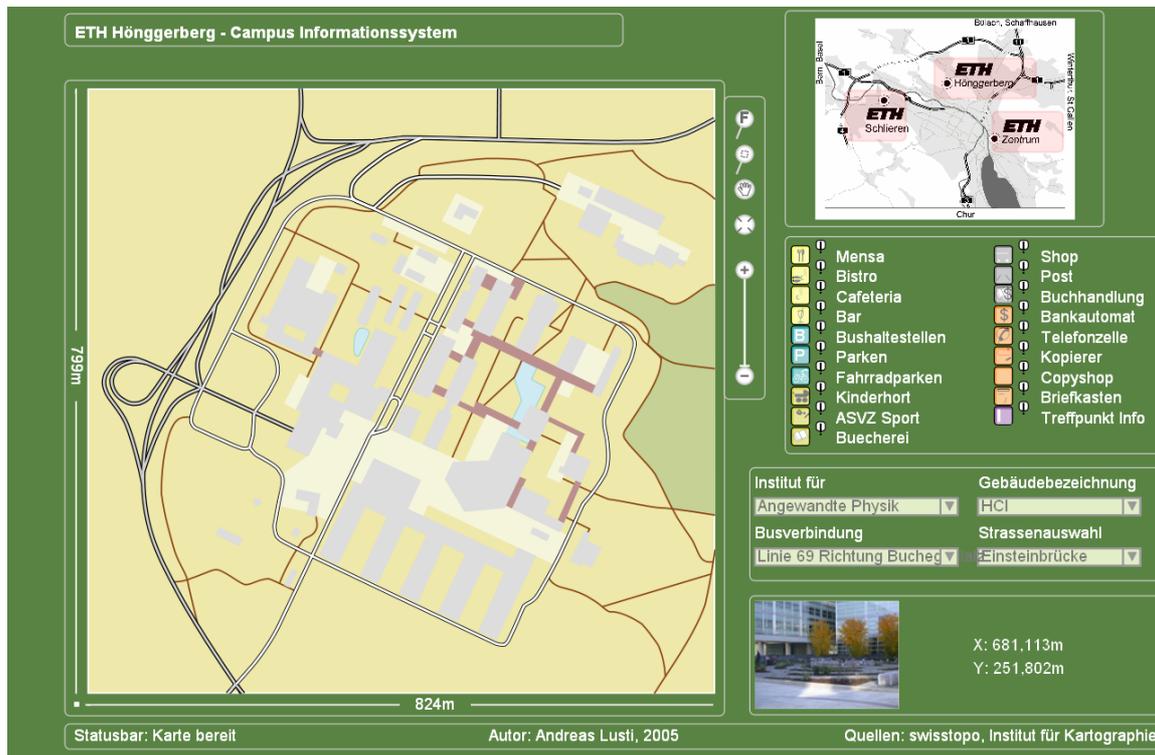


Abb. 2: Campusinformationssystem des ETH-Campus Höggerberg

Ein grosser Vorteil dieser Webkarte ist sicherlich die vereinfachte Gebäudesuche. Es muss nicht mehr die ganze Karte nach einem bestimmten Gebäudekürzel abgesucht werden, sondern das gesuchte Gebäude kann ganz bequem in der Auswahlliste ausgewählt und in der Karte angezeigt werden. Auch die Suche nach einem bestimmten Institut oder nach Einrichtungen bzw. Services wird mit diesem Prototypen ermöglicht.

Das Campusinformationssystem des ETH-Campus Höggerberg weist jedoch noch einige Schwachstellen auf.

- Fährt man mit der Maus über ein Gebäude auf der Karte, so wird dieses Gebäude markiert. Doch diese Information nützt dem Nutzer nichts. Ihm werden keine Informationen zu diesem Gebäude übermittelt, wie beispielsweise das Gebäudekürzel.
- Es ist nicht gut zu erkennen, welche Icons aktiv sind. Die Farbe der Icons in der Legende wird zwar kräftiger, doch der Unterschied zur matten inaktiven Ansicht ist vom Nutzer nicht besonders gut zu erkennen.

- Die Busverbindungen bzw. die Buslinien sind in der Karte nur sehr schlecht zu erkennen. Zudem stimmen sie nicht hundertprozentig mit der Basiskarte überein.
- Der Prototyp kann nicht in allen Browsern betrachtet werden.

1.3 Zielsetzung

Im vorliegenden Projekt soll nun eine interaktive zweidimensionale Karte für den ETH-Campus Zentrum erstellt werden. Damit im Internet ein aktueller Zustand gewährleistet werden kann, muss die Karte möglichst einfach aufgebaut sein. So können Aktualisierungen oder mögliche Erweiterungen ohne grossen Aufwand vorgenommen werden.

Die Karte soll durch Zusatzinformationen ergänzt und mit einfachen Interaktionen aufgewertet werden. Mit Hilfe der intuitiven Interaktionen soll beispielsweise die Gebäudesuche erleichtert und die Informationsvermittlung verbessert werden. Dadurch entstehen mehr Nutzungssituationen.

2 Konzeption eines Campusinformationssystems

2.1 Inhalt

Die Aufgabe besteht darin, eine interaktive Karte für den Campus Zentrum herzustellen, welche auch ins Internet gestellt werden soll. Sie soll die aktuellen Gebäude der ETH darstellen. Zudem sollen auch Informationen über die Gebäude, die Institute und die Einrichtungen angezeigt werden können.

Die Zielgruppe bilden ETH-Studenten und Besucher der ETH, welche ein Gebäude oder Institut suchen. Die Karte soll aber auch einen Überblick über die verschiedenen Dienstleistungen der ETH und deren Standorte verschaffen.

Da die Karte für viele Personen zugänglich ist, muss sie möglichst einfach zu bedienen sein, damit sich auch Anfänger sofort in der Navigation zurechtfinden.

Die Hauptkarte dient der allgemeinen Orientierung und setzt sich aus den folgenden Ebenen zusammen:

- ETH-Gebäude
- öffentliche Gebäude
- sonstige Gebäude
- bebaute Flächen
- Limmat

Die interaktive Webkarte deckt ein sehr grosses Gebiet ab, obwohl der Campus nur ein geringes Ausmass hat. Grund dafür ist die Integration der Gegend um den Hauptbahnhof in die Karte, damit sich auch ausländische Besucher einfach orientieren können. Damit der Nutzer weiss wo er sich befindet, wird zusätzlich eine Referenzkarte eingebaut.

Zur einfachen Handhabung enthält die Webkarte eine Navigationshilfe. Verschiedene Werkzeuge ermöglichen es dem Nutzer, sich in der Karte frei zu bewegen. Anhand des Zoom-Sliders, der Referenzkarte und des Massstabsbalkens ist ersichtlich, welche Zoom-Stufe gewählt ist. Eine Koordinatenanzeige soll auch Nutzern mit GPS-Geräten eine effektive Wegfindung ermöglichen. Zudem wird dem Nutzer ein Übersichtsplan über die verschiedenen ETH-Standorte zur Verfügung gestellt. Damit soll die Grundlage für eine spätere „Verlinkung“ aller ETH-Standorte gelegt werden.

