

Diplomarbeit in Kartographie  
Institut für Kartographie  
Departement Bau, Umwelt und Geomatik

ETH Zürich

---

# Visualisierung der Herkunft der ETH-Studierenden in einer SVG-Karte

## Entwicklung eines Prototyps

---

### **Autorin:**

Regula Stopper  
Studentin der Geomatikingenieurwissenschaften an der ETH Zürich  
e-mail: regula@stopper.ch

### **Betreuung:**

Prof. Dr. Lorenz Hurni, Institut für Kartographie, ETH Zürich  
Olaf Schnabel, Institut für Kartographie, ETH Zürich  
Dr. Urs Hugentobler, Abteilung Controlling, ETH Zürich  
Michael Fuchs, Abteilung Controlling, ETH Zürich

Diplomarbeit zur Erlangung des Titels „Dipl. Geomatik-Ing. ETH“, erstellt zwischen Januar und April  
2005 am Institut für Kartographie (IKA), Departement Bau, Umwelt und Geomatik (D-BAUG)  
Eidgenössische Technische Hochschule Zürich (ETHZ)



## Vorwort

Diese Diplomarbeit ist die Abschlussarbeit meines Studiums der Geomatikingenieurwissenschaften an der ETH Zürich. Sie wurde am Institut für Kartographie der ETH Zürich unter der Leitung von Prof. Dr. L. Hurni angeboten. Mein Interesse an der Kartographie wurde bereits während des Studiums durch die Lehrveranstaltungen des Instituts geweckt. Mit der Absolvierung eines Praktikums im Bereich Multimedialkartographie wurden zusätzlich meine Kenntnisse im Bereich der Internetkartographie vertieft und mein Entscheid für eine Diplomarbeit in diesem Bereich gefällt.

Das Internet ist für viele Menschen ein Bestandteil des normalen Lebens geworden. Immer mehr Wirtschafts- oder Wissensbereiche setzen ihre Zukunft auf das Internet. Umso spannender ist das Erlebnis, ein eigenes Internetprojekt auf die Beine zu stellen. Bis anhin habe ich das Internet fast ausschliesslich aus dem Blickwinkel des Anwenders erlebt. Diese Diplomarbeit ermöglichte mir einen Blick in die Welt der Entwicklung von Internetprojekten.

Diese Studienarbeit richtet sich an die Leserschaft, die sich mit dem Thema „Erstellen von interaktiven SVG-Karten“ befassen. Obwohl ich versuchte, möglichst viele Begriffe in einem Glossar oder in erklärenden Texten zu definieren, verwendet diese Arbeit einige Fachausdrücke, die gewisse Grundkenntnisse in diesem Themenbereich erfordern.

Meinen besonderen Dank möchte ich meinem Betreuer Olaf Schnabel, Doktorand am Institut für Kartographie der ETH Zürich, aussprechen. Seine Hilfe und Anregungen waren wertvoll und trugen zum Gelingen meiner Arbeit bei. Ebenfalls ein grosses Dankeschön geht an Andreas Neumann und Tobias Dahinden, Institut für Kartographie der ETH Zürich. Auch sie haben wesentlich zum Gelingen meiner Diplomarbeit beigetragen. Andreas Neumann bin ich sehr dankbar für das Bereitstellen einiger seiner Skripte und seine Hilfe und auch Tobias Dahinden möchte ich für seine Hilfe und seine guten Ratschläge danken.

Grosser Dank gilt auch Herrn Prof. Dr. Lorenz Hurni, der mir diese Diplomarbeit überhaupt ermöglichte.

Mein persönlicher Dank geht an meine Eltern, die mir dieses Studium ermöglicht und mich immer unterstützt haben. Besonders meiner Mutter Marlies Stopper gilt ein grosses Dankeschön. Sie hat diese Arbeit mit grösster Sorgfalt und Geduld gegengelesen.

Regula Stopper  
Zürich, 26. April 2005

## Zusammenfassung

Die ETH Zürich ist die einzige deutschsprachige Technische Hochschule der Schweiz. Sie verzeichnet im Jahre 2003 ungefähr 12'500 Studierende, wobei ein beachtlicher Teil der Studierenden aus dem Ausland kommt. Die Herkunft der Studierenden wird seit 1990 in einer Oracle-Datenbank erfasst. Die vorliegende Arbeit zeigt eine Möglichkeit auf, wie die Herkunft der Studierenden – in der Schweiz und im Ausland – in einer Karte, die im Internet aufrufbar und interaktiv ist, visualisiert werden kann.

Der Schwerpunkt bei der Entwicklung der Karte liegt bei der Trennung der Präsentation von den Daten. Die Sachdaten werden weiterhin in der Oracle-Datenbank gespeichert und verwaltet. Mittels Datenbankabfragen werden die gewünschten Daten für die Darstellung in der Karte extrahiert.

In der Karte kann der Benutzer folgende Einstellungen tätigen: Wahl des Kartenausschnittes, Bestimmung der Herkunft, darzustellendes Jahr, Status der Studierenden, Studienkategorie, Studiengang und Geschlecht. Eine Navigationshilfe, in Form einer kleinen Übersichtskarte, erlaubt das gezielte Ein- und Auszoomen in den Kartenausschnitten.

Das Projekt wird in SVG (Scalable Vector Graphics) visualisiert. Die Interaktionen der Karte erfolgen mit JavaScript und die Datenbankabfragen mit PHP (PHP-Hypertext Preprocessor). Anhand der Einstellungen, die der Benutzer beliebig im Rahmen der sinnvollen Vorgaben bestimmen kann, wird die Datenbankabfrage zusammengestellt und ausgeführt. Die Ergebnisse werden in der PHP- und der JavaScript-Datei so verarbeitet, dass sie schliesslich in der SVG-Karte dargestellt werden können. Die Darstellung erfolgt mittels Choroplethen in einer Schweizer-, einer Europa- und einer Weltkarte.

Die Ergebnisse werden in Relativwerten dargestellt. Es sind dies die Anzahl Studierender pro Bezirk oder Land in Bezug zu dessen Gesamtbevölkerungszahl. Für die Choroplethendarstellung wird mit Wertklassen gearbeitet, was zur Folge hat, dass die Werte innerhalb einer Klasse nicht untereinander vergleichbar sind. Damit alle Werte miteinander verglichen werden können, wird parallel zur Legende der Karte ein Diagramm erstellt, das alle Werte der Grösse nach auflistet und diese so vergleichbar macht.

Das Resultat der Arbeit ist eine funktionale und benutzerfreundliche Karte. Ihre Umsetzung zeigt auf, dass es mit SVG in Verbindung mit Datenbankabfragen und der Skriptsprache JavaScript möglich ist, die Herkunft der Studierenden der ETH in einer interaktiven Karte, die im Internet aufrufbar ist, zu visualisieren.

## **Abstract**

The ETH Zurich is the only german-speaking Technical University in Switzerland and in the year 2003 had nearly 12'500 enrolled students. Most of them were of swiss origin but there were also a significant number of foreign students. The provenance of the students has been stored in an Oracle-Database since 1990. This thesis demonstrates the possibility visualizing the provenance of ETH-students within a chart format and since the provenance data is of interest to users in Switzerland and abroad, the visualization is designed to be available as an interactive web-based tool.

The focal point of the present thesis is the division of presentation and data. Furthermore, the data is stored and managed in an existing Oracle-Database, where the data is extracted using SQL-queries.

The user can choose any selection of the following settings: chart, provenance (nationality or residence before commencing studies), year, status of enrollment, category of the studies, study course and gender. A navigation allows to targeted zooming within the charts. The navigation is represented by a small generalized map.

The project has been visualised using SVG (Scalable Vector Graphics). The interactions are executed by the scripting programming language JavaScript and PHP (PHP-Hypertext Preprocessor) is used for making database-queries. The settings chosen by the user define the database-query, that will be executed. The results are manipulated and prepared in the PHP- and JavaScript-File. Finally they are visualized in the SVG-chart with choropleths in three different charts: Switzerland, Europe and World.

The results are presented with relative values, such as number of students per number of inhabitants of a district (in the swisschart) or of a land (within the chart of Europe or the chart of world). The presentation of the results with choropleths pools values into different classes, which renders them difficult to compare to one another. For this reason a diagram is created, in which all values are arranged according to size, therefore all values can be compared.

The developed chart is functional and user-friendly. The realization shows that SVG in conjunction with database-queries and the scripting programming language JavaScript, is an appropriate instrument for visualizing the provenance of the ETH-students interactively via the internet.

# Verzeichnisse

## Inhaltsverzeichnis

<b>Vorwort</b> .....	<b>I</b>
<b>Zusammenfassung</b> .....	<b>II</b>
<b>Abstract</b> .....	<b>III</b>
<b>Verzeichnisse</b> .....	<b>IV</b>
Inhaltsverzeichnis.....	IV
Abbildungsverzeichnis.....	VI
Tabellenverzeichnis.....	VII
Abkürzungsverzeichnis.....	VIII
<b>1 Einleitung und Problemstellung</b> .....	<b>1</b>
1.1 Ausgangslage.....	1
1.2 Auftrag und Zielsetzung.....	1
1.3 Aufbau der Arbeit.....	3
<b>2 Theoretische Grundlagen</b> .....	<b>4</b>
2.1 Visualisierung statistischer Daten.....	4
2.1.1 Darstellung qualitativer Werte.....	4
2.1.2 Darstellung quantitativer Werte.....	5
2.1.2.1 Stetige Mengendarstellung.....	5
2.1.2.2 Intervalldarstellung.....	6
2.1.3 Darstellungsmethode.....	8
2.1.3.1 Signaturen.....	8
2.1.3.2 Diagramme.....	9
2.2 Webkarten.....	10
2.2.1 Geschichte der Webkartographie.....	10
2.2.2 Arten von Web-Karten.....	11
2.2.3 Vor- und Nachteile.....	12
2.2.4 Darstellung einer Webkarte.....	13
<b>3 Konzept</b> .....	<b>15</b>
3.1 Inhalt der SVG-Karte.....	15
3.2 Layout des Projektes.....	17
3.3 Wahl der Techniken.....	18

3.3.1 Trennung von Präsentation und Daten.....	18
3.3.2 Präsentation.....	19
3.3.3 Client-Server-Kommunikation.....	19
<b>4 Material und Methodik.....</b>	<b>22</b>
4.1 Verwendete Daten.....	22
4.1.1 Daten in Oracle-Datenbank.....	22
4.1.2 Geometriedaten.....	23
4.1.2.1 Schweizerkarte.....	23
4.1.2.2 Weltkarte.....	23
4.1.2.3 Europakarte.....	24
4.2 Verwendete Programme.....	24
4.3 Arbeitsablauf.....	26
<b>5 Umsetzung.....</b>	<b>27</b>
5.1 Datenaufbereitung.....	27
5.1.1 Aufbereitung der Geometriedaten.....	27
5.1.1.1 Generalisierung der Weltkarte.....	28
5.1.1.2 Aggregation der Schweizerkarte.....	29
5.1.1.3 Erweitern der Attributtabelle.....	29
5.1.2 Aufbereitung der Datenbank.....	30
5.2 Konvertieren der Shapefiles in SVG-Dateien.....	31
5.3 Erstellen der Haupt-SVG-Datei.....	33
5.4 Implementierung der Kartenbestandteile.....	34
5.4.1 Geometrie.....	34
5.4.2 Navigation.....	34
5.4.3 Selektionslisten.....	35
5.5 Implementierung der Interaktionen.....	36
5.5.1 Switchen zwischen einzelnen Kartenausschnitten.....	37
5.5.2 Switchen zwischen Wohnort vor Studienbeginn und Nationalität.....	38
5.5.3 Ablauf der Datenbankabfrage.....	40
5.5.3.1 Zusammenstellung der Datenbankabfrage.....	40
5.5.3.2 Übergeben der Variablen an PHP.....	44
5.5.3.3 Erstellung und Ausführung des PHP-Skriptes.....	44
5.5.3.4 Verarbeitung der Ergebnisse in JavaScript.....	49
5.5.4 Funktionen der Karte.....	57
5.5.4.1 Dynamisches Anzeigen von Zusatzinformation.....	57

5.5.4.2 Dynamischer Selektionslisteninhalt der Anzahl Klassen.....	58
5.5.4.3 Color-Picker.....	59
5.6 Resultat des Projektes .....	60
<b>6 Diskussion des Resultats.....</b>	<b>61</b>
6.1 Resultat für die Schweiz.....	61
6.2 Resultat für die Welt und Europa.....	62
6.3 Browserkompatibilität.....	63
6.4 Kompatibilität mit SVG-Viewern und Betriebssystemen.....	64
6.5 Flexibilität des Projektes.....	64
6.5.1 Hinzufügen von weiteren Jahrgängen.....	64
6.5.2 Ändern der Spaltennamen der Datenbank.....	65
6.5.3 Ändern der Codierung der Attribute.....	65
6.5.4 Änderung der Geometrie .....	65
<b>7 Ausblick.....</b>	<b>66</b>
7.1 Optimierungsmöglichkeiten des Projektes.....	66
7.1.1 Visualisierung der Auslandskarten mit Absolutwerten.....	66
7.1.2 Modifizierung des Diagramms.....	67
7.2 Erweiterung des Projektes.....	67
7.2.1 Sprachauswahl.....	68
7.2.2 Animation.....	68
7.2.3 Dynamische Selektionslisteninhalte.....	68
7.2.4 Speichern der Geometrie in der Datenbank.....	69
7.2.5 Adaptives Zooming.....	69
7.3 Entwicklungspotenzial des Projektes.....	69
<b>8 Glossar.....</b>	<b>71</b>
<b>9 Bibliographie.....</b>	<b>74</b>
9.1 Literatur.....	74
9.2 Quellen aus dem WWW.....	75
9.3 Weiterführende Literatur.....	76
<b>Anhang.....</b>	<b>77</b>