Zur Abfrage von thematischen Informationen enthält die Karte zwei unterschiedliche Abfragewerkzeuge. Mit Hilfe von Checkboxen sollen einzelne Einrichtungen (wie

beispielsweise die Sportanlagen) je nach Interesse des Betrachters in der Hauptkarte ein- bzw. ausgeschaltet werden können.

Zur Gebäude- und Institutssuche werden in der Webkarte Auswahllisten zur Verfügung gestellt. Wird nun beispielsweise ein Institut selektiert, so wird das Gebäude, in welchem sich das gesuchte Institut befindet, in der Karte gekennzeichnet. Zusätzlich werden für den Nutzer interessante Informationen (wie Webseite und Telefonnummer) angezeigt.

Auch beim Überfahren der Gebäude und Icons in der Karte mit der Maus werden die dazugehörigen Informationen angezeigt. Zudem wird das entsprechende Gebäude in der Karte eingefärbt.

2.2 Layout

Die Karte soll in ihrem Grundgerüst aus vier verschiedenen Bereichen bestehen: Einem Kartenfenster für die eigentliche Hauptkarte, einem räumlichen Navigationsfenster, einer Ebenensteuerung für das Ein- und Ausblenden von geokodierten Informationen und einem Informationsfenster.

Das Kartenfenster wird in die linke Bildschirmhälfte platziert und dient der Darstellung der topografischen und thematischen Daten. Grund für die Platzierung ist die Leserichtung des Nutzers, in Europa üblicherweise von links oben nach rechts unten. Aufgrund der grösseren Nord-Süd Ausdehnung des Gebiets wird das Kartenfenster eine rechteckige Form im Hochformat erhalten.

Das Navigationsfenster wird in die rechte obere Ecke neben der Hauptkarte, aber unter den Titel platziert. Dort werden dem Nutzer die Werkzeuge zur Verfügung gestellt, damit er sich frei in der Karte bewegen kann.

Die thematische Navigation erfolgt im Fenster mit der Ebenensteuerung, welches sich direkt unterhalb des Navigationsfensters befindet. Im oberen Teil werden die Checkboxes platziert und im unteren Teil die Auswahllisten.

Gleich darunter befindet sich das Informationsfenster. Die Daten, welche hier dargestellt werden, werden aus der Datenbank abgefragt und nur auf Anfrage (Mausklick, etc.) dargestellt. Dieser Teil der Karte wird somit jeweils neu generiert, während die Basiskarte immer dieselbe bleibt.

Eine Skizze des Layouts ist in Abbildung 3 dargestellt.

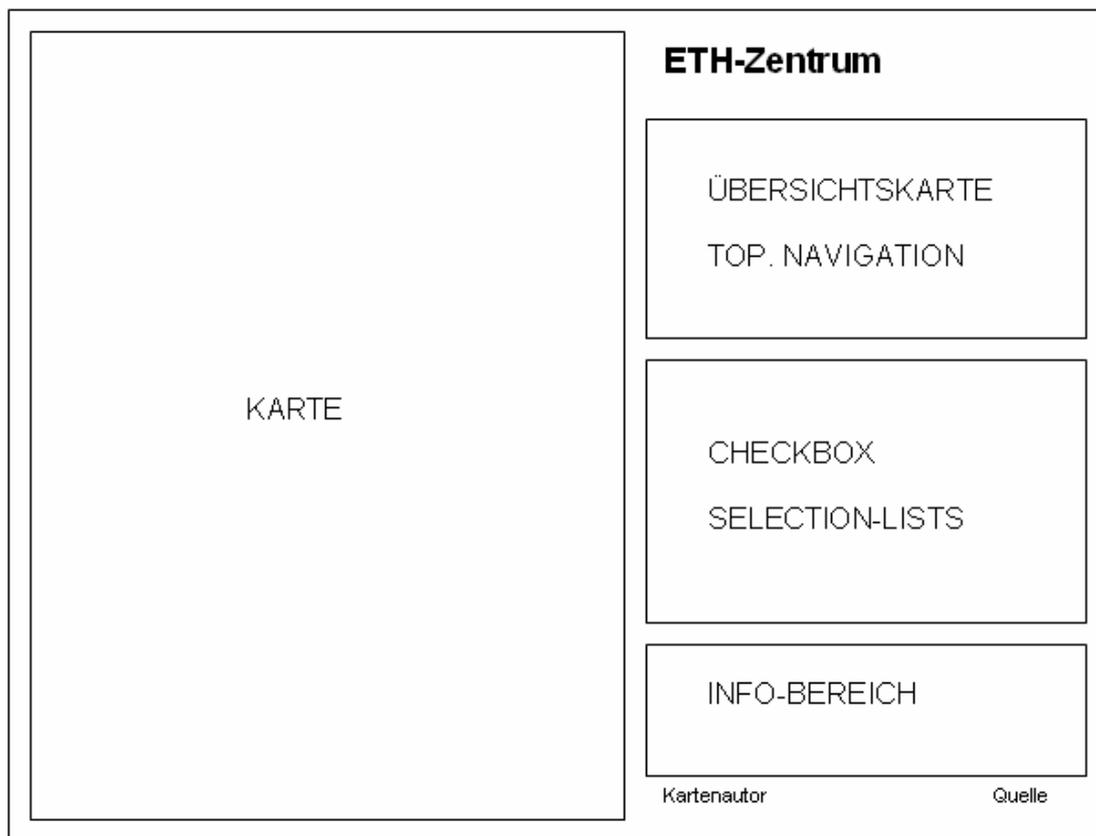


Abb. 3: Layoutentwurf

2.3 Farb- und Schriftkonzept

Die gesamte Webkarte wird in dezenter und aufeinander abgestimmten Farben gehalten. Dies soll dazu beitragen, dass die Aufmerksamkeit des Nutzers auf die jeweiligen Informationen gelenkt wird. Der Hintergrund der gesamten Webkarte soll in einem nicht allzu grellen Orange eingefärbt werden. Damit sich die vier Kartenbereiche gut voneinander unterscheiden lassen, werden sie mit einem Rechteck eingefasst, welches in einem stärkeren Orangeton eingefärbt ist (siehe Tabelle 1).

Die Basiskarte wird, mit Ausnahme der Limmat, in verschiedenen Graustufen dargestellt. Eines der Hauptthemen der Karte, die ETH-Gebäude, wird mit einem Rotton eingefärbt. Die Tram- und Buslinien werden in einem Blauton gehalten und die öffentlichen Gebäude in einem bläulichen Grau (siehe Tabelle 1).

Die Icons werden gruppenweise mit derselben Farbe eingefärbt. Siehe dazu Abbildung 6. Für die Markierung eines ausgewählten Gebäudes in der Karte wird ein intensives Gelb gewählt. Diese soll dem Nutzer in der sonst dezent gehaltenen Basiskarte sofort ins Auge springen und damit soll die Suche nach dem Gebäude erleichtert werden.

Die Schrift soll neutral und gut leserlich sein, aber nicht im Vordergrund stehen. Aus diesem Grund wird ein dunkles Grau und die Schrift Arial, Helvetica gewählt (siehe Tabelle 2).

Karte	Hintergrund		floralwhite	rgb(255, 250, 240)
	Kästchen		oldlace	rgb(253, 245, 230)
Hauptkarte	ETH-Gebäude		darksalmon	rgb(233, 150, 122)
	öffentliche Gebäude		dimgray	rgb(105, 105, 105)
	Umgebungs-Gebäude		darkgray	rgb(169, 169, 169)
	bebaute Fläche		dimgray	rgb(105, 105, 105)
	Limmat		lightskyblue	rgb(135, 206, 250)
	Grünfläche		gainsboro	rgb(220, 220, 220)
Legende	Verkehr		steelblue	rgb(70, 130, 180)
	Essen und Trinken		gold	rgb(255, 215, 0)
	Services der ETH		yellowgreen	rgb(154, 205, 50)
	Einkaufen		sienna	rgb(160, 82, 45)
	Services ausserhalb der ETH		tomato	rgb(255, 99, 71)

Tab. 1: Farbtabelle

	Schriftart	Schriftgrösse	Farbe
Kartentitel	Arial, Helvetica	22	 dimgray rgb(105, 105, 105)
Überschrift der Kästchen	Arial, Helvetica	14	 dimgray rgb(105, 105, 105)
Beschriftung in Kästchen	Arial, Helvetica	10	 dimgray rgb(105, 105, 105)
Text in Auswahlliste	Arial, Helvetica	10	 dimgray rgb(105, 105, 105)
Beschriftung in Karte	Arial, Helvetica	12	 dimgray rgb(105, 105, 105)
Tooltip-Text	Arial, Helvetica	12	 dimgray rgb(105, 105, 105)
Informationstext	Arial, Helvetica	10	 dimgray rgb(105, 105, 105)
Impressums-Überschrift	Arial, Helvetica	18	 dimgray rgb(105, 105, 105)
Impressumstext	Arial, Helvetica	10	 dimgray rgb(105, 105, 105)

Tab. 2: Tabelle der Schriften

2.4 Technik

Die Aufbereitung der Kartengrundlagen erfolgt im ESRI ArcGIS 9.1.

Zur Umsetzung der Webkarte sollen die Programmiersprachen SVG für die grafische Benutzeroberfläche sowie die Karte und Javascript für die Funktionen und Interaktionen zum Einsatz gelangen.

Da die Karte möglichst flexibel und erweiterbar aufgebaut sein soll, ist es sinnvoll, die thematischen Daten in einer PostgreSQL-Datenbank zu speichern. So können die Daten jederzeit bearbeitet unabhängig von der SVG-Karte bearbeitet werden. Eine Änderung in der Karte ist dann nicht nötig.

Um die Karte mit den Daten aus der Datenbank zu beliefern, wird mit PHP-Scripts gearbeitet.

Auf die Benutzung von CSS-Dateien wurde aus Performance-Gründen verzichtet.

3 Technische Grundlagen

3.1 ArcGIS

ArcGIS ist ein weit verbreitetes Geoinformationssystem, welches von ESRI entwickelt wurde. Mit ArcGIS lassen sich Geodaten erzeugen, bearbeiten, analysieren und darstellen [wikipedia, 2007].

3.2 SVG

SVG ist die Abkürzung von Scalable Vector Graphics und ist ein Standard des W3C (World Wide Web Consortium). SVG ist eine Programmiersprache zur Beschreibung von zweidimensionalen Vektorgrafiken.

SVG basiert auf der weit verbreiteten Auszeichnungssprache XML und ist somit von Mensch und Computer lesbar. Die Bearbeitung von SVG kann durch einfache Text- oder XML-Editoren (z.B. Oxygen), aber auch durch speziell dafür entwickelte Programme erfolgen (z.B. Inkscape).

SVG-Grafiken können in den Webbrowsern Mozilla Firefox und Opera von Haus aus dargestellt werden. Der Internet Explorer benötigt zur Darstellung noch einen speziellen SVG-Viewer bzw. Plug-In, welches jedoch kostenlos herunter geladen und installiert werden kann [wikipedia, 2007].

3.3 JavaScript

JavaScript ist eine Skriptsprache, die unter dem Namen ECMAScript durch die ECMA International standardisiert wurde. Da es sich um eine clientseitige Sprache handelt, muss sie lediglich vom Webbrowser interpretiert werden.

JavaScripts bieten die Möglichkeit, dynamische Webseiten zu generieren. Im JavaScript können Funktionen programmiert werden, mit welchen auf die Dokumentenstruktur einer HTML-Seite zugegriffen und deren Gestaltung beeinflusst werden können. Somit werden sie benutzt, um Funktionen in einer SVG-Datei umzusetzen. So kann beispielsweise bei einer Mausbewegung über ein Objekt eine JavaScript-Funktion aufgerufen werden, welche dem Objekt eine andere Farbe zuweist [wikipedia, 2007].

3.4 PHP

PHP (PHP Hypertext Preprocessor) ist eine serverseitige Skriptsprache, die hauptsächlich zur Erstellung von dynamischen Webseiten verwendet wird. Im Gegensatz zur clientseitigen Skriptsprache (z.B. JavaScript), muss der Code nicht an den Webbrowser übermittelt werden, sondern wird direkt vom Webserver interpretiert. Erst dessen Ergebnis wird an den Client geschickt und ist somit für den Benutzer einsehbar.

Ausserdem erlaubt PHP die Abfrage von serverseitigen Datenbanken und Dateien [wikipedia, 2007].

3.5 PostgreSQL

PostgreSQL ist eines der am weitesten fortgeschrittenen objektrelationalen Datenbanksysteme. Es ist eine freie Software und kann einfach installiert werden. Ausserdem bietet sie eine Schnittstelle zur serverseitigen Skriptsprache PHP und ermöglicht somit einen effizienten Zugriff auf die Datenbank [wikipedia, 2007].

4 Umsetzung

4.1 Datengrundlagen

Wie bereits erwähnt, existieren für den ETH-Campus im Zentrum bereits digitale Karten. Da in der Webkarte eine Koordinatenanzeige erwünscht ist, können diese vorhandenen Datensätze nicht übernommen werden. Es ist eine Neudigitalisierung erforderlich.

Eine Kartengrundlage dafür ist der Übersichtsplan der Stadt Zürich im Massstab 1:5'000 aus der Kartensammlung der ETH Zürich. Dieser liegt als Rasterbild im TIFF-Format vor und ist bereits georeferenziert.

Weitere Kartengrundlagen sind die bereits existierenden Übersichtskarten der ETH Zürich, Campus Zentrum sowie die Homepage der ETH Zürich, www.ethz.ch. Aber auch das Handbuch 2005/2006 ist eine gute Datengrundlage. Wichtige Informationen zu den Instituten und den Einrichtungen können daraus entnommen werden.