## Abbildungsverzeichnis

Abbildung 1: Choroplethen.....	10
Abbildung 2: Kreisdiagramm.....	10
Abbildung 3: Stabdiagramm.....	10
Abbildung 4: Layout der Karte.....	17
Abbildung 5: Client-Server Kommunikation.....	20
Abbildung 6: Ablaufprozess.....	20
Abbildung 7: Navigation.....	34
Abbildung 8: Kartenausschnitt.....	37
Abbildung 9: Herkunft: Nationalität.....	39
Abbildung 10: Herkunft: Wohnort vor Studienbeginn.....	39
Abbildung 11: Farbverlauf.....	53
Abbildung 12: Einfärbung "Quantile".....	54
Abbildung 13: Einfärbung "Gleiche Intervalle".....	54
Abbildung 14: Legende.....	55
Abbildung 15: Diagramm "gleiche Intervalle".....	56
Abbildung 16: Diagramm "Quantile".....	56
Abbildung 17: Diagramm "Vergleich der Geschlechter".....	57
Abbildung 18: Anzeigen des Pfades.....	57
Abbildung 19: Anzeigen von Information in der Legende.....	57
Abbildung 20: Anzeigen von Information im Diagramm.....	58
Abbildung 21: Default.....	58
Abbildung 22: vier Werte.....	58
Abbildung 23: drei Werte.....	58
Abbildung 24: zwei Werte.....	58
Abbildung 25: Color-Picker.....	59
Abbildung 26: Resultat des Projektes.....	60
Abbildung 27: Diagramm mit hohem Wert Liechtenstein.....	62
Abbildung 28: Diagramm mit Wert Luxemburg.....	63
Abbildung 29: Mögliche Diagrammanpassungen (Quelle: [Boundford, 2000], S.68).....	67

## Tabellenverzeichnis

Tabelle 1: Verwendete Programme.....	24
Tabelle 2: Auflistung der Dateien.....	36

## Abkürzungsverzeichnis

BfS	<b>B</b> undesamt für <b>S</b> tatistik
dpi	<b>D</b> ots <b>p</b> er <b>i</b> nch, Mass für die Bildschirmauflösung
ETH	<b>E</b> idgenössische <b>T</b> echnische <b>H</b> ochschule
GIS	<b>G</b> eografisches <b>I</b> nformations <b>S</b> ystem
HSV	<b>H</b> ue, <b>S</b> aturation, <b>V</b> alue
JPG	<b>J</b> oint <b>P</b> hotographic Experts <b>G</b> roup, Rasterformat
lpi	<b>L</b> ines <b>p</b> er <b>i</b> nch, Mass für die Rasterweite
PHP	PHP-Hypertext Preprocessor
RGB	<b>R</b> ot, <b>G</b> rün, <b>B</b> lau
SHP	Shapefile (ESRI-Format)
SQL	<b>S</b> tructured <b>Q</b> uery <b>L</b> anguage
SVG	<b>S</b> calable <b>V</b> ector <b>G</b> raphics
URL	<b>U</b> niform <b>R</b> esource <b>L</b> ocator
W3C	<b>W</b> orld <b>W</b> ide <b>W</b> eb <b>C</b> onsortium
XML	<b>e</b> Xtensible <b>M</b> ark-up <b>L</b> anguage

# 1 Einleitung und Problemstellung

## 1.1 Ausgangslage

Die Eidgenössische Technische Hochschule Zürich (ETH Zürich) wurde im Jahre 1855 gegründet. Sie ist eine Bildungs- und Forschungsinstitution der Schweizerischen Eidgenossenschaft. 18'000 Personen aus dem In- und Ausland studieren, forschen und arbeiten an der ETH Zürich. [URL: ETH Zürich]

Gemäss dem Jahresbericht 2003 der ETH Zürich betrug deren Anzahl an Studierenden im Jahre 2003 12'626, wovon 2'525 ausländische Studierende waren. In diesen Zahlen sind die Diplom-, Doktorats- und Nachdiplomstudierenden enthalten. Jeder Studierende muss bei seiner Immatrikulation ein Formular mit einigen Angaben über seine Person ausfüllen. Zu diesen Angaben gehören der Wohnort vor Studienbeginn, die Nationalität, das Geschlecht, der Studiengang, etc. Diese Informationen werden seit 1990 in eine Oracle-Datenbank<sup>1</sup> eingetragen und verwaltet. Jeder Studierende wird mit über 50 Attributen beschrieben. Wobei einige Attribute ausschliesslich der Datenverwaltung dienen und zusätzlich zu den ausgefüllten Formularen eingetragen werden.

Wenn man bedenkt, dass alle neueintretenden Studierenden der ETH Zürich mit über 50 Einträgen in der Datenbank beschrieben werden, und das seit 14 Jahren, kann man sich gut vorstellen, dass die Datenbank ein immenses Ausmass angenommen hat.

Die ETH Zürich will nun diese Datengrundlage nutzen und mit einer geeigneten Visualisierung diese Daten einem breiten Publikum zugänglich machen, wobei die Herkunft der Studierenden im Vordergrund steht. Als Veröffentlichungsmedium stellt sich das Internet als geeignet dar, da in der heutigen Zeit die breite Öffentlichkeit das Internet gut kennt und oft benutzt. Die Interaktivität der Karte spielt eine grosse Rolle, da der Benutzer so die Daten, die er betrachten will, selber auswählen und zusammenstellen kann.

## 1.2 Auftrag und Zielsetzung

Mein Auftrag ist die Erstellung eines Prototyps, der in erster Linie die Herkunft der Studierenden auf geeignete Weise visualisiert, indem er die statistischen Daten der Datenbank nutzt. Es soll eine interaktive Bildschirmkarte erstellt werden, die im Internet präsentiert werden kann. Wichtig ist, dass mittels Skripten auf die Datenbank zugegriffen werden kann. So werden allfällige Änderungen der Daten in der Datenbank direkt in die Karte übertragen.

---

<sup>1</sup> Siehe Glossar: Stichwort „Datenbank“

Die Interaktion der Karte spielt eine grosse Rolle, da der Benutzer die darzustellenden Daten selber bestimmen kann. Dies wird ermöglicht, indem dem Benutzer mittels Selektionslisten eine Auswahl an Datengruppen vorgegeben wird, die er sich beliebig zusammenstellen kann.

Die Herkunft der Studierenden soll in Landkarten auf geeignete Weise dargestellt werden. Dazu ist zu beachten, dass die ETH-Studierenden mehrheitlich aus der Schweiz, einige aber auch aus dem Ausland kommen. Aus diesem Grund sollen verschiedene Karten zur Verfügung gestellt werden, zwischen denen der Benutzer auswählen kann. Eine Schweizer-, eine Europa- und eine Weltkarte würden sich dafür gut eignen.

Es wird gewünscht, dass der Benutzer zu folgenden Themenbereichen eine Auswahl treffen kann:

- Herkunft der Studierenden (Wohnort vor Studienbeginn oder Nationalität)
- darzustellendes Jahr
- Neueintritte / Bestand
- Studienkategorie (Diplom-, Nachdiplomstudierende, Doktorierende)
- Studienfach
- Geschlecht

Die Aufteilung der Herkunft in „Wohnort vor Studienbeginn“ und „Nationalität“ ist nötig, da diese Attribute unterschiedliche Werte aufweisen können. Das ist beispielsweise bei einem italienischen Absolventen der Fall, der in Deutschland studiert hat und anschliessend an der ETH Zürich doktoriert.

Die Unterscheidung zwischen Bestand und Neueintritten sieht so aus, dass unter den Bestand alle Studierenden vom ersten bis zum letzten Semester eines Studienganges fallen, wogegen die Neueintritte nur die Studierenden berücksichtigt, die ins erste Semester eintreten.

Meine Diplomarbeit soll eine Möglichkeit aufzeigen, wie die aufgeführten Anforderungen der ETH funktionell umgesetzt werden können. Die Karte ist benutzerfreundlich zu gestalten und allfällige Grenzen der Technik sind ausführlich zu beschreiben.

Das Zielpublikum des Projektes sind einerseits die Auftraggeber selber, die das Projekt für statistische Analysen verwenden könnten. Andererseits soll das Projekt zu einem späteren Zeitpunkt der breiten Öffentlichkeit auf dem Internet zur Verfügung gestellt werden.

### 1.3 Aufbau der Arbeit

Die vorliegende Arbeit ist so aufgebaut, dass zuerst einige theoretische Grundlagen zum Thema aufgeführt werden. Die allgemeine Theorie über die Visualisierung statistischer Daten und die Webkartographie sind Bestandteile dieses Kapitels. Danach werden die technischen Grundlagen und das der Arbeit zugrunde liegende Konzept erläutert. Das Konzept beschreibt die Inhalte und die Gliederung der Karte, aber auch die Wahl der Techniken.

In einem weiteren Kapitel wird auf die verwendeten Materialien und die Methodik eingegangen. Dabei werden in einem ersten Schritt die Datengrundlagen beschrieben. Es wird sodann aufgezeigt, welche Daten für das Projekt benutzt wurden und wie sie vorliegen. Die verwendeten Programme und der Arbeitsablauf werden ebenfalls erläutert

Schliesslich wird spezifisch auf die Umsetzung des Projektes eingegangen. Hier werden die einzelnen Schritte, von der Datenaufbereitung über die Datenbankabfragen bis hin zur Visualisierung der Ergebnisse in der SVG-Karte, genau beschrieben.

Im anschliessenden Kapitel wird das Resultat diskutiert. Auf die Resultate für die Schweiz und das Ausland wird getrennt eingegangen. Es werden auch die Browser-, SVGViewer<sup>2</sup>- und Betriebssystemkompatibilität und die Flexibilität des Projektes erläutert.

Im letzten Kapitel wird auf mögliche Erweiterungen des Projektes eingegangen und auf gewisse Optimierungsmöglichkeiten des Projektes hingewiesen.

---

<sup>2</sup> Siehe Glossar: Stichwort „Viewer“

## 2 Theoretische Grundlagen

Das Hauptthema dieser Arbeit ist die Visualisierung statistischer Daten in einer Karte. Die theoretischen Grundlagen dieser Thematik wird im Folgenden kurz erläutert. Da die Visualisierung in einer Webkarte vorgenommen wird, werden auch einige Grundaspekte der Webkartographie aufgeführt.

### 2.1 Visualisierung statistischer Daten

Die Visualisierung statischer Daten kann auf verschiedene Arten geschehen. Die Theorie der thematischen Kartographie liefert die Grundregeln, nach welchen Kriterien statistische Daten untersucht und deren passende Visualisierung gefunden werden. Die Wahl der Darstellungsmethode hängt von den gegebenen Daten ab. In einem ersten Schritt muss die Datengrundlage genau studiert werden, bevor die Darstellungsmethode gewählt werden kann. Die Punkte, die untersucht werden müssen sind gemäss Hurni<sup>3</sup>:

- der Streubereich der abzubildenden Merkmalswerte,
- die Frage, ob auch Mengen von der Grösse Null darzustellen sind,
- die mögliche Länge der graphischen Variablen je nachdem, ob es sich um quantitative (Absolut- oder Relativwerte), qualitative oder ob es sich um geordnete Werte handelt,
- Die Frage, ob die Information zählbar, messbar oder nur schätzbar wiederzugeben ist,
- die verschiedenen Interpretationsmöglichkeiten von Mengendarstellungen,
- der Wertvergleich verschiedener Formen.

Auf die Darstellung von qualitativen und quantitativen Werten wird im Weiteren genauer eingegangen.

#### 2.1.1 Darstellung qualitativer Werte

Qualitative Werte stellen die Art, Gattung oder Eigenschaft eines Objektes dar [Imhof, 1972]. Mengen spielen hier keine Rolle. Qualitative Karten geben nur die Objektqualität zu erkennen und beantworten damit die Frage „Was ist wo?“. Ein Beispiel dafür sind politische Karten, die beispielsweise Staatsgebiete darstellen. Als Gestaltungsmittel dienen Flächenfarben, längs der Grenzen oft lineare Signaturen oder Farbsäume. Die Kartengrundlage beschränkt sich auf die wichtigste Topographie (Orte, Verkehrs- und Gewässernetz); Verwaltungssitze werden durch lokale Signaturen oder Unterstreichungen hervorgehoben [Hake et al., 2002].

---

3 [Hurni, 2002]

### 2.1.2 Darstellung quantitativer Werte

Quantitative Karten bringen Grössen, Mengen, Werte usw. des Objektes zum Ausdruck und beantworten damit die Frage „Wieviel ist wo?“ [Hake et al., 2002].

Zur Wahl der geeigneten Darstellungsmethode quantitativer Werte müssen folgende Fragen beantwortet werden:

- Handelt es sich um Absolut- oder Relativwerte?
- Welche Darstellungsart wird gewählt: Stetige Mengendarstellung oder Intervalldarstellung?
- Falls Intervalldarstellung: Wie werden die Grenzen gewählt (Klassierungsart)

#### *Absolut- und Relativwerte*

Bei quantitativen Werten wird die Unterscheidung zwischen Relativ- und Absolutwerten gemacht. Relativwerte stellen Anteile oder Relationen dar (z.B. Zahl von Arbeitslosen pro Einwohnerzahl einer Gemeinde), und Absolutwerte selbständige Mengen (z.B. Anzahl Einwohner einer Gemeinde). Je nach Klasse werden andere Darstellungsmethoden gewählt. Absolutwerte werden mittels lokalen Signaturen, die den jeweiligen Flächen zugeordnet sind, dargestellt. Für die Darstellung relativer Grössen wird die Flächenfüllung verwendet, die mittels Tonwert-Variationen voneinander getrennt werden [Hake et al., 2002].