4.2 Datenaufbereitung

4.2.1 Digitalisierung im ArcGIS

Der Übersichtsplan der Stadt Zürich liegt als Rasterbild im TIFF-Format vor und ist bereits georeferenziert. Somit müssen die einzelnen Bildkacheln nur ins ArcMap geladen und das Koordinatensystem CH1903_LV03 angegeben werden. Die einzelnen Kacheln liegen dann bereits an der richtigen Raumposition, als Koordinaten werden Schweizer Landeskoordinaten angegeben.

Damit in der Webkarte Interaktionen mit den einzelnen Gebäuden der ETH möglich sind, müssen diese ins Vektorformat überführt, also abdigitalisiert werden. Da der Übersichtsplan der Stadt Zürich aufgrund der hohen Informationsdichte und des grossen Speicherplatzbedarfs für den gewünschten Kartenausschnitt nicht als Basiskarte für Internetkarten geeignet ist, muss der gesamte Kartenausschnitt digitalisiert werden.

Anschliessend werden die einzelnen Ebenen mit einer Maske auf den passenden Kartenausschnitt reduziert.

In Tabelle 3 sind alphabetisch geordnet die digitalisierten Ebenen aufgeführt. Diese Informationen reichen für die Basiskarte bereits aus.

Layername	Feature Type	Attribute
bebaute_flaechen_Clip	polygon	-
Buslinien_Clip	polyline	Buslinie
Bus_Haltestellen_Clip	point	<ul style="list-style-type: none"> • Name • x_Koord • y_Koord
Gebaeude	polygon	Name (Gebäudekürzel)
Gruenflaeche_Clip	polygon	-
Limmat_Clip	polygon	-
sonstige_OEV-Haltestellen_Clip	point	<ul style="list-style-type: none"> • Name • x_Koord • y_Koord
sonstige_OEV-Linien_Clip	polyline	-
Tramlinien_Clip	polyline	Tramlinie
Tram_Haltestellen_Clip	point	<ul style="list-style-type: none"> • Name • x_Koord • y_Koord
Umgebungs-Gebaeude_Clip	polygon	-

Tab. 3: Auflistung der digitalisierten Ebenen

4.2.2 Datenhaltung im PostgreSQL

Die Sachdaten sind in einer PostgreSQL-Datenbank abgelegt. Die Datenbank selbst ist auf dem Server www.carto.net angelegt. Mit dem kostenlosen Programm „pgAdmin III“ werden die Tabellen für die Sachdaten angelegt. Beispielsweise wird eine Tabelle „gebäude“ für die Informationen über die Gebäude angelegt.

Viele der benötigten Daten sind bereits auf Datenbanken der ETH gespeichert. Diese Daten sind jedoch nicht zentral in einer Datenbank gespeichert, sondern in vielen verschiedenen, die jeweils auch verschiedenen administrativen Leitern unterstellt sind. Somit ist es zurzeit leider nicht möglich, die bereits bestehenden Datenbanken zu verwenden. Die Daten müssen daher von Hand eingegeben werden. Da die Tabellen der Datenbank wie Excel-Tabellen aussehen und funktionieren, sind Eingaben und Manipulationen ganz intuitiv vorzunehmen.

Die Zusammenhänge der einzelnen Tabellen sind in Abbildung 4 dargestellt.

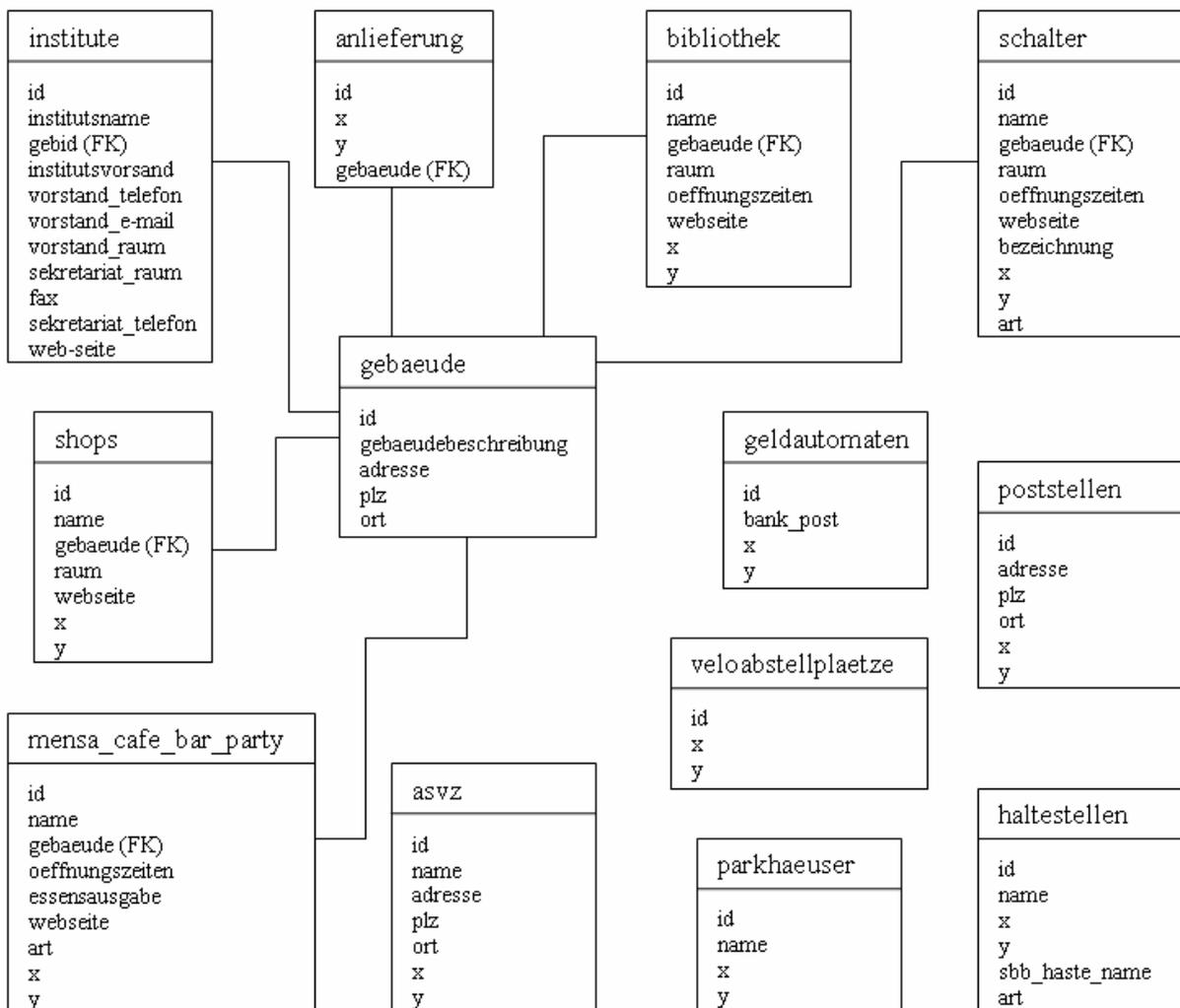


Abb. 4: Schema der Datenbank

4.3 Konvertierung der GIS-Daten in SVG

Für den Export der Geometrien aus ArcMap wird ein Konverter „ogis2svg“ verwendet, der von Andreas Neumann entwickelt worden ist. Mit Hilfe dieses Konverters, der von www.carto.net/papers/svg/utills/shp2svg/ herunter geladen werden kann, werden die GIS-Datensätze in das SVG-Format umgewandelt.

Mit dem Befehl `ogis2svg - input dateiname.shp - output dateiname.svg - roundval 0.1` wird das Shapefile in ein SVG-File transformiert. Dabei bestehen jeweils noch diverse Auswahlmöglichkeiten, um den Objekten die Attribute aus den Shapefiles zuzuordnen.

Die Konvertierung wird für jedes Shapefile separat durchgeführt. Die Konvertierung für das Shapefile „Tramlinie“ ist in Abbildung 5 dargestellt.

```
Z:\GIS_Daten>ogis2svg --input Tramlinien.shp --output tramlinien.svg --roundval
0.1
working on layer tramlinien ...
converting shapefile to a temporary sqlfile ... done.
tablename: tramlinien
The following attributes are available. Please select the attributes you want to
include in the SVG export:
Attribute=gid, Type=serial; Do you want to include it [y|n]?n
Attribute=id, Type=int4; Do you want to include it [y|n]?n
Attribute=tramlinie, Type=varchar; Do you want to include it [y|n]?y
You selected the following Attributes: tramlinie
Which one would you like to select as a unique svg-id?
Type in attribute Name or 'none' if you don't want to include a unique id:
none
you selected "none" as a unique attribute ...
Do you want to group the data according to one attribute? <type 'y' or 'n'>
Would you like to include event-handlers to the individual elements? Type <y|n>
n
Geometry type = MULTILINESTRING
Converted 10 MULTILINESTRING objects - Done.
xmin: 683010.6, xmax: 684813, ymin: 246740.8, ymax: 249229.4
writing SVG file ...
removing temporary sqlfile ...done!
done?
```

Abb. 5: Konvertierung des Shapefiles „Tramlinien“ in eine SVG-Datei

4.4 Grafische Benutzeroberfläche

Die SVG-Dateien werden in eine bereits bestehende grafische Benutzeroberfläche eingebunden.

Als Grundlage für die grafische Benutzeroberfläche dient das Grundgerüst von Andreas Neumann, welches unter www.carto.net/papers/svg/navigationTools/ bezogen werden kann. Dieses bietet bereits die gängigen Funktionen für die räumliche Navigation und die Ebenensteuerung.

Die konvertierten Shapefiles aus ArcMap werden als Gruppen in die SVG-Datei eingebunden. Dort werden ihnen über die Style-Eigenschaften die entsprechenden Farben und Linienstärken zugewiesen. Der Gruppe „gebäude“ werden zusätzlich noch die Funktionen onmouseover und onmouseout angehängt. Die einzelnen Objekte sind als Pfade, mit einem Attributnamen, abgespeichert.

Das Grundgerüst muss anschliessend an den Layoutentwurf angepasst werden. Die Anordnungen der Überschrift, des Impressums und der Navigationselemente werden verändert und um weitere Elemente ergänzt.

4.5 Symbolisierung

Ein Teil der thematischen Navigation erfolgt durch Icons. Diese sollten auch ohne Beschriftung sofort vom Nutzer erkannt und interpretiert werden können.

Abbildung 6 gibt einen Überblick über die Themenbereiche und die einzelnen Icons.

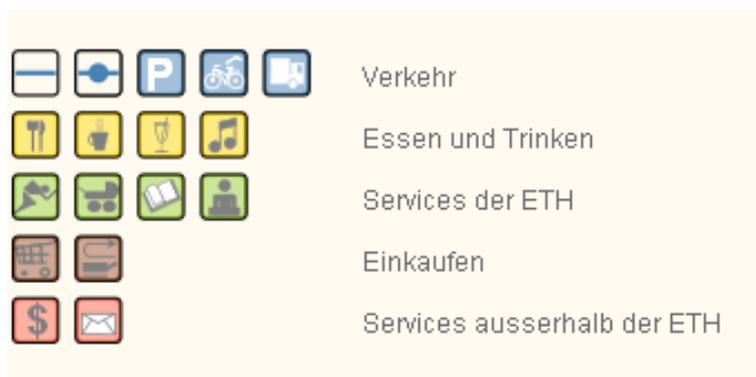


Abb. 6: Symbolisierung

Ein Grossteil der Icons wird vom Prototyp von Kerstin Schreiner übernommen und farblich an das Layout und den Themenbereich angepasst. Ausserdem wurden unnötige CSS-Referenzen und SVG-Geometrien beseitigt. Die anderen werden im Programm Illustrator erstellt und als SVG-Datei abgespeichert. Somit können sie ohne Probleme in die SVG-Datei übernommen werden.

4.6 Interaktivität

4.6.1 Räumliche Navigation

Das Fenster mit der räumlichen Navigation besteht aus einer Referenzkarte, einem Übersichtsplan über die ETH-Standorte, einem Massstabsbalken, einer Koordinatenanzeige und den Navigationswerkzeugen. Diese Werkzeuge bieten die grundlegenden Funktionen wie Zoom, Pan, Zoom-Rechteck aufziehen, Schritt zurück, Schritt vorwärts, Kartenmittelpunkt definieren, Gesamtansicht und Informationsabfrage. Mit Hilfe der Informationsabfrage kann nach dem Navigieren wieder in den Informationsmodus zurückgewechselt werden.

Das Kartenbild soll mit Hilfe des adaptiven Zoomens vor Überlastung bewahrt bleiben. So werden die Strichstärke der Tram- und Buslinien, die Grösse der Symbole und die Grösse der Gebäudebeschriftung an den jeweiligen Massstab angepasst.

4.6.2 Thematische Navigation

Im Fenster mit der Ebenensteuerung befinden sich einerseits Icons (Checkboxen), mit welchen der Nutzer Informationen ein- bzw. ausschalten kann. Andererseits sind dort auch die Auswahllisten platziert, mit welchen einzelne Gebäude bzw. Institute gesucht und angezeigt werden können.

In Abbildung 7 ist das Fenster mit der Ebenensteuerung abgebildet.



Abb. 7: Fenster mit der Ebenensteuerung

Checkboxes

Die Beschreibung und Funktionalität der Checkboxes sind bereits im Grundgerüst der grafischen Benutzeroberfläche vorhanden.

Diese Checkboxes rufen die Funktion „loadData“ auf, welche die Daten für das entsprechende Icon aus der Datenbank abrufen und in ein Array speichert. Die Datenbank wird jedoch nur beim ersten Mal kontaktiert, also wenn das Array noch leer ist.

Anschliessend wird die Funktion „toggleMapLayer“ aufgerufen. Diese Funktion verändert den jeweiligen Sichtbarkeitsstatus „visibleStatus“ auf „sichtbar“ bzw. „unsichtbar“ (visible oder hidden) und den Schriftstil des Legendentextes „labelText“ (font-weight) auf „fett“ (bold) bzw. normal.