#### *Stetige Mengendarstellung oder Intervalldarstellung*

Zur Darstellung quantitativer Informationen wird zwischen stetiger Mengendarstellung und Intervalldarstellung unterschieden. Bei der stetigen Mengendarstellung werden die Werte beispielsweise direkt mit Zahlen in der Karte eingetragen. Die Werte können beliebig ganze, positive, negative und Dezimalzahlen annehmen. Wertintervalle sind Grössen, die für einen Bereich (von... bis...) stehen. In der Regel ist die Wiedergabe mit stetigen Mengen vorzuziehen, da nur so die volle Information vermittelt wird. Der Übergang auf eine Intervallskala bedeutet immer das Verlassen der quantitativen Information und ein Übergehen auf Grössenklassen, also auf eine geordnete Skala. Damit ist in der Regel ein Informationsverlust verbunden [Hurni, 2002].

#### 2.1.2.1 Stetige Mengendarstellung

Für die stetige Mengendarstellung gibt es viele verschiedene Methoden (siehe [Hurni, 2002]). Es werden hier nur einige Beispiele erwähnt.

- Die Werte werden direkt mit Zahlen in die Karte eingetragen. Selbst geringe Wertunterschiede können dadurch genau abgelesen werden.

- Es wird eine einzige Werteinheit gewählt, mit der alle stetigen Werte umgesetzt werden müssen. Die Grössenvorstellung wird durch die Wiederholung dieser Zeichen erschaffen.
- Statt einer Werteinheit werden verschiedene, gut abgestufte Werteinheiten gewählt, aus denen alle nötigen Werte zusammengesetzt werden können.

Wichtig bei der stetigen Mengendarstellung ist, dass alle Werte einzeln aus der Karte gelesen werden können.

### **2.1.2.2 Intervalldarstellung**

Bei der Intervalldarstellung wird mit Wertklassen gearbeitet. Die einzelnen Werte können somit nicht mehr aus der Karte gelesen werden. Die Wahl der Klassierungsart, der Zahl der Stufen und der Klassenintervalle sind wichtige Kriterien, die bei der Intervallbildung berücksichtigt werden müssen.

#### *Klassierungsart*

Bei der Festlegung der Intervallgrenzen müssen der Streubereich und die Häufigkeitsgruppen der Daten genau betrachtet werden. Die Häufigkeitsverteilung ergibt sich zum Beispiel aus der Auflistung aller Daten in einem Histogramm. Die dort auftretenden Minima und Maxima legen die Grundlage für die Berechnung der einzelnen Gruppen, die nach mathematischen Regeln erfolgt. Dazu muss aber zuerst die Zahl der Wertstufen festgelegt werden.

#### *Zahl der Wertstufen*

Die Anzahl der Stufen hängt von der Signatur und des Minimal- und Maximalwertes der Daten ab. Sie sollte bei mehrfarbiger Darstellung mit Kombination von Farbton- und Helligkeitsvariation zwischen 3 bis 7 Stufen betragen, weil sonst das Unterscheidungsvermögen zwischen benachbarten Stufen nicht mehr ausreichend gewährleistet ist [Hurni, 2002].

#### *Wahl der Klassenintervalle*

Es gibt verschiedene Klassierungsarten für die Intervalle. Die Klassierungsart wird der Verteilung der Häufigkeitsgruppen angepasst. Es empfiehlt sich, alle quantitativen Werte nach ihrer Grösse zu ordnen. Damit erhält man eine Verteilung, welche erlaubt, die Klassengrenzen gezielt zu setzen. Wenn die Klassierungsart nach reiner Zufälligkeit bestimmt wird und nicht auf die vorliegenden Daten abgestimmt ist, wird die Aussage der thematischen Karte verfälscht.

Häufige Klassierungsarten sind „gleiche Intervalle“ und „Quantile“:

- **gleiche Intervalle:** Es wird von einer konstanten Intervallbreite aller Wertgruppen ausgegangen. Dabei wird die Differenz des Maximal- und des Minimalwertes berechnet und diese Zahl durch die Anzahl Intervalle geteilt. Dies ergibt die Breite einer Intervallgruppe.

$$\text{Klassenbreite} = \frac{(\text{Maximalwert} - \text{Minimalwert})}{\text{Klassenanzahl}}$$

Bei dieser Berechnungsmethode sind die Intervallgrenzen lückenlos. Es kann jedoch vorkommen, dass in einer Klasse beispielsweise nur ein Wert liegt und in einer anderen Klasse fast 90% aller Werte liegen. Dies kann zu Fehlinterpretationen der Daten führen. Ausreissern der Datenreihe wird so eine zu grosse Bedeutung zugesprochen.

- **Quantile:** Das Prinzip der Quantile sorgt dafür, dass alle Klassen im Idealfall die gleiche Anzahl Werte enthalten. Jedoch kann es durch Wertgleichheit und eine nicht durch die Klassenanzahl teilbare Datenmenge zu einer Verschiebung dieser idealisierten Verteilung kommen. Die Anzahl der Daten, die innerhalb einer Klasse zu liegen kommen wird mit folgender Formel berechnet:

$$\text{Anzahl der Werte einer Klasse} = \frac{\text{Anzahl aller Werte}}{\text{Klassenanzahl}}$$

Zur Verteilung der Daten und für die Berechnung der Intervallgrenzen müssen zuerst alle Werte aufsteigend (kleinster Wert zuerst) sortiert werden. Mit der Anzahl Klassen können nun die Grenzen der einzelnen Intervalle bestimmt werden. Dabei werden die effektiven Werte des ersten und des letzten Klassenmitgliedes als Intervallgrenzen übernommen. Aus dieser Klassifizierung ergibt sich eine unterbrochene Skala, da die Bereiche zwischen den angrenzenden Unter- und Obergrenzen zweier Klassen keiner dieser beiden zugeordnet werden können.

Dies kann den Betrachter einer Karte, besonders wenn es sich um einen Laien handelt, leicht irritieren. Deshalb kann auf eine solche unterbrochene Klassifizierung verzichtet werden, indem die entsprechenden Grenzwerte aus den Werten der jeweils angrenzenden Unter- und Obergrenze gemittelt werden.

### 2.1.3 Darstellungsmethode

Die Wahl der geeigneten Darstellungsmethode von Objekten und Werten ist eine wichtige Grundlage für eine gute Karte. In den folgenden Abschnitten wird auf Signaturen und Diagramme eingegangen.

#### 2.1.3.1 Signaturen

Statistische Daten können in der thematischen Kartographie durch zahlreiche Signaturen dargestellt werden. Die meisten lassen sich gemäss Arnberger<sup>4</sup> auf drei Signaturgrundarten zurückführen:

- **Punktartige oder Figurensignaturen** stellen ortsgebundene Objekte adäquat dar. Sie können in Form von geometrisch geformten (Kreise, Rechtecke, etc.) oder bildhaften (vereinfachten Miniaturbildchen) Lokalsignaturen (Gebäude, Bäume, etc.) auftreten.
- **Linienartige Signaturen** geben linienhaft reduzierte oder streckenbezogene Objekte adäquat wieder. Mittels Linien ist es möglich, flächig darstellbare Diskreta (z.B. Kulturarten, etc.) grundrisstreu oder -ähnlich abzugrenzen oder linienhafte Erscheinungen darzustellen.
- **Flächenartige Signaturen** stellen raumerfüllende oder raumbezogene Objekte adäquat dar. Es stehen folgende flächenfüllende Signaturen zur Verfügung: Flächenraster, Flächenmuster, Strukturraster, Farbflächen und Flächentöne.

Ein ganz hervorragendes Mittel, um flächenhaft verbreitete Objekte sowohl in ihren qualitativen als auch quantitativen Eigenschaft darzustellen, ist die Flächentönung durch Farben, welche zwischen dem Weisspol und dem Schwarzpol der Farbkugel liegen. [Arnberger, 1997]

Eine häufige flächenartige Signatur, die durch die Flächentönung bestimmt ist, stellt die Choroplethendarstellung<sup>5</sup> dar. Diese Methode erfordert Relativwerte und eine Intervalldarstellung. Die aus den Einzelwerten gebildeten Intervalle unterscheiden sich durch graphische Variation von Farbtönen oder -helligkeiten. Dabei werden die ganzen Flächen eingefärbt. Ein Beispiel für Choroplethen zeigt Abbildung 2.

---

4 [Arnberger, 1997]

5 Siehe Glossar: Stichwort „Choroplethen“

### 2.1.3.2 Diagramme

„Diagramme sind graphische Mittel zur Wiedergabe quantitativer Daten, vor allem statistischer Grössen. Graphische Variationen ändern Art und Umfang der Aussage nicht, können aber die Verdeutlichung fördern“ [Hake et al., 2002].

Diagramme bieten eine bessere und genauer ablesbare Information über absolute und relative Wertgrössen als Signaturen. Diagramme bieten viel umfangreichere Möglichkeiten einer genaueren quantitativen und untergliederten qualitativen Aussage, als alle anderen signaturhaften Darstellungsformen. Die Gestaltung der Diagramme unterliegt den Gesetzmässigkeiten und dem logischen Aufbau der graphischen Darstellung, wie sie die Statistik verwendet. [Arnberger, 1997]

Die verschiedenen Diagramme zeichnen sich durch unterschiedliche Eigenschaften aus. Für die Wahl des Diagrammtyps, der den Zweck der Karte ausdrücken muss, kommen gemäss Hurni<sup>6</sup> u. a. folgende Kriterien in Frage:

- Ist ein Vergleich der Gesamtmengen möglich?
- Ist ein Vergleich der sich entsprechenden Teilmengen zwischen verschiedenen Diagrammen möglich?
- Können Nullmengen dargestellt werden?
- Kann der Anteil der Teilmenge an der Gesamtmenge eines Diagramms erfasst werden?
- Können negative Werte dargestellt werden?
- Etc.

Da die Auflistung aller Diagrammartentypen an dieser Stelle zu ausführlich wäre, werden nur zwei Beispiele vorgestellt, die stellvertretend für alle anderen Diagrammtypen das Prinzip der Diagrammdarstellung erläutern sollen.

Häufig verwendete Diagrammartentypen sind Kreis- und Stabdiagramme:

- **Kreisdiagramm:** Voraussetzung hierfür sind Absolutwerte. Die Wertunterschiede werden mit dem Grössenunterschied der verschiedenen Kreise dargestellt. Die Kreise selber können weiter in Sektoren unterteilt sein, um zusätzliche Verhältnisse zu visualisieren. Die Werte, die in den Sektoren visualisiert werden, stellen dann aber keine Absolutwerte, sondern Anteile der Gesamtmasse dar.

Ein Beispiel für Kreisdiagramme zeigt Abbildung 3.

---

6 [Hurni, 2002]

- **Stabdiagramme:** Es sind ebenfalls Absolutwerte vorausgesetzt. Die Wertunterschiede werden mittels unterschiedlicher Stablänge oder -breite visualisiert. Die Stäbe selber können wieder in Unterkategorien unterteilt sein, die wieder Anteile an der Gesamtmasse darstellen. Eine häufige Anwendung ist das Histogramm, bei dem die Stäbe aneinander gereiht werden. Somit können Werte direkt verglichen werden. Oftmals treten Histogramme in Verbindung mit einer Wertskala auf.

Ein Beispiel für Stabdiagramme zeigt Abbildung 1.

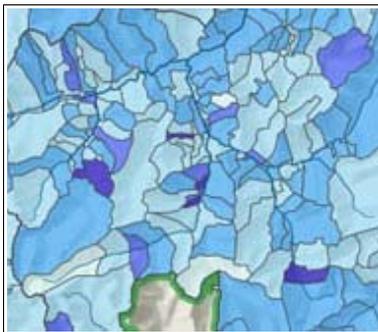


Abbildung 1: Choroplethen

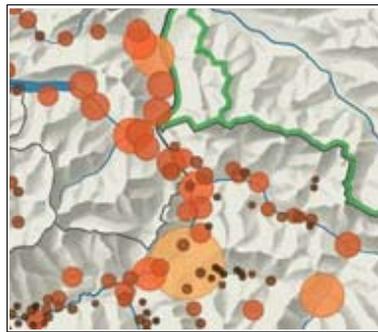


Abbildung 2: Kreisdiagramm

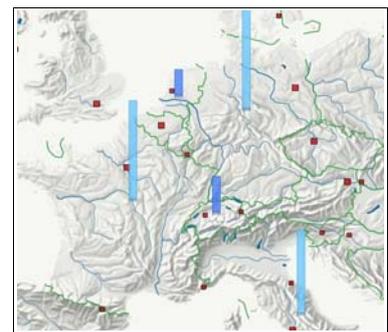


Abbildung 3: Stabdiagramm

Zu beachten ist, dass Diagrammfiguren oftmals derart grosse Räume überdecken, dass sie zwar unter Umständen noch lagerichtig eingetragen werden können – d.h., dass sich der Mittelpunkt des Diagramms mit der genauen Ortslage des darzustellenden Objektes deckt – immer aber andere Karteninhalte verdeckt oder verdrängt werden. [Arnberger, 1997]

## 2.2 Webkarten

Die Entwicklung der Webkartographie steht in direktem Zusammenhang mit der rasanten Entwicklung des Mediums Internet. In den nächsten Abschnitten wird zuerst eine kurze Zusammenfassung der geschichtlichen Entwicklung der Webkartographie aufgeführt. Anschliessend werden die verschiedenen Arten von Webkarten und die Vor- und Nachteile einer Webkarte aufgeführt.