Wird die Checkbox beispielsweise auf visible gesetzt, ist sie aktiv. Dann werden in der Karte die entsprechenden Icons angezeigt. Damit der Nutzer auch sieht, welche Checkboxes aktiv sind, werden die Beschriftungen in der Legende fett geschrieben.

Wird die Checkbox jedoch auf hidden gesetzt, werden die Icons in der Karte ausgeblendet, die Schrift wieder zurückgesetzt und das Symbol matt dargestellt.

Auswahllisten

Wie in Abbildung 7 zu sehen ist, werden zwei Auswahllisten eingefügt. Aus diesen Auswahllisten können Gebäudekürzel bzw. Institute ausgewählt werden.

Wird beispielsweise ein Gebäudekürzel ausgewählt, wird dieses Gebäude in der Karte gelb markiert und zusätzlich mit einem Fadenkreuz hervorgehoben. Die Informationen zu diesem ausgewählten Gebäude werden im Informationsfenster angegeben.

Alle Informationen für die beiden Auswahllisten werden bereits beim Laden der Webkarte automatisch aus der Datenbank abgerufen. Wird nun ein gestimmtes Gebäudekürzel selektiert, müssen nur noch die entsprechenden Informationen im Informationsfenster angezeigt werden und eine Datenbank-Abfrage ist nicht mehr nötig.

4.6.3 Onmouseover

Beim Überfahren eines Gebäudes mit der Maus sollen im Informationsfenster dazugehörige Informationen angezeigt werden. Zudem soll das gewünschte Gebäude gelb eingefärbt und beim Verlassen wieder der ursprüngliche Zustand hergestellt werden. Dazu wird in der SVG-Datei die Gruppe „gebaeude“ mit einem „onmouseover“- und einem „onmouseout“-Event versehen.

Beim Überfahren eines Gebäudes wird die Funktion „showGebName“ aufgerufen und zusätzlich ein Parameter übergeben. Die Funktion löscht alle bisher angezeigten Informationen, färbt das ausgewählte Gebäude gelb ein und gibt die Gebäudeinformationen, welche sie aus einem Array ausliest, im Informationsfenster aus.

Wenn die Maus das Gebäude wieder verlässt, wird die Funktion „hideGebName“ ausgeführt. Diese setzt den ursprünglichen Zustand wieder her und löscht die Gebäudeinformationen im Informationsfenster.

4.6.4 Tooltip

Beim Überfahren eines Icons auf der Karte werden in einem kleinen Rechteck direkt neben dem Icon die zusätzlichen Informationen zu diesem Objekt angegeben. Diese Interaktion wird mit der Funktion „tooltipIni“ aufgerufen.

In der Funktion „generateUseElements“ werden der Variablen „tooltipText“ Werte, welche aus einem Array ausgelesen werden, hinzugefügt. Diese Informationen werden in einem eigens dafür generierten Rechteck angezeigt.

Beim Verlassen des Icons werden mit der Funktion „deleteChildren“ diese erzeugten Informationen wieder gelöscht.

4.6.5 Datenbankabfrage mittels PHP

Zur Kommunikation zwischen der SVG-Datei und der Datenbank wird PHP verwendet. Der Ablauf einer solchen Datenbankabfrage ist in Abbildung 8 illustriert.

Beim Klick auf ein Icon wird mit JavaScript die Funktion „loadData“ aufgerufen. Diese wiederum ruft mit der Funktion „getXMLData“ die PHP-Datei auf und übergibt die Parameter „Tabellenname“ und „Spalten“. Anhand dieser Angaben stellt die PHP-Datei die Verbindung mit der Datenbank her und sendet die SQL-Abfrage an die Datenbank. Das Resultat der Abfrage wird mit Hilfe von PHP-Code in ein XML-Dokument geschrieben. Dieses Dokument wird an die ebenfalls mit dem „getXMLData“-Aufruf angegebene Funktion „callbackPHPRequest“ übergeben. Das Resultat wird anschliessend in der callbackPHPRequest-Funktion in einem Array abgespeichert und kann für alle weiteren Interaktionen verwendet werden, ohne dass noch mal eine Abfrage an die Datenbank erfolgen muss.

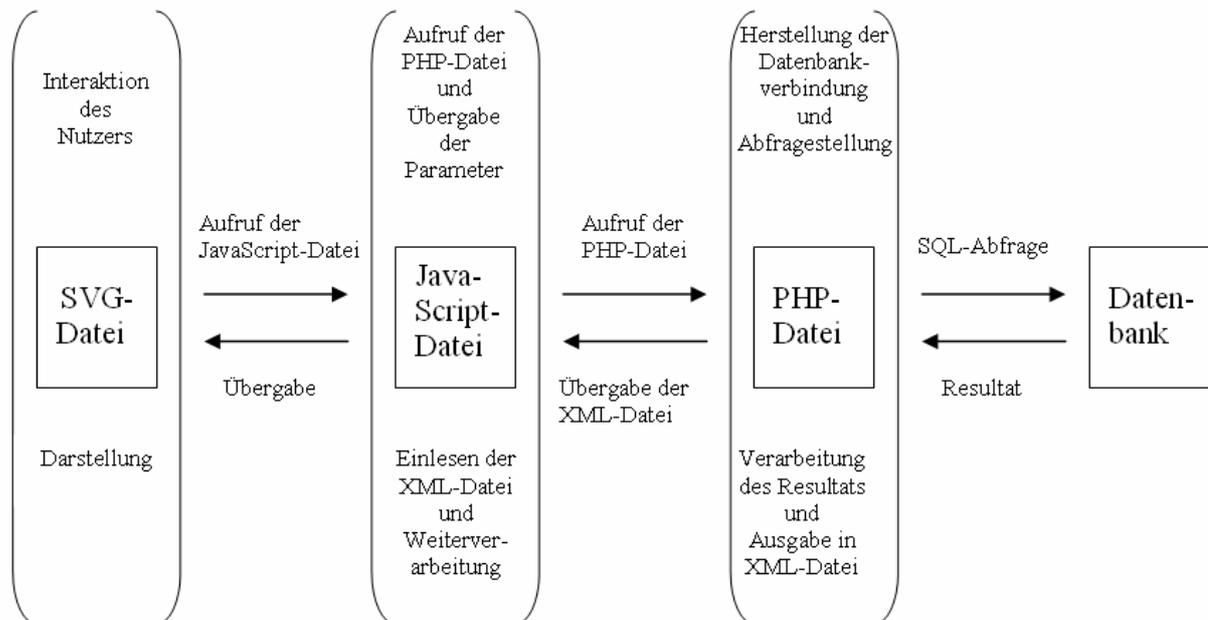


Abb. 8: Ablauf einer Datenbankabfrage

4.6.6 Fahrplanauskunft

Die Webkarte wird mit einer Fahrplanauskunft aufgewertet. Die zugrunde liegende Idee stammt aus der Webseite www.map.search.ch. Überfährt der Nutzer mit der Maus ein Haltestellen-Icon, so wird ihm nicht nur der Name dieser Haltestelle angezeigt, sondern auch die nächsten Tram- und Busverbindungen.

Auch diese Abfrage wird mittels einer PHP-Datei ermöglicht. Beim Überfahren des Icons wird mit JavaScript die onmouseover-Funktion „loadAbfahrtszeiten“ aufgerufen. Diese

Funktion ruft die PHP-Datei „fahrplan“ auf und übergibt ihr die id der betreffenden Haltestelle.

Für die Abfrage der Abfahrtszeiten auf der SBB-Homepage, benötigt man die folgenden Parameter:

- aktuelle Zeit („curr_time“):
Diese kann mit dem PHP-Befehl `strftime(„%H:%M“)` vom Computer automatisch generiert werden.
- Haltestellenname, wie er in der URL verwendet wird („sbb_hast_name“):
Dieser Name ist in der Datenbank abgespeichert und muss nur von dort abgefragt werden.

Die PHP-Datei stellt somit als erstes die Verbindung mit der Datenbank her und fragt in der Tabelle „haltestellen“ die Spalte „sbb_hast_name“ ab. Anschliessend werden die oben erwähnten Parameter in die folgende URL eingefügt und die Abfrage gestartet:

```
http://fahrplan.sbb.ch/bin/bhftafel.exe/dn?input='.$row["sbb_hast_name"].'&selectDate=to  
day&time='.$curr_time.'+&boardType=dep&REQProduct_list=1%3A1111111111000000  
&maxJourneys=10&start=%BB%A0Anzeigen&distance=1'
```

Die nun abgefragte Webseite wird mittels PHP als Text in eine Variable geschrieben. Aus diesem Text werden die Tram- bzw. Busnummern, die Abfahrtszeiten (Abbildung 9) und die Endhaltestellen mittels weiterer PHP-Textoperationen herausgelesen, bearbeitet und in ein XML-Dokument geschrieben.

```
29 $code1 = '<span class="bold">';  
30 $code2 = '&station_type=dep">';  
31 $code3 = '<td class="result">';  
32  
33 $handle = fopen($requeststring, "rb");  
34 $contents = stream_get_contents($handle);  
35 $times=explode($code1,$contents);  
36 $i = 0;  
37 for ($a=0;$a<count($times);$a++)  
38 {  
39     $outputTimes=substr($times[$a],0,5);  
40     if ($outputTimes{2}=="")  
41     {  
42  
43         $fahrplan[$i]['time'] = $outputTimes;  
44         $i++;  
45     }  
46 }  
--
```

Abb. 9: PHP-Code für das Herauslesen der Abfahrtszeiten

Das erzeugte XML-Dokument wird in der SVG-Datei durch die Funktion „callbackFahrplan“ gelesen und in ein JavaScript-Array abgespeichert. Anschliessend werden die Informationen in ein Tooltip integriert, welches in der Webkarte sowohl den Haltestellenamen wie auch eine Auflistung der nächsten Tram- und Busverbindungen anzeigt.

5 Anleitung zur Aktualisierung

5.1 Änderung von Informationen zu den Gebäuden oder Instituten

Bei Änderungen von Informationen zu den Gebäuden oder Instituten, müssen nur in der Datenbank Änderungen vorgenommen werden.

Ändert beispielsweise der Hauptsitz eines Instituts, dann muss nur in der Spalte „gebid“ der Tabelle „institute“ das neue Gebäudekürzel hineingeschrieben werden. Wird anschliessend in der Webkarte dieses Institut aufgerufen, wird, nachdem die Karte neu geladen ist, automatisch das richtige Gebäude in der Karte angezeigt. An der Auswahlliste muss nichts verändert werden.

5.2 Änderung von Gebäudestandorten

Bei einer allfälligen Aktualisierung eines Gebäudestandortes, müssen in der SVG-Datei die Pfade in die entsprechende Gruppe verschoben werden und in der Datenbank ein paar wenige Anpassungen vorgenommen werden.

5.2.1 Aufhebung eines Standorts

Wird beispielsweise ein Standort aufgehoben, muss für die Hauptkarte in der Gruppe „gebaeude“ der entsprechende Pfad ausgeschnitten und in die Gruppe „umgebungs_gebaeude“ verschoben werden.

Für die Referenzkarte muss nur in der Gruppe „ref_gebaeude“ der Pfad gelöscht werden. An den anderen Gruppen muss nichts verändert werden, da diese mit der Hauptkarte verlinkt sind und sich somit automatisch anpassen.

Zusätzlich muss auch die Datenbank angepasst werden. So muss in der Datenbank in der Tabelle „gebaeude“ das entsprechende Gebäude aus der Tabelle gelöscht werden.

5.2.2 Neuer Standort

Für den Fall, dass ein neues Gebäude von der ETH gemietet wird, muss dieses Gebäude von der Gruppe „umgebungs_gebaeude“ in die Gruppe „gebaeude“ verschoben und zusätzlich in die Gruppe „ref_gebaeude“ kopiert werden. Anschliessend muss dem Pfad das Attribut „attrib:name“ und das (von der ETH vergebene) Gebäudekürzel hinzugefügt werden.

In der Datenbank muss dieses neue Gebäude in der Tabelle „gebaeude“ erfasst werden. Dabei muss darauf geachtet werden, dass unter „id“ dasselbe Gebäudekürzel angegeben wird wie bereits zuvor in der SVG-Datei unter „attrib:name“.

5.3 Änderung von Service-Standorten oder -informationen

Bei Änderungen von Informationen zu den Service-Standorten oder den Serviceinformationen müssen nur in der Datenbank Änderungen vorgenommen werden.

Sollte beispielsweise eine neue Bibliothek entstehen, dann muss die Tabelle „bibliothek“ um diese erweitert werden. Die Koordinaten des Standorts können mit Hilfe der Koordinatenanzeige in der Webkarte ermittelt und anschliessend in die Datenbank geschrieben werden.

Ändern sich hingegen beispielsweise die Öffnungszeiten einer Bibliothek, so müssen diese in der Tabelle „bibliothek“ in der Spalte „oeffnungszeiten“ angepasst werden.

5.4 Änderung der Haltestellen oder der öV-Linien

Für den unwahrscheinlichen Fall, dass sich Änderungen im öffentlichen Verkehr ergeben, müssen im ArcGIS die Geometrien des entsprechenden Shape-Files bearbeitet bzw. neu erstellt werden. Anschliessend müssen diese veränderten Geometrien ins SVG-Format exportiert werden. Die Einstellungen für eine solche Konvertierung mit „ogis2svg“ sind der Abbildung 5 zu entnehmen.

In der SVG-Datei müssen die Pfade der vorhandenen Gruppe gelöscht und durch die Pfade in der neu erstellten SVG-Datei ersetzt werden.

5.5 Erweiterung der Datenbank

5.5.1 Erweiterung einer Tabelle

Sollen zusätzliche Informationen in der Webkarte angegeben werden, die noch nicht in der Datenbank erfasst sind, so müssen die existierenden Tabellen erweitert werden.