### 2.2.1 Geschichte der Webkartographie

Das Internet<sup>7</sup> ging Ende der 1960er Jahre aus dem militärischen ARPANET<sup>8</sup> hervor und wurde später benutzt, um Universitäten und Forschungseinrichtungen zu vernetzen, zunächst in den USA, später dann auch weltweit. Es wurde also ausschliesslich von Experten bedient. Starke

<sup>7</sup> Vgl. Definition im Glossar

<sup>8</sup> **A**dvanced **R**esearch **P**rojects **A**gency **N**etwork. Computernetz des U.S. Departement of Defense

Auftrieb erhielt das Internet seit Anfang der 1990er Jahre durch das World Wide Web<sup>9</sup>, kurz WWW. Es wurde von Tim Berners-Lee entwickelt. Mit Webbrowsern<sup>10</sup> konnten nun auch Laien auf das Netz zugreifen. [URL: net-lexikon]

Das Internet kann von jedem Computer, der mit den nötigen Werkzeugen ausgerüstet ist, gestartet werden. Dies bewirkte, dass die Benutzerzahl in den letzten Jahren rasant zunahm. Mit der wachsenden Zahl von Nutzern wurden auch viele kommerzielle Angebote ins Netz gestellt, darunter auch kartografische Projekte. Das Anzeigen von Informationen mit Raumbezug in einer Karte liegt nahe. Die Webkartographie gewinnt darum, parallel zur Verbreitung des Internets, immer mehr an Bedeutung. Mit den Jahren sind die Projekte der Webkartographie immer grösser und dynamischer geworden. Am Anfang waren die Internetkarten statisch. Interaktionen waren nicht oder nur eingeschränkt möglich. In den letzten Jahren wurden die Webkarten immer interaktiver. In der heutigen Zeit findet der Benutzer kaum noch Projekte, die keine Interaktion zulassen.

### 2.2.2 Arten von Web-Karten

Unter Web-Karte versteht man jede Art von Karte, die im Internet zu finden ist. In der Webkartographie unterscheidet man grundsätzlich zwischen statischen und dynamischen Karten, die wiederum in view-only und interaktive Karten unterteilt werden [Sieber, 2003]:

- **Statische Karten** stellen meistens Zustandsdarstellungen zu einem bestimmten Zeitpunkt in einer festgelegten räumlichen Umgebung dar. Der Benutzer einer solchen Karte erhält lediglich Informationen über einen Zeitpunkt oder ein Gebiet. Die einfachste Methode, eine statische Karte zu erzeugen, ist das Scannen einer Papierkarte [Cartwright, 2003].
- **Dynamische Karten** zeigen räumliche oder zeitliche Veränderungen von Objekten auf. Dies kann zum Beispiel ein virtueller Flug über ein Gelände sein.
- **View-only Karten** können vom Benutzer lediglich betrachtet werden. Einflüsse auf das Kartenbild hat der Benutzer keine. Sie können statisch oder dynamisch sein.
- **Interaktive Karten** geben dem Benutzer die Möglichkeit, auf die Karte Einfluss zu nehmen und Veränderungen vorzunehmen.

Die Wahl der Art der Web-Karte wird immer durch den Verwendungszweck und die Anforderungen an das zu erstellende Projekt vorgegeben.

---

<sup>9</sup> Siehe Glossar: Stichwort „World Wide Web“

<sup>10</sup> Siehe Glossar: Stichwort „Webbrowser“

### 2.2.3 Vor- und Nachteile

So schön und fortschrittlich sich die Entwicklung der Kartographie im Internetbereich auch anhört, gibt die Technik Grenzen vor, die immer wieder zu Problemen führen. Obwohl die Techniken permanent weiterentwickelt und den Bedürfnissen der Kartographen angepasst werden, existieren Hindernisse, die bei Papierkarten nicht auftreten.

Eine Webkarte wird immer in einem Bildschirm ausgegeben und der Computer ist das Übertragungsmedium der Daten. Diese Fakten können bei der Erstellung einer Webkarte Probleme bereiten und müssen bereits bei der Konzeption eines Projektes berücksichtigt werden. Die zu beachtenden Punkte können gemäss Schnabel<sup>11</sup> in drei Gruppen aufgeteilt werden:

- **Hardwareprobleme**

- Internet-Verbindungstechnik: Die Übertragungsgeschwindigkeit und -rate variiert je nach verwendetem Übertragungsmedium stark.
- Anzeigemedium Bildschirm: durchschnittliche Bildschirmauflösung 72 dpi<sup>12</sup> (neuere 96dpi); Die Anzeige ist immer konstant (Bildschirmgrösse); Farben werden je nach Bildschirm anders dargestellt.

- **Softwareprobleme**

- Browserkompatibilität: Dem Benutzer stehen verschiedene Browser zur Verfügung. Jeder Browser interpretiert die Tags<sup>13</sup> unterschiedlich; einige Formate benötigen für die richtige Interpretation ein Plugin<sup>14</sup>.

- **Kartengestaltungsprobleme**

- Farben: Die Farben einer Karte sollten gut abgestimmt sein. Geringe Farbunterschiede z. B. bei Intervallangaben sollten vermieden werden.
- Lesbarkeit: Eine Karte sollte nicht mit Informationen überfüllt werden. Durch die Interaktion (Ein- und Ausschalten von Information) kann dies schnell vorkommen.

In einer Webkarte können sowohl Raster- als auch Vektorobjekte eingebunden werden. Bei der Verwendung dieser Formate, müssen folgende Gegebenheiten beachtet werden:

---

11 [Schnabel, 2004]

12 **dots per inch**: Masseinheit zur Angabe der Auflösung graphischer Ein- / Ausgabeeinheiten

13 Siehe Glossar: Stichwort „Tag“

14 Siehe Glossar: Stichwort „Plugin“

- **Rasterformate**

Die Grösse und die Anzahl der Pixel bestimmen die mit dem Speichern festgelegte Auflösung des Bildes. So entstehen bei grossen Bildern mit einer hohen Auflösung grosse Dateien<sup>15</sup>, die viel Speicherplatz benötigen. Beim Vergrössern (Einzoomen) eines Rasterbildes entstehen Treppenstufeneffekte, was die Darstellung negativ beeinflusst. Der Vorteil von Rasterbildern ist, dass sie in jedem Browser ohne zusätzliche Erweiterungen pixelgenau dargestellt werden.

- **Vektorformate**

Die Objekte werden durch mathematische Funktionen beschrieben. Diese können einzeln bearbeitet und beliebig skaliert werden. Die Verknüpfung der Objekte mit Sachdaten ist möglich und es entstehen nur kleine Dateien bei grossen Bildern. Der Nachteil von Vektordaten ist, dass sie ein Plugin zur Darstellung im Browser benötigen.

#### 2.2.4 Darstellung einer Webkarte

Für die Darstellung einer Webkarte gibt es einige Grundregeln, die bei der Erstellung eines Projektes berücksichtigt werden müssen. Der Benutzer soll sich möglichst schnell und ohne grosse Hilfe in der Karte zurechtfinden und alle möglichen Interaktionen verstehen.

Die Geometrie muss den grössten Teil des Bildschirmes einnehmen. Die restlichen Kartenbestandteile müssen so angeordnet werden, dass ihre Funktion und Zugehörigkeit sofort erkennbar sind. Beispielsweise wird eine allfällige Navigationshilfe in der Nähe der Geometrie platziert.

Falls der Benutzer die Einstellungen der Karte verändern kann, soll die Anordnung der verschiedenen Einstellungen gut gewählt sein. Dafür gibt es zwei Möglichkeiten:

- Dem Benutzer kann eine Reihenfolge der vorzunehmenden Einstellungen vorgegeben werden. Er wird Schritt für Schritt durch die Einstellungen geführt, die er der Reihe nach verändern kann. Dies wird erreicht, indem nicht alle Einstellungen auf einmal eingeblendet werden, sondern in Abhängigkeit voneinander erscheinen.
- Dem Benutzer werden alle möglichen Interaktionen von Anfang an zur Verfügung gestellt. Die Reihenfolge der getätigten Einstellungen ist beliebig.

Die Einstellungen zur Karte, meistens in Form von Selektionslisten und Schaltflächen, müssen unter sich und in Bezug zur Karte logisch angeordnet werden. Der Benutzer soll auf den ersten Blick den Zweck und den Bezug der Interaktionen erkennen.

---

<sup>15</sup> Siehe Glossar: Stichwort „Datei“

Falls trotzdem Unklarheiten bei der Bedienung von Kartenelementen auftreten können, kann mit einem Hilfetext, in dem Anweisungen beschrieben sind, gearbeitet werden.

## 3 Konzept

Dieses Kapitel beschreibt das Konzept, das den Inhalt der Karte definiert und deren Anordnung auf dem Bildschirm erläutert. Die Wahl der Techniken wird ebenfalls beschrieben.

### 3.1 Inhalt der SVG-Karte

Die Herkunft der Studierenden stellt das Hauptthema der Karte dar. Es wird in „Wohnort vor Studienbeginn“ und „Nationalität“ aufgeteilt.

Diese zwei Hauptthemen werden in drei verschiedenen Karten dargestellt, die der Benutzer beliebig auswählen kann:

- Schweizer Landeskarte, aufgeteilt in Bezirksgrenzen
- Europakarte, aufgeteilt in Länder
- Weltkarte, aufgeteilt in Länder

Aus Darstellungsgründen wird die Herkunft der Studierenden in der Schweizerkarte auf der Ebene der Bezirksgrenzen visualisiert. Die Einteilung der Schweizerkarte in Gemeindegrenzen hätte zur Folge, dass das Auge des Betrachters mit zu viel Information konfrontiert würde. Das Lesen des Karteninhaltes wäre dadurch erschwert.

Die Wahl der Darstellungsmethode fiel zu Gunsten der Choroplethendarstellung aus. Das Einfärben der Flächen (Bezirke oder Länder) eignet sich für dieses Projekt besonders gut, da so die Wertunterschiede der einzelnen Flächen gut sichtbar sind. Die darzustellenden Zahlen sind auf Grund der Choroplethendarstellung Relativwerte und stehen in Bezug zu der Einwohnerzahl eines Bezirkes bzw. eines Landes. Da die Studierenden mehrheitlich aus der Schweiz kommen, wird die Schweizerkarte als Defaultkarte eingesetzt.

Ein Sonderfall stellt die Einstellung „Nationalität“ dar: Die Nationalität der Studierenden kann nicht in der Schweizerkarte dargestellt werden, da die verschiedenen Länder von Interesse sind. Wenn also die Einstellung „Nationalität“ ausgewählt ist, wird die Schweizerkarte ausgeblendet. Der Benutzer kann aber immer noch zwischen der Europa- und der Weltkarte auswählen.

Weiter werden verschiedene Elemente in die Karte integriert, die nachfolgend beschrieben sind.

### *Navigation*

Damit der Benutzer den Überblick in der Geometrie nicht verliert, wird eine Navigationshilfe implementiert, die es u.a. ermöglicht, in die Karte ein- und auszuzoomen. Dieses Werkzeug ist vor allem in der Weltkarte sehr nützlich, da so kleinere Länder, die in der Übersicht nicht zu sehen sind, hervorgehoben werden können.

### *Selektionslisten*

In der Webkarte werden mehrere Selektionslisten zur Verfügung gestellt, die die Datenbankabfrage steuern und deren Einstellungen der Benutzer beliebig im Rahmen der vorgegebenen Inhalte zusammenstellen kann:

- darzustellendes Jahr (1990-2004)
- Status der Studierenden (Bestand oder Neueintritte)
- Studienkategorie (alle Studienkategorien, Diplom-, Nachdiplomstudierende und Doktorierende)
- Studiengang (alle Studiengänge, Liste der einzelnen Studiengängen)
- Geschlecht (gesamt, Männer, Frauen, Vergleich Männer - Frauen)

Weiter werden zwei Selektionslisten implementiert, die die Klassierung der übermittelten Daten steuern und die der Benutzer wiederum beliebig im Rahmen der sinnvollen Vorgaben verändern kann:

- Klassierungsart (gleiche Intervalle oder Quantile)
- Anzahl der Klassen (2-5)

### *Color-Picker*

Im Zusammenhang der Choroplethendarstellung wird ein Color-Picker zur Verfügung gestellt, der es erlaubt, die Farben der Darstellung zu ändern. Im Color-Picker kann der Benutzer eine neue Farbe auswählen. Die Karte wird anschliessend in dieser Farbgebung eingefärbt. Der Farbverlauf der unterschiedlichen Klassen wird automatisch der ausgewählten Farbe angepasst.

### *Diagramm*

Die Unterschiede der einzelnen Werte der Bezirke bzw. Länder kommen auf Grund der Intervalldarstellung nicht zum Ausdruck. Daher wird ein Diagramm implementiert, das alle Werte der Bezirke bzw. Länder der Grösse nach auflistet. So können die einzelnen Werte untereinander verglichen werden.

### Anzeigen von Zusatzinformationen

Da der Benutzer mit Sicherheit nicht alle Bezirks- und Ländernamen der Karte auswendig kennt, werden beim Überfahren des Pfades mit dem Maus-Cursor die dazugehörigen Namen angezeigt. Ebenfalls wird die Einwohnerzahl und die Anzahl Studierenden angezeigt.

## 3.2 Layout des Projektes

Die Bildschirmkarte weist drei Hauptteile auf:

- Geometrie (Hauptkarte), aufgeteilt in Karte, Legende und Diagramm
- Navigation
- Einstellungen zur Karte

Die Anordnung der Elemente in der Karte sieht folgendermassen aus:

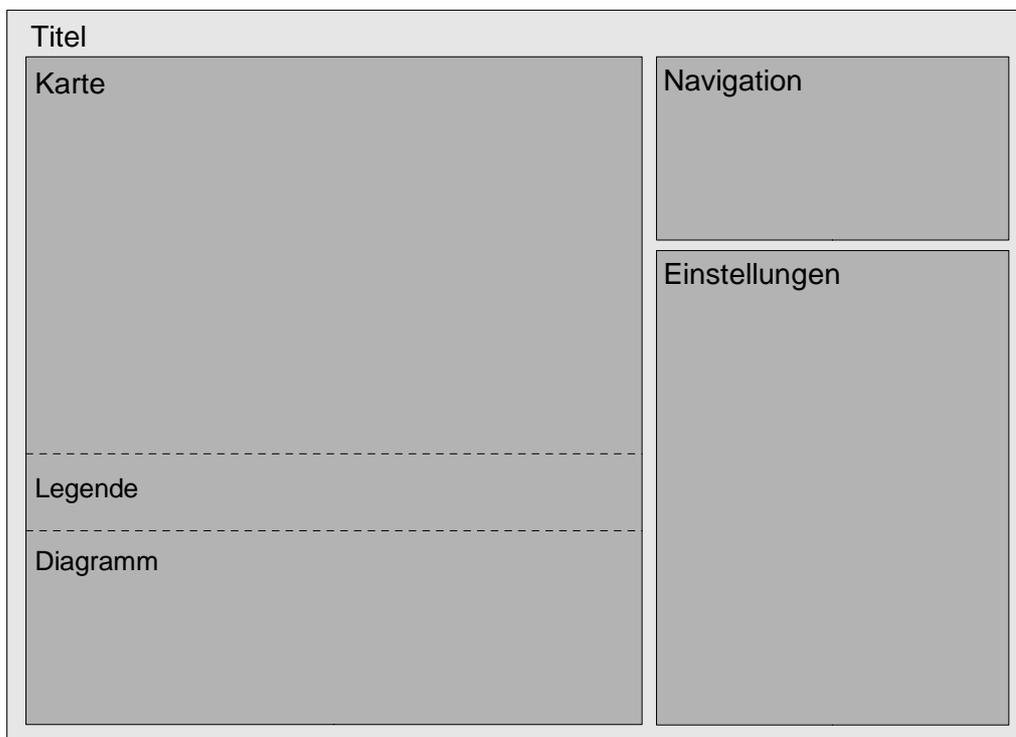


Abbildung 4: Layout der Karte

Die Grösse des Platzhalters für die Geometriedaten hängt wesentlich von der Ausdehnung der einzelnen Geometriedaten ab. In diesem Projekt wird mit drei unterschiedlichen Geometriedatensätzen (Schweiz, Europa, Welt) gearbeitet, die alle eine eigene Ausdehnung besitzen. Die Schweizer- und die Weltkarte besitzen beide ein rechteckiges Format, Europa dagegen ein eher quadratisches Format. Die unterschiedlichen Formate müssen bei der Platzierung der

Karten berücksichtigt werden.

Die Legende und das eingebaute Diagramm beziehen sich auf die Geometriedaten und werden deshalb direkt unterhalb dieser Daten platziert. In diesen Bereich fallen auch die Informationen, die beim Überfahren der einzelnen Pfade angezeigt werden. Sie werden ebenfalls unterhalb der Geometriedaten aufgeführt.

Die Navigation wird in der rechten oberen Ecke neben der Hauptkarte platziert. Somit kann der direkte Bezug zu den Geometriedaten hergestellt werden.

Gleich darunter folgen die Einstellungen, die der Benutzer vornehmen kann. Die Einstellungen werden in eine Haupt- und zwei Untergruppen aufgeteilt. Die Reihenfolge der zu tätigenen Einstellungen kann der Benutzer frei wählen.

In der Hauptgruppe ist die Wahl des Kartenausschnittes, die Herkunft nach Wohnort vor Studienbeginn oder die Nationalität, das darzustellende Jahr und der darzustellende Status der Studierenden (Bestand oder Neueintritte der ETH) zu bestimmen.

In der ersten Untergruppe kann der Benutzer die Abfrage weiter spezifizieren. Es können die Studienkategorie, der Studiengang und das Geschlecht bestimmt werden.

In der zweiten Untergruppe werden die Einstellungen zur Klassierung vorgenommen. Die Klassierungsart (Quantile oder Gleiche Intervalle) und die Anzahl Klassen werden in diesem Bereich ausgewählt. Das Zusatzelement „Color-Picker“, mit dem der Benutzer die Darstellungsfarbe verändern kann, wird ebenfalls in die Gruppe der Klassierungsattribute gesetzt.

Rechts unten befindet sich ein Link zu den Quellen- und Impressumsangaben dieser Arbeit. Beim Anklicken dieser Schrift erscheint ein Feld mit den entsprechenden Angaben.

### **3.3 Wahl der Techniken**

Dieses Unterkapitel erläutert, welche Instrumente und Formate für die Umsetzung dieses Projektes verwendet werden und wie das Zusammenspiel der einzelnen Instrumente funktioniert. Die verschiedenen Instrumente selber sind hier nicht ausführlich erläutert. Für nähere Informationen ist das Kapitel „Verwendete Instrumente“ des Anhangs A zu konsultieren.

#### **3.3.1 Trennung von Präsentation und Daten**

Eine Grundvoraussetzung der Aufgabenstellung ist die Trennung der Daten von der Präsentation. Das heisst, dass alle Daten der ETH in der vorhandenen Datenbank verwaltet werden können und die für die Karte benötigten Informationen mittels Abfragen extrahiert werden. In

der Karte selber sollen diese Daten in geeigneter Form dargestellt werden.

### 3.3.2 Präsentation

Für die Visualisierung des Projektes im Internet wird SVG (Scalable Vector Graphics) verwendet. SVG ist eine Sprache zur Beschreibung und Integration von Vektorgrafiken (aber auch Rastergrafiken und Text) und ist XML-basiert. Damit der Browser den SVG-Code interpretieren und somit darstellen kann, wird ein Plugin benötigt, das auf dem Computer installiert werden muss.

SVG selber ist ein mehr oder weniger statisches Format, das aber mit Hilfe von JavaScript interaktiv gestaltet werden kann. Damit kann der Benutzer mittels Klicken, Überfahren, etc. eines vorgegebenen Objektes ein Event auslösen, der eine Veränderung der Karte (z.B. Anzeigen von Informationen) bewirkt.

Die Geometriedaten liegen als Shapefiles<sup>16</sup> vor. Für die Bearbeitung dieser Vektordaten wird ArcGIS verwendet. ArcGIS ermöglicht die Manipulation sowohl der Geometrie selber, als auch deren Attributtabelle. Die Shapefiles können mittels eines Programmes ins SVG-Format konvertiert werden. Dabei werden die Attribute übernommen.

### 3.3.3 Client-Server-Kommunikation

Die Daten der ETH liegen in einer Oracle-Datenbank und werden mittels PHP und SQL abgefragt. PHP steht für PHP-Hypertext Preprocessor und ist eine serverseitige Skriptsprache<sup>17</sup>. SQL steht für Structured Query Language und ist eine ANSI<sup>18</sup>-Standardcomputersprache für den Zugriff und die Manipulation von Datenbanksystemen.

PHP wird als Serverschnittstelle zwischen der Serversoftware und der Serverextension, welche ein GIS<sup>19</sup> oder eine Datenbank sein kann, verwendet. Wenn der Server<sup>20</sup> eine Anfrage eines Klienten erhält, wird der PHP-Code ausgeführt.

---

16 Siehe Glossar: Stichwort „Shapefile“

17 Siehe Glossar: Stichwort „Skriptsprache“

18 **American National Standards Institute**

19 Siehe Glossar: Stichwort „GIS“

20 Siehe Glossar: Stichwort „Server“

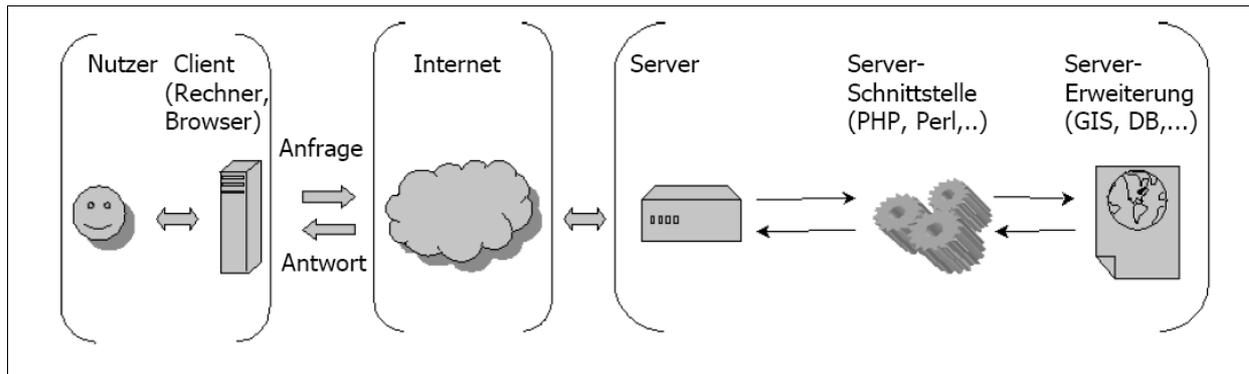


Abbildung 5: Client-Server Kommunikation

Der PHP-Code kann entweder im SVG-File eingebettet sein oder als eigene Datei bestehen. Falls der PHP-Code in das SVG eingebettet ist, filtert ein PHP-Interpreter auf dem Server den ganzen PHP-Code aus dem umgebenden SVG- oder HTML-Code heraus und führt die Anweisungen aus. Das Resultat wird in ein SVG- oder HTML-Code transformiert und zurück zum Klienten gesendet. Der Vorteil an diesem Vorgang ist, dass der User seinen Standard-Browser benutzen kann, ohne Plugins installieren zu müssen [Held et al., 2003].

Im vorliegenden Projekt wird eine eigene PHP-Datei erstellt. Der Ablauf, der im Hintergrund der Karte passiert, sieht wie folgt aus:

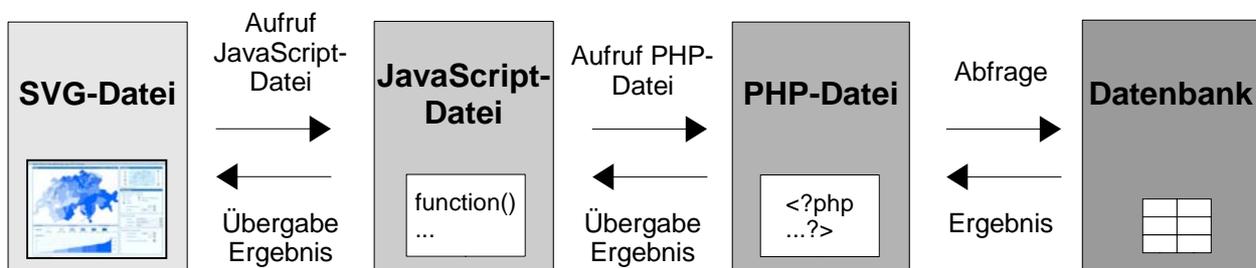


Abbildung 6: Ablaufprozess

Der Benutzer ruft bei sämtlichen Interaktionen die JavaScript-Datei auf. Wenn es sich bei der Interaktion um eine Änderung der Datenbankabfrage handelt, ruft die JavaScript-Datei die PHP-Datei auf, die wiederum die Abfrage an die Datenbank weiterleitet. Mittels SQL-Befehlen, die im PHP-Code eingebettet sind, werden die gewünschten Daten aus der Datenbank extrahiert und dem PHP übergeben. Die PHP-Datei verarbeitet die Ergebnisse und stellt sie der JavaScript-Datei zur Verfügung. In der JavaScript-Datei werden die Ergebnisse eingelesen und gespeichert. Hier werden die Daten so bearbeitet, dass sie schliesslich in der SVG-Karte dargestellt werden können.

Der Vorteil einer PHP-Datei, die nicht in die SVG-Datei integriert ist, besteht darin, dass die Ergebnisse in der JavaScript-Datei eingelesen und gespeichert werden können. Die von der Abfrage zurückgelieferten Resultate können so mehrmals für verschiedene Berechnungen benutzt werden. Im vorliegenden Projekt werden die Resultate u.a. für die Einfärbung der Karte und für die Berechnung der Intervallgrenzen verwendet.

## 4 Material und Methodik

Dieses Kapitel beschreibt das Material und die Methodik, die für die Ausführung des Projektes verwendet und angewendet wurden. Die Daten und die Programme, mit denen gearbeitet wurde, sind kurz beschrieben. Zudem wird der Arbeitsablauf erläutert, nach dem in diesem Projekt vorgegangen wurde.

### 4.1 Verwendete Daten

Für diese Diplomarbeit werden grundsätzlich zwei Datengruppen verwendet. Es sind einerseits Geometriedaten für die Schweizer- und die Weltkarte und andererseits Sachdaten, die in einer Oracle-Datenbank verwaltet werden. Nachfolgend werden die zwei Datentypen beschrieben und der Verwendungszweck erläutert.

#### 4.1.1 Daten in Oracle-Datenbank

Die ETH Zürich erfasst seit 1990 alle Angaben zur Herkunft der Studierenden in einer Oracle Datenbank. Dabei existiert eine Haupttabelle, in der alle Studierenden je mit mehr als 50 Attributen beschrieben werden. Weiter gibt es zwei relationale<sup>21</sup> Tabellen, die den Zuweisungscodes zu Gemeinden und Ländern enthalten. Die Haupttabelle der Studierenden enthält rund 150'000 Einträge. Eine effiziente Verwaltung dieser grossen Datenmenge kann nur in einer Datenbank erfolgen. Um die Datennachführung effizient zu gestalten, werden die Daten nicht in eigene Dateien exportiert, sondern in der Datenbank belassen. Mittels Abfragen wird auf die Datenbank zugegriffen und die gewünschten Daten herausgelesen.

Die Verwaltung der Oracle-Datenbank liegt in der Abteilung Controlling der ETH Zürich. Alle Anpassungen und Änderungen von Einträgen müssen demzufolge durch die Abteilung Controlling vorgenommen werden. Für diese Diplomarbeit wurden spezielle Ansichten erstellt, auf die die Autorin zugreifen kann und in denen die Abfragen ausgeführt werden. Diese Ansichten enthalten alle Tabellen, die für die Erstellung des Projektes benötigt werden. Die Autorin hat in diesen Ansichten nur ein Leserecht. Mutationen sind unmöglich.

---

<sup>21</sup> Relational: In Verbindung stehend mit etwas. Hier: die Gemeindetabelle steht in Verbindung mit der Studierendentabelle mittels eines Schlüssels (Attribut).

### 4.1.2 Geometriedaten

Unter den Geometriedaten versteht man Daten, die für den Raumbezug sorgen. Die Geometrie beschreibt Lage und Form der Objekte (hier: administrative Grenzen). Die Geometriedaten müssen für das Projekt in Form von Vektordaten vorliegen. Jedes Objekt wird also durch eine Reihe von Koordinatenpaaren, die seine Form definieren, dargestellt.

Shapefiles eignen sich für dieses Projekt besonders gut, da sie Vektordatensätze sind und bereits Werkzeuge existieren, die die Konvertierung von Shapefiles in SVG-Formate ermöglichen.

Im Folgenden wird näher auf die zur Verfügung stehenden Geometriedaten der Schweizer- und der Weltkarte eingegangen.

#### 4.1.2.1 Schweizerkarte

Für die Geometrie der Schweizerkarte steht der VECTOR200 Datensatz<sup>22</sup> vom Bundesamt für Landestopographie zur Verfügung.

„VECTOR200 ist ein kleinmasstäbliches 2D-Landschaftsmodell, welches inhaltlich und geometrisch auf der Landeskarte und der Strassenkarte 1:200'000 basiert. VECTOR200 gibt die natürlichen und künstlichen Objekte der Landschaft im flexiblen Vektorformat wieder und beschreibt rund 426'000 Objekte mit Lage, Form und ihren Nachbarschaftsbeziehungen (Topologie) sowie der Objektart und weiteren Sachattributen. Sein Perimeter umfasst die ganze Schweiz und das angrenzende Ausland mit einer totalen Fläche von rund 103'200 km<sup>2</sup> [swisstopo, 2004]“.

Die Objekte sind in sechs thematische Ebenen aufgeteilt, die als separate Shapefiles vorliegen. Da für diese Diplomarbeit nur die administrativen Grenzen relevant sind, wird nur die Ebene „Grenzen“, die 12'000 Objekten enthält, verwendet.

Die Attributtabelle dieser Ebene ist sehr ausführlich. Jedes Objekt ist durch 14 Attribute beschrieben. Darunter befinden sich auch die Bezirksnummer, die im Verlauf der Arbeit benötigt werden.

#### 4.1.2.2 Weltkarte

Auch für die Weltkarte steht ein ESRI<sup>23</sup>-Datensatz zur Verfügung. Es handelt sich um den ESRI-Weltdatensatz. In diesem sind die Grenzen aller Länder aufgeführt. Die Grenzverläufe sind sehr detailliert aufgezeichnet. Dadurch werden die Dateien sehr gross.

---

<sup>22</sup> Nachführungsstand: 1.1.2003

<sup>23</sup> Softwarefirma für Geoinformationssysteme

Die Attributtabelle ist in dieser Datei nicht sehr ausführlich. Sie enthält lediglich die Namen aller Länder in englischer Sprache. Auffallend ist, dass alle abhängigen Gebiete (z.B. Faeroer Inseln, die zu Dänemark gehören) als eigene Staaten mit einem eigenen Landnamen aufgeführt sind.

### 4.1.2.3 Europakarte

Für die Europakarte wird kein neuer Datensatz verwendet. Die zu Europa gehörenden Länder werden vom Weltkarten-Datensatz übernommen.

## 4.2 Verwendete Programme

Dieser Abschnitt enthält eine Tabelle, die die verwendeten Programme und deren Verwendungszweck auflistet. Anschliessend wird kurz auf die einzelnen Komponenten eingegangen. Detailliertere Informationen über die Programme sind dem Anhang A zu entnehmen.

Programm	Programmtyp	Verwendungszweck	Status
ArcGIS	GIS-Programm	Datenaufbereitung	Lizenzpflichtige Software
Shp2svg	Konverter	Konvertierung der Shapefiles in SVG-Dateien	Open Source
Ultra Edit	TextEditor	Erstellen und Ändern der Codes von SVG, JavaScript und PHP	Lizenzpflichtige Software
Apache <sup>24</sup>	Webserver <sup>25</sup>	Server	Open Source
Oracle-Client	Werkzeug für Oracle	Datenbankabfragen in Oracle-Datenbank	Open Source
TOra <sup>26</sup>	Werkzeug für Oracle	Prüfen der SQL-Befehle	Open Source

Tabelle 1: Verwendete Programme

ArcGIS ist ein GIS-Programm, mit dem die Datenaufbereitung durchgeführt wurde. Für die Konvertierung der Shapefiles in SVG-Dateien wurde der Konverter Shp2svg verwendet (mehr dazu ist dem Kapitel 5.2 „Konvertieren der Shapefiles in SVG-Dateien“ zu entnehmen).

<sup>24</sup> [URL: Apache]; siehe Glossar: Stichwort „Apache-Server“

<sup>25</sup> Siehe Glossar: Stichwort „Webserver“

<sup>26</sup> [URL: TOra]

Ultra Edit ist ein Texteditierprogramm. In ihm werden alle Anpassungen der Codes von SVG, JavaScript und PHP vorgenommen.

Eine Besonderheit stellt die Verwendung des Webservers Apache dar. Er bietet die Möglichkeit, mittels serverseitiger Skriptsprachen (hier PHP) Webseiten dynamisch zu gestalten.

Der Apache-Server muss auf dem Computer installiert werden. Er ist modular aufgebaut. Durch entsprechende Module können beispielsweise Datenbankabfragen ausgeführt werden. Der Apache-Server bietet wie bereits erwähnt die Möglichkeit, mittels serverseitigen Skriptsprachen (hier PHP) Webseiten dynamisch zu gestalten. Diese Skriptsprachen sind keine Bestandteile des Webservers, sondern müssen entweder durch Module eingebunden werden oder über die CGI<sup>27</sup>-Schnittstelle angesprochen werden. Im Falle von PHP muss das Modul „php5\_module“ (PHP-Version 5) eingebunden werden, falls dieses noch nicht eingebunden ist.

Damit Datenbankabfragen auf eine Oracle-Datenbank ausgeführt werden können, muss die entsprechende Erweiterung (extension=php\_oci8.dll) in der php.ini-Datei freigeschaltet werden. Dies geschieht durch das Löschen des Strichpunktes vor dem Namen der Erweiterung. Das Freischalten dieser Erweiterung erlaubt die Verwendung von Oracle-Funktionen<sup>28</sup> (oci-Funktionen) in den PHP-Dateien.

Zusätzlich muss ein Oracle-Client<sup>29</sup> auf dem Computer installiert werden, der als Glied zwischen der PHP-Datei und der Datenbank agiert. Der Oracle-Client sorgt dafür, dass die Datenbank die von der PHP-Datei gesendeten SQL-Befehle richtig interpretiert und somit die richtigen Resultate zurückliefert.

Tora<sup>30</sup> ist ein Werkzeug, das die Verbindung zur Datenbank herstellen kann. Abfragebefehle können so geprüft werden. Die Resultate werden direkt im Programm-Fenster angezeigt.

---

27 **Common Gateway Interface:** Allgemeine Vermittlungs-Schnittstelle

28 Weitere Informationen zu Oracle-Funktionen sind im Referenzbuch zu PHP zu finden. [URL: PHP]

29 [URL: Oracle]

30 [URL: Tora]

### 4.3 Arbeitsablauf

Folgende Schritte sind von der Datenaufbereitung bis zum Funktionieren der Karte nötig:

- **Datenaufbereitung**

Zuerst müssen alle Datensätze dem Projekt entsprechend angepasst und aufbereitet werden. Es sind dies die Geometriedatensätze und diejenigen der Datenbank.

- **Konvertierung der Shapefiles in SVG-Dateien**

Die Shapefiles müssen in SVG-Dateien konvertiert werden, damit sie später in die Haupt-SVG-Datei implementiert werden können.

- **Erstellen der Haupt-SVG-Datei**

Es wird eine SVG-Datei erstellt, in die später die Bestandteile der Karte implementiert werden. Das Layout der Karte, das im Konzept definiert wurde, muss in diesem Schritt umgesetzt werden.

- **Implementierung der Kartenbestandteile**

Die Kartenbestandteile „Geometrie“, „Navigation“, „Selektionslisten“ und „Color-Picker“ werden in die SVG-Datei integriert. Die Grösse und Position der Bestandteile müssen dem Kartenlayout angepasst werden.

- **Implementierung der Interaktionen**

Schliesslich kann mit der eigentlichen Erstellung der Interaktionen begonnen werden. Zu den Interaktionen gehören die Einfärbung der Karte, das Erstellen der Legende und des Diagramms und das Anzeigen von Zusatzinformationen. Bevor aber mit der Einfärbung der Karte begonnen werden kann, muss die Datenbankabfrage ausgeführt werden, die die Resultate zurückliefert und mit denen weiter gearbeitet wird.

Die Einzelheiten und die konkrete Durchführung der aufgelisteten Schritte sind im nachfolgenden Kapitel „Umsetzung“ nachzulesen.

## 5 Umsetzung

In diesem Kapitel wird die Umsetzung des Projektes, von der Datenaufbereitung bis hin zur Realisierung der SVG-Karte, beschrieben. Anhand der verschiedenen Codes wird erläutert, wie die einzelnen Interaktionen funktionieren und was bei deren Einbindung beachtet werden muss.

Das Projekt besteht aus einer SVG-Datei und mehreren Javascript-Dateien, die für das richtige Funktionieren der Karte verantwortlich sind und die in der SVG-Datei eingebunden werden müssen. Weiter existieren zwei PHP-Skripte, die für die Datenbankabfrage zuständig sind. In den folgenden Abschnitten wird näher auf diese Skripte eingegangen.

### 5.1 Datenaufbereitung

Die Datenaufbereitung muss sowohl für die Geometriedaten als auch für die Datenbank durchgeführt werden.

#### 5.1.1 Aufbereitung der Geometriedaten

Wie bereits erwähnt wurde, liegt die Geometrie der Schweizer- und der Weltkarte in Shapefiles vor. Die Aufbereitung der Geometriedaten erfolgt durch deren Generalisierung oder durch eine Aggregation<sup>31</sup> von Daten. Zusätzlich müssen die Attributtabelle beider Geometriedatensätze auf die Ansprüche der zukünftigen Interaktionen der SVG-Karte angepasst werden.

Die Generalisierung oder Aggregation der Daten wird aus folgenden Gründen durchgeführt:

- Der Detaillierungsgrad der Linienführung der Geometriedaten muss dem Massstabsbereich und dem Kartenzweck des Projektes angepasst werden. Da in diesem Projekt die Geometrie die Darstellungsgrundlage für die Visualisierung der Herkunft der Studierenden liefert, ist keine detaillierte Linienführung der Geometriedaten notwendig. Weil die Geometriedaten für dieses Projekt einen zu hohen Detaillierungsgrad aufweisen, muss eine Generalisierung durchgeführt werden.
- Für die Integration der Daten in die SVG-Karte müssen die Shapefiles in SVG-Dateien konvertiert werden. Zu beachten ist dabei die Grösse (in KB) der entstehenden SVG-Dateien. Je grösser die SVG-Datei, desto länger hat der Browser, um die Karte zu laden. Die Ladezeit sollte möglichst klein gehalten werden, da die Geduld des Benutzers nach einer gewissen Wartezeit überstrapaziert ist und er das Laden der Karte abbricht.

Um die Dateigrösse zu testen, wurden die Originalshapefiles der Schweizer- und der Weltkarte in eine SVG-Datei konvertiert. Dabei stellte sich bei beiden Dateien heraus,

---

<sup>31</sup> Unter Aggregation von Daten versteht man das Zusammenfassen detaillierter Daten zu grösseren Einheiten.

dass die akzeptable Dateigrösse bei weitem überschritten ist. Die Dateigrösse muss also auf geeignete Weise verringert werden. Dies kann mit einer Generalisierung oder Aggregation der Geometriedatensätze erreicht werden.

Die Datenaufbereitung der Schweizer- und der Weltkarte wird nach unterschiedlichen Vorgehensweisen durchgeführt. Der Datensatz der Weltkarte wird auf Grund der detaillierten Linienführung der Ländergrenzen generalisiert. Im Datensatz der Schweizerkarte hingegen wird lediglich eine Aggregation der Gemeindegrenzen zu Bezirksgrenzen durchgeführt.

#### 5.1.1.1 Generalisierung der Weltkarte

„Die Generalisierung besteht hauptsächlich aus der zweckentsprechenden Auswahl und Zusammenfassung der Objekte sowie der möglichst lagegenauen, charakteristischen, richtigen und eindeutigen grafischen Darstellung“ [SGK, 2002]

Im vorliegenden Projekt erfolgte die Generalisierung hauptsächlich durch das Löschen von Zwischenpunkten eines Pfades. Dabei ist es wichtig, die ursprüngliche Form des Objektes beizubehalten.

Eine Generalisierung von Datensätzen hat immer auch eine Verringerung der Dateigrösse zur Folge, was für dieses Projekt ein guter Nebeneffekt ist.

In der Weltkarte sind über 200 Länder mit ihren Grenzen eingezeichnet. Der Verlauf der Grenzen ist sehr detailliert. Dies hat zur Folge, dass die Grenze eines Landes mit bis zu mehreren tausend Punkten beschrieben ist. Dazu kommt, dass die Grenzen zwischen zwei nebeneinanderliegenden Ländern doppelt aufgeführt sind. Das heisst, dass beispielsweise die Grenzpunkte zwischen der Schweiz und Deutschland einmal für Deutschland und einmal für die Schweiz aufgelistet sind. Diese Doppelführung der Koordinatenpunkte vergrössert die SVG-Datei zusätzlich. Darum ist eine Generalisierung der Weltkarte unausweichlich. Diese kann auf zwei Arten erfolgen:

- Automatische Generalisierung in ArcGIS
- Generalisierung von Hand in ArcGIS

Beim Testen der automatischen Generalisierung stellten sich zwei Negativpunkte heraus, die zur Verwerfung dieser Methode führten. Zum einen entstanden durch die Zweifachführung der Grenzlinien zweier nebeneinander liegender Länder bei der Generalisierung Sliver Polygons<sup>32</sup>, weil jede Grenzlinie einzeln generalisiert wird. Zum anderen bereitete die Eliminierung der

---

<sup>32</sup> Sliver Polygon: Eine kleine Fläche (Splitterpolygon), die geformt wird, wenn zwei oder mehr Polygone, nicht exakt aneinandergrenzen [geolexikon 2005].

Sliverpolygons Probleme.

Die Generalisierung musste somit von Hand durchgeführt werden. Weil der Aufwand für die Generalisierung aller Pfade zu aufwändig wäre und im Rahmen dieser Diplomarbeit nicht durchführbar ist, wurden nur die Küstenlinien generalisiert.

Im Weiteren wurden bei Inselstaaten, die mehrere Inseln beinhalten (z.B. die Malediven), die kleinen Inseln gelöscht und nur zwei bis drei Hauptinseln beibehalten. Die Ausdehnung der Inselgruppe soll jedoch weiterhin ersichtlich sein.

Kleine Inseln, die selber keine Staaten verkörpern, sondern Bestandteile eines grösseren Staates sind, wurden ebenfalls aus der Geometrie gelöscht. Ein Beispiel dafür sind die Kanalinseln, die zwischen Frankreich und England liegen und politisch zu Grossbritannien gehören.

#### **5.1.1.2 Aggregation der Schweizerkarte**

Das verwendete VECTOR200 Shapefile „Grenzen“ enthält die Pfade der Gemeindegrenzen. Da für die vorliegende Arbeit aus Gründen der Darstellung und der Dateigrösse die Bezirksebene gewählt wurde, mussten die Gemeindegrenzen zu Bezirken zusammengefasst werden. Dies kann mit dem ArcToolbox Werkzeug „Dissolve“ des Programms ArcGIS gelöst werden. Dieses Werkzeug fasst Objekte mit denselben Attributwerten zu neuen Objekten zusammen. Der Benutzer kann das Attribut, auf welches sich die Neugliederung der Geometrie bezieht, angeben. In unserem Beispiel werden nun die Gemeinden anhand der Bezirksnummer zu Bezirken zusammengefasst.

#### **5.1.1.3 Erweitern der Attributtabelle**

Den vorhandenen Attributtabelle der Geometrien fehlen Attribute, die für die SVG-Karte nötig sind. Dies kann ein eindeutiger Identifikationsschlüssel wie im Falle der Weltkarte oder Attribute wie Bezirksnamen im Falle der Schweizerkarte sein.

##### *Weltkarte*

Die Attributtabelle der Weltkarte enthält nur die englischen Namen der Länder. Um eine eindeutige Identifizierung der Objekte (hier der Länder) zu erhalten, wird zusätzlich ein Code für die Ländernamen eingeführt. Dies ist nötig, weil je nach verwendeter Sprache die Orthographie der Ländernamen variieren kann und somit eine eindeutige Identifizierung fehlt. Es gibt zwei unterschiedliche Codierungen für den Identifikationsschlüssel. Einerseits der ISO<sup>33</sup>-Ländercode, der eine Codierung in Form von zwei Buchstaben aufweist, und der Staatscode des Bundes-

---

<sup>33</sup> International Standardization Organisation: 1947 gegründete Organisation, die die Normung weltweit koordiniert.

amtes für Statistik Schweiz, der eine Codierung von vier aufeinanderfolgenden Zahlen aufweist.

- **ISO-Code:**

Der ISO-Code ist von der Norm ISO 3166, die die Ländercodierung regelt, abgeleitet und wird international verwendet. Die ISO-Codierung umfasst 240 Länder. Der Nachteil der ISO-Codierung ist, dass Gebiete, die kein eigenständiger Staat sind, sondern politisch von einem anderen Staat abhängig sind, mit einem separaten Code aufgelistet sind. Dies führt zu Problemen, da nicht für alle ISO-codierten Gebiete Einwohnerzahlen, welche für diese Diplomarbeit von Notwendigkeit sind, auszumachen sind.

- **Staatscode des Bundesamtes für Statistik Schweiz:**

Der Staatscode entstammt dem Staaten- und Gebietsschlüssel des Bundesamtes für Statistik. Dieser Code sieht für jeden anerkannten Staat eine vierstellige Nummer vor. Politisch abhängige Gebiete werden mit demselben Staatscode wie das Mutterland versehen. So wird zum Beispiel Grönland mit demselben Staatscode wie Dänemark versehen, da die Insel ein von Dänemark abhängiges Gebiet ist.

In dieser Arbeit wird der Staatscode als Identifikationsschlüssel übernommen und die Attributtabelle um diesen erweitert.

#### *Schweizerkarte*

Die Attributtabelle der Schweizerkarte besitzt bereits die Bezirksnummer als eindeutigen Identifikator. Die Tabelle der Schweizerkarte wird um das Attribut „Bezirksname“ erweitert, da der Bezirksname zu einem späteren Zeitpunkt in der SVG-Karte angezeigt werden soll.

### **5.1.2 Aufbereitung der Datenbank**

Die Aufbereitung der Oracledatenbank wurde von Michael Fuchs von der Abteilung Controlling durchgeführt.

Zusätzlich zu den bereits vorhandenen Daten der Datenbank wurden zwei Textdateien erstellt, die anschliessend in die Datenbank eingelesen wurden. Die Textdateien enthalten die Bevölkerungszahlen für jeden Bezirk und jedes Land im Zeitraum von 1990 - 2004.

Die Angaben über die Bevölkerungszahlen eines Landes oder Bezirkes werden für die Berechnung der darzustellenden Relativwerte benötigt.

Im Weiteren werden die Inhalte der beiden Textdateien erläutert:

### *Weltbevölkerungszahlen getrennt nach Ländern*

In dieser Textdatei sind alle Länder mit der zugehörigen Staatsnummer und deren Bevölkerungszahlen im Zeitraum von 1990-2004 aufgelistet. Die Bevölkerungszahlen wurden der Internetseite des „U.S Census Bureau“ entnommen. Diese Seite führt eine Computerdatenbank mit demografischen und sozio-ökonomischen Daten von 227 Ländern im Zeitraum zwischen 1950 und 2050. Der Benutzer kann eine Abfrage mittels Auswahlmenüs zusammenstellen. Die Resultate werden im Browser selbst angezeigt. [URL: U.S.C.B.]

Mit dem „Kopieren und Einfügen“ Verfahren werden die Werte in eine Exceltabelle übertragen und zusammengestellt.

Da die Einträge der U.S. Census Computerdatenbank nicht exakt der Staatscodierung des Bundesamtes für Statistik entspricht, muss die Excel-Tabelle noch entsprechend angepasst werden.

### *Bevölkerungszahlen getrennt nach Bezirken*

Das Bundesamt für Statistik führt Excel-Tabellen mit den Bevölkerungszahlen der Schweiz, die auf dem Internet<sup>34</sup> frei zur Verfügung stehen. Die Bevölkerungszahlen sind auf allen administrativen Ebenen erhältlich, das heisst auf Gemeinde-, Bezirks- und Kantonebene. Die offiziellen Zahlen des Jahres 2004 sind bis zum Abschluss dieser Arbeit nicht erhältlich. Deshalb wurden für das Jahr 2004 provisorisch die Zahlen des Jahres 2003 übertragen.

Zu beachten ist, dass sich die für dieses Projekt verwendete Geometrie auf das Jahr 2002 bezieht. Das heisst, dass die Bezirksgrenzen den Nachführungsstand vom 1.1.2003 besitzen. Seit diesem Datum wurde im Kanton St.Gallen eine Bezirksmutation durchgeführt. Die ursprünglich 14 Bezirke wurden neu zu 8 Bezirken zusammengefasst. Die Bevölkerungszahlen dieser Bezirke wurden der Geometrie angepasst. Demzufolge ist die Bevölkerungszahl des Kantons St.Gallen auch im Jahre 2004 immer noch auf 14 Bezirke verteilt.

## **5.2 Konvertieren der Shapefiles in SVG-Dateien**

Es existieren verschiedene Möglichkeiten Shapefiles in SVG-Dateien zu konvertieren. In dieser Arbeit wurden hierfür zwei Instrumente getestet. Einerseits shp2svg von Nedjo Rogers und andererseits Ogis2svg von Andreas Neumann (Institut für Kartographie der ETH Zürich)

- **shp2svg**: Dies ist ein Skript, das in ArcView von ESRI gestartet werden kann und exportiert den Inhalt einer Ansicht in ArcView in ein SVG-Format. Es kann in einem Text Editor öff-

---

34 [URL: BfS]

net werden und so den Ansprüchen des Benutzers angepasst werden. Olaf Schnabel, Doktorand am Institut für Kartographie, hat eine Anpassung des Skriptes auf diese Diplomarbeit vorgenommen. Das Resultat des ursprünglichen Konvertierungsskriptes ist eine in eine HTML-Seite eingebundene SVG Karte inklusive einer Legende und einigen einfachen Interaktionsmöglichkeiten. Die Anpassung des Skriptes hat zur Folge, dass die Georeferenzierung der Daten beibehalten und kein HTML File generiert wird. Dazu kommt, dass die Anzahl der Nachkommastellen in einer Auswahl angegeben werden kann. Das Resultat des abgeänderten Skripts ist eine einfache SVG-Datei mit den Pfaden der Objekte und der ursprünglichen Koordinatenangaben.

Dieses Skript steht im Internet als „Download“<sup>35</sup> zur Verfügung. Eine Gebrauchsanweisung für die Benutzung des Skriptes ist ebenfalls auf der angegebenen Internetseite erhältlich.

- **Ogis2svg:** Die Shapefile-Datei wird in einem ersten Schritt in eine SQL-Datei konvertiert. Hier werden alle Informationen der Shapefile-Datei übernommen. Diese SQL-Datei wird in einem weiteren Schritt mittels des Perl-Skriptes Ogis2svg.pl in eine SVG-Datei umgewandelt. Dabei wird dem Benutzer ein Auswahlmeneü präsentiert, in dem er angeben kann, welche Attribute in die SVG-Datei integriert werden sollen. Zudem ist ein Attribut zu bestimmen, das als ID übernommen wird. Mit dieser Konvertierungsmethode wird die Georeferenzierung beibehalten und die Nachkommastellen können angegeben werden.

Der „Shape to SVG-Konverter“ steht im Internet als „Download“<sup>36</sup> zur Verfügung. Eine ausführliche Anweisung zur Benutzung und Installation des Werkzeuges ist ebenfalls auf dem Internet zu finden.

In der vorliegenden Arbeit wurde das Instrument Ogis2svg verwendet. Sein Vorteil ist, dass die Koordinaten der Pfade relativ zueinander angegeben werden. Das heisst, dass bei einem Pfad nur die erste Koordinate in ihrer vollen Länge angegeben wird. Alle weiteren Koordinaten werden als Differenz zur ersten Koordinate angegeben.

Mit der Verwendung von Relativpfaden kann die Filegrösse klein gehalten werden, was ein grosser Vorteil ist. Ein weiterer Vorteil des Ogis2svg ist, dass die Attribute als Namespaces<sup>37</sup> (hier: „attrib:“) übernommen werden.

Folgende Einstellungen werden bei der Konvertierung mit Ogis2svg vorgenommen:

---

35 [URL: shp2svg]

36 [URL: Ogis2svg]

37 Namespace (Namensraum): Bereich für Namen oder Zone mit bestimmtem Inhalt. Jeder Namespace hat dafür sein eigenes Präfix, gefolgt von einem Doppelpunkt.

- **Weltkarte:** 3 Nachkommastellen; Attribute, die in die SVG-Datei übernommen werden: Staatscode (state\_code) und Landname (placename), Staatscode wird als ID für die SVG-Objekte übernommen.
- **Schweizerkarte:** 1 Nachkommastelle; Attribute, die in die SVG-Datei übernommen werden: Bezirksnummer (bezirksnr) und Bezirksname (bezname); Bezirksnummer wird als ID für die SVG-Objekte übernommen.

Der erhaltene SVG-Code der Pfade kann in diesem Zustand in die Haupt-SVG-Datei übernommen werden.

### 5.3 Erstellen der Haupt-SVG-Datei

Es wird eine SVG-Datei erstellt, in die dann später alle Kartenbestandteile integriert werden. In dieser SVG-Datei wird das Layout der Karte, das im Konzept erstellt wurde, umgesetzt. Rechtecke, in die dann später die einzelnen Elemente der Karte implementiert werden, werden hier gezeichnet und definiert. Auch die verschiedenen Titel und Textelemente, die die Karte aufweist, werden bereits in diesem Abschnitt gesetzt.

Die Attribute (Füll-, Strichfarbe, Schriftart, Schriftgrösse, etc.) der Zeichen- und Textelemente werden in einem externen CSS-Stylesheet<sup>38</sup> definiert. Das Aussehen der Objekte der SVG-Datei wird durch das Stylesheet global bestimmt. Den Objekten selber kann mit dem Befehl class="" ein Stil zugewiesen werden, indem so auf das CSS-Stylesheet referenziert wird:

```
SVG-Code:
  <style type="text/css"><![CDATA[ //Beginn des Stylesheet-Abschnittes

  .bezirke{fill:rgb(255,255,255);fill-rule:evenodd;stroke:rgb(0,0,0);
    stroke-width:0.1px;}

  ...
  ]]></style> //Ende des Stylesheet-Abschnittes

<g id="bezirke" class="bezirke" />
```

Im vorliegenden Projekt wird das Aussehen der Rechtecke, der Titel, der Untertitel der Textelemente, aber auch das Aussehen der Geometriedaten und des Diagramms global definiert.

In der SVG-Datei können in einer Definitionssektion am Anfang der Datei Symbole definiert werden, auf die dann in der gesamten SVG-Datei zugegriffen werden kann. Dies hat den Vorteil, dass bei Symbolen, die mehrmals in der Karte vorkommen, nicht jedesmal deren Pfade

---

<sup>38</sup> Siehe Glossar: Stichwort „CSS“

aufgeführt werden muss. Es kann einfach auf das Symbol in der Definitionssektion verwiesen werden. Im vorliegenden Projekt werden alle Symbole definiert, die in der Karte benötigt werden.

## 5.4 Implementierung der Kartenbestandteile

In diesem Unterkapitel werden die einzelnen Interaktionen und deren Funktionen aufgelistet. Zudem werden die Schritte und wichtigen Punkte erläutert, die es benötigt, damit die Interaktionen einwandfrei funktionieren.

### 5.4.1 Geometrie

Die Implementierung der Geometriedaten geschieht auf folgende Weise: Der SVG-Code der einzelnen Pfade muss an der richtigen Stelle in die SVG-Karte eingefügt werden. Da SVG textbasiert ist, bleiben die Geometriedaten überschaubar. Jeder Pfad weist ein Attribut mit dem zugehörigen Länder- oder Bezirksnamen auf.

SVG-Code:

```
<path id="bez_101" attrib:bezname="Affoltern am Albis" d="M678670 -2453971532
2201838 802144.2 119.11-87.2 111.1131.7 1351150.8 317.51166.7
222.2123.8 ..... 119.11-71.4 -157z" />
```

In der Schweizerkarte wird zusätzlich zu den Bezirks- auch noch die Kantons Grenzen und die Seen eingefügt. Der Benutzer kann sich anhand dieser Elemente besser in der Karte orientieren.

### 5.4.2 Navigation



Abbildung 7: Navigation

Für die Navigation wird ein Skript verwendet, das Andreas Neumann vom Institut für Karto-

graphie der ETH Zürich im Internet als „Download“<sup>39</sup> zur Verfügung stellt. Dabei müssen die Skripte `mapApp.js`, `helper_functions.js`, `navigation.js` und `slider.js` in die SVG-Datei eingebunden werden. Die Navigationshilfe stellt eine kleine Übersichtskarte (Referenzkarte) zur Verfügung, in der analog zur grossen Karte navigiert werden kann. Folgende Funktionalitäten stehen zur Verfügung:

- Ein- und Auszoomen bis zu einem maximalen bzw. minimalen Wert
- Aufziehen eines Zoomrechteckes in der Hauptkarte (Drag und Drop)
- Umherschieben des Zoomrechteckes in Haupt- und Übersichtskarte
- Anzeigen der Vollansicht
- Setzen von neuem Mittelpunkt des Zoomrechteckes (durch Klicken in Haupt- oder Übersichtskarte)
- Anzeigen der Koordinaten sowohl in der Haupt- als auch in der Referenzkarte

Bei der Implementierung der Navigationshilfe muss darauf geachtet werden, dass das Verhältnis „Länge zu Breite“ der Hauptkarte gleich dem Verhältnis „Länge zu Breite“ der Referenzkarte ist. Die Ausdehnung der beiden Bildausschnitte (Haupt- und Referenzkarte) muss ebenfalls gleich sein, weil ansonsten die Koordinatenanzeige nicht mehr stimmt.

Da das Navigationstool noch nicht für das Switchen zwischen unterschiedlichen Kartenausschnitten vorgesehen war, musste es noch erweitert werden:

Das Navigationstool kreiert im Hintergrund ein Rechteck mit der Ausdehnung der Hauptkarte. Dieses Rechteck muss bei jedem Wechsel zwischen den Kartenausschnitten gelöscht werden, damit die neuen Koordinaten übernommen werden. Ebenfalls muss der ZoomSlider gelöscht werden, da sonst mehrere ZoomSlider zu sehen sind.

Die Anweisungen zur Implementierung des Navigationstools sind auf der Internetseite<sup>40</sup> nachzulesen.

### 5.4.3 Selektionslisten

Für die Selektionslisten wurde das Selektionslistentool<sup>41</sup> von Andreas Neumann, Institut für Kartographie der ETH Zürich, verwendet. Die Implementierung der Selektionslisten geschieht auf folgende Weise: Im SVG-Code muss für jede Selektionsliste eine Gruppe mit einer eindeutigen Identifizierung (ID) erstellt und die Einbindung der JavaScript-Datei „`selectionList.js`“

---

39 [URL: Navigation]

40 [URL: Navigation]

41 [URL: Selektionsliste]

vorgenommen werden. Die Selektionslisten selber werden in der Haupt-JavaScript-Datei „main.js“ definiert. Hier werden Position, Grösse und Inhalt festgelegt. Im selectionList-Skript können zudem die Farben der Listen angepasst werden.

Die Anweisungen zur Implementierung des Selektionslistenstools sind im Internet<sup>42</sup> nachzulesen.

## 5.5 Implementierung der Interaktionen

In diesem Unterkapitel wird auf die Implementierung der Interaktionen eingegangen. Dazu gehören das Wechseln des Kartenausschnittes und die Wahl der Herkunft, aber auch die einzelnen Selektionslisten. Ebenfalls wird auf den Ablauf eingegangen, der bei einer Datenbankabfrage im Hintergrund der Karte geschieht. Die einzelnen Schritte, die von der Benutzereinstellung in der Karte über die Datenbankabfrage bis hin zur Darstellung der Resultate in der Karte reichen, werden in diesem Unterkapitel beschrieben.

Es wird mit verschiedenen Dateien gearbeitet, deren Zusammenwirken für das einwandfreie Funktionieren des Projektes unerlässlich ist. Damit der Leser die Übersicht über die einzelnen Dateien nicht verliert, werden diese anschliessend in einer Tabelle aufgelistet. Dabei wird der Dateityp, der Dateiname und die Funktion der Datei angegeben.

Dateiname	Dateityp	Funktion
index.svg	SVG	Layout und Geometrie
main.js	JavaScript	Haupt-JavaScript-Datei
navigation.js	JavaScript	Navigation
selectionList.js	JavaScript	Selektionslisten
colourpicker.js	JavaScript	Color-Picker
mapApp.js	JavaScript	Navigation, Selektionslisten, Color-Picker
helper_functions.js	JavaScript	Navigation, Selektionslisten, Color-Picker
slider.js	JavaScript	Navigation, Color-Picker
mouse_over_effects.js	JavaScript	Anzeigen von Zusatzinformationen
oraclepop.php	PHP	Datenbankabfrage Bevölkerungszahl
oraclestud.php	PHP	Datenbankabfrage Studierendenzahl

Tabelle 2: Auflistung der Dateien

42 [URL: Selektionsliste]

Nachfolgend wird auf diese Dateien Bezug genommen, so dass der Benutzer die einzelnen Schritte besser verstehen kann.

### 5.5.1 Switchen zwischen einzelnen Kartenausschnitten

Der Benutzer kann wählen, ob er die Herkunft der Studierenden der Schweiz, von Europa oder der Welt visualisieren will. Hierfür kann er unter drei Kartenausschnitten auswählen. Da nur jeweils eine Karte angezeigt wird, steht eine Switch-Schaltfläche zur Verfügung. Diese ist immer nur bei dem Kartenausschnitt sichtbar, der gerade ausgewählt ist.



Abbildung 8:  
Kartenausschnitt

Im SVG-Code der Datei „index.svg“ wird über den schwarzen Punkt (siehe Abb. 8) ein Event gelegt, der beim Klicken ausgelöst wird.

SVG-Code:

```
<use xlink:href="#switchCircle" id="ChCircle" visibility="visible"
onclick="handleToggleButton(evt, 'Switzerland');" x="720" y="310"/>
```

In der JavaScript-Datei „main.js“ ist die Funktion „handleToggleButton“ definiert und wird ausgeführt, sobald ein Benutzer den Event auslöst.

Die Steuerung der Schaltfläche geschieht mit der Sichtbarkeit (*visibility*) des schwarzen Punktes. Im SVG-Code kann ein Objekt mit dem Attribut *visibility* entweder als sichtbar (*visibility="visible"*) oder als nicht sichtbar (*visibility="hidden"*) gesetzt werden. Im Javascript-Code kann mittels *.getAttributeNS* (*null*, *"visibility"*) die Sichtbarkeit eines Objektes abgefragt und mittels *.setAttributeNS* (*null*, *"visibility"*, *"..."*) gesteuert werden. Der Javascript-Code für die Switch-Schaltfläche sieht dann folgendermassen aus, wobei in diesem Fall die Schweizerkarte angezeigt wird:

Javascript-Code:

```
worldpoint.setAttributeNS (null, "visibility", "hidden");
europoint.setAttributeNS (null, "visibility", "hidden");
chpoint.setAttributeNS (null, "visibility", "visible");
```

Für das Anzeigen der ausgewählten Pfade (Schweiz, Europa, Welt) wird ebenfalls mit der Sichtbarkeit gearbeitet. Die Pfade der Bezirke liegen in einer Gruppe mit der id="CHMap", diejenigen der Weltkarte liegen in der id="WorldMap". Wenn die Sichtbarkeit der ganzen Gruppe eingeschaltet ist, werden alle Pfade angezeigt.

Für das einwandfreie Funktionieren des Navigationstools müssen bei einem Wechsel des Kartenausschnittes einige Attribute geändert und angepasst werden. Es sind dies Attribute der Hauptkarte (MainMap), der Referenzkarte (ReferenceMap) und dem hellblauen Zoomrechteck der Referenzkarte. Dabei handelt es sich vorwiegend um Rechtecks- oder Bildausschnitt-Attribute wie Positionierung, Breite, Höhe und Ausdehnung:

Javascript-Code:

```
ReferenceMapLayer.setAttributeNS(null, "x", "760");
ReferenceMapLayer.setAttributeNS(null, "y", "75");
ReferenceMapLayer.setAttributeNS(null, "viewBox", "-180.0 -83.6 360.0
173.6");
ReferenceMapLayer.setAttributeNS(null, "width", "205.39");
ReferenceMapLayer.setAttributeNS(null, "height", "99.04");
```

### 5.5.2 Switchen zwischen Wohnort vor Studienbeginn und Nationalität

Beim Wechsel zwischen dem „Wohnort vor Studienbeginn“ und der „Nationalität“ wird lediglich mit dem Ein- und Ausschalten der Sichtbarkeit von Objekten gearbeitet.

Da das Anzeigen der Nationalität der Studierenden in der Schweizerkarte keinen Sinn macht, wird mit zwei unterschiedlichen Funktionen gearbeitet. Zum einen wird die Switch-Schaltfläche der Herkunft wie der des Kartenausschnittes mit der Sichtbarkeit gesteuert. Dies geschieht mit der Funktion „selectOrigin“, die in der JavaScript-Datei „main.js“ definiert ist.

Zum anderen wird bei der Einstellung „Nationalität“ die Schweizerkarte ausgeblendet.

Kartenausschnitt:	Herkunft:
<input type="radio"/> Welt	<input type="radio"/> Wohnort vor Studienbeginn
<input checked="" type="radio"/> Europa	<input checked="" type="radio"/> Nationalität

Abbildung 9: Herkunft: Nationalität

Wenn auf die Einstellung „Wohnort vor Studienbeginn“ gewechselt wird, wird die Schweiz wieder eingeblendet.

Kartenausschnitt:	Herkunft:
<input type="radio"/> Welt	<input checked="" type="radio"/> Wohnort vor Studienbeginn
<input type="radio"/> Europa	<input type="radio"/> Nationalität
<input checked="" type="radio"/> Schweiz	

Abbildung 10: Herkunft: Wohnort vor Studienbeginn

Dies wird im SVG-Code der „index.svg“-Datei abgehandelt, indem auf den schwarzen Punkt zusätzlich zur „selectOrigin“-Funktion die „handleToggleButton“-Funktion gelegt wird, die den Wechsel zwischen den Kartenausschnitten steuert:

SVG-Code:

```
<use xlink:href="#switchCircle" id="residenceCircle" x="830" y="270"
onclick="selectOrigin(evt, 'Residence');handleToggleButton(evt,
'Switzerland');"/>

<use xlink:href="#switchCircle" id="nationalityCircle" x="830" y="290"
onclick="selectOrigin(evt, 'Nationality');handleToggleButton(evt,
'Europe');"/>
```

Dabei werden bei „Wohnort vor Studienbeginn“ die Schweiz und bei „Nationalität“ Europa als Defaultwerte