In der Datenbank müssen die entsprechenden Tabellen mit neuen Spalten erweitert und anschliessend gefüllt werden. Anschliessend muss die Abfrage in der PHP-Datei um diese Spalten erweitert werden. In der SVG-Datei sind die entsprechenden Anpassungen in den Funktionen „callbackPHPRequest“ und „generateUseElements“ vorzunehmen.

5.5.2 Erweiterung der Datenbank um eine Tabelle

Für den Fall, dass die Webkarte durch die Anzeige eines weiteren Icons erweitert werden soll, müssen viele Änderungen vorgenommen werden. Einerseits muss in der Datenbank eine neue Tabelle erstellt werden. In der SVG-Datei muss eine leere Gruppe angelegt und Skripte geschrieben werden. Und selbstverständlich darf auch die Erstellung eines neuen Symbols nicht vergessen werden.

6 Diskussion des Resultats

6.1 Beurteilung

Die interaktive Webkarte bietet verschiedene Funktionen an. Es wurden die wichtigsten Interaktionsmöglichkeiten umgesetzt und funktionstüchtig gemacht.

Neben der üblichen räumlichen Navigation solcher Webkarten, erlaubt diese auch das Ein- und Ausschalten von Icons, sowie die Navigation mittels Auswahllisten. Zudem wurde die Karte durch Zusatzinformationen aufgewertet.

Ein grosser Vorteil gegenüber der vorhandenen statischen Karte ist sicherlich die vereinfachte Suchmöglichkeit nach Gebäuden mittels einer Auswahlliste. Aber auch die Zusatzinformationen zu den Gebäuden, den Instituten und den sonstigen Einrichtungen werten die Karte auf.

Aufgrund der zeitlichen Beschränkung und des grossen Zeitaufwandes für die Datenaufbereitung konnten leider nicht alle Funktionen implementiert werden. So musste beispielsweise auf die Angabe der Gebäudeöffnungszeiten und eine Druckfunktion verzichtet werden. Das Drucken ist momentan jedoch durch die Druckmöglichkeit im Browser möglich. Wünschenswert wäre jedoch ein Ausdruck mit integrierter Legende und einem Massstabsbalken.

Auch das Ziel einer möglichst einfachen Aktualisierbarkeit der Webkarte wurde dank der Anbindung einer Datenbank erreicht. Es müssen nur in Ausnahmefällen Shape-Files ins SVG exportiert werden. Aber auch dieser Export ist mit Hilfe des Konvertierungstools „ogis2svg“ in wenigen Schritten vollbracht. Alle anderen Änderungen können in der Datenbank vorgenommen werden.

6.2 Erweiterungsmöglichkeiten

Das Endresultat ist zwar sehr zufrieden stellend, doch die Webkarte weist noch einige Verbesserungs- bzw. Erweiterungsmöglichkeiten auf:

- Wie bereits erwähnt, wäre es wünschenswert, wenn noch eine Druck-Funktion eingebaut werden würde. Auf diese Weise könnte der Ausdruck noch mit einer Legende und mit einem Massstabsbalken ergänzt werden.
- Eine Such-Funktion könnte die Suche nach einem bestimmten Gebäude noch mehr erleichtern. Momentan muss das gewünschte Gebäude in der relativ langen

Auswahlliste gesucht werden. Ein Such-Fenster, wo das Gebäude eingegeben und anschliessend gesucht werden kann, könnte die Karte aufwerten. Doch auch bereits das Scrollen mit dem Rad an der Maus könnte die Suche erleichtern.

- Ist der Nutzer auf der Suche nach einem bestimmten Schalter, muss er mit der Maus über jedes einzelne Icon in der Karte fahren bis er den gesuchten Schalter findet. Auch diese Suche könnte dem Nutzer vereinfacht werden. Beispielsweise könnte in der Legende ein aufklappbares Fenster implementiert werden, wo die einzelnen Schalter auswählbar sind.
- Da die ETH auch von vielen fremdsprachigen Leuten besucht wird, wäre es sicherlich sinnvoll, wenn auf der Webseite eine Umschaltung der Sprache möglich wäre. Hierfür müsste die Datenbank an eine solche Sprachumschaltung angepasst werden und in der SVG-Datei müsste eine solche Funktion integriert werden. Das ist im Prinzip recht einfach möglich.
- Weiter sollte für die Freistellung der Icons gesorgt werden. Im Bereich des Hauptgebäudes sind viele Einrichtungen platziert und somit überschneiden sich die Icons, was die Lesbarkeit der Karte verschlechtert.
- Auch eine Erweiterung des schon integrierten adaptiven Zoomens wäre wünschenswert. So könnten ab einer gewissen Zoom-Stufe die Strassen beschriftet werden. Dies würde sicherlich der Orientierung dienen.

7 Schlusswort

Für mich war die bearbeitete Aufgabenstellung eine echte Herausforderung und ich blicke auf eine intensive und lehrreiche Zeit zurück. Dieses Projekt ermöglichte es mir, mich mit den eingesetzten Techniken (SVG, JavaScript, PHP und Datenbank) etwas vertraut zu machen.

Diese Arbeit wurde innerhalb von 14 Wochen bearbeitet. Durchschnittlich waren pro Woche 12 Stunden vorgesehen. Diese zeitliche Beschränkung setzt klare Grenzen. Die Aufgabenstellung wurde zwar erfüllt und die SVG-Applikationen funktionieren auch, doch die Webseite konnte nicht bis zur Perfektion ausgearbeitet werden. Es gibt noch einige kleinere Probleme und Verbesserungsmöglichkeiten. Doch alles in Allem ist das Endprodukt äusserst zufrieden stellend.

8 Literaturverzeichnis

<http://www.ethz.ch>

<http://www.sbb.ch>

<http://www.carto.net>

<http://de.wikipedia.org/wiki/ArcGIS>

<http://de.wikipedia.org/wiki/SVG>

<http://de.wikipedia.org/wiki/JavaScript>

<http://de.wikipedia.org/wiki/PHP>

<http://de.wikipedia.org/wiki/PostgreSQL>

<http://www.pgadmin.org>

<http://ch2.php.net>

<http://de.selfhtml.org>

9 Abbildungs- und Tabellenverzeichnis

Abbildungsverzeichnis

Abb. 1:	Übersichtsplan der ETH Zürich, Campus Zentrum.....	2
Abb. 2:	Campusinformationssystem des ETH-Campus Hönggerberg.....	3
Abb. 3:	Layoutentwurf	7
Abb. 4:	Schema der Datenbank	14
Abb. 5:	Konvertierung des Shapefiles „Tramlinien“ in eine SVG-Datei.....	15
Abb. 6:	Symbolisierung	16
Abb. 7:	Fenster mit der Ebenensteuerung	17
Abb. 8:	Ablauf einer Datenbankabfrage.....	19
Abb. 9:	PHP-Code für das Herauslesen der Abfahrzeiten.....	20

Tabellenverzeichnis

Tab. 1:	Farbtabelle	8
Tab. 2:	Tabelle der Schriften.....	8
Tab. 3:	Auflistung der digitalisierten Ebenen.....	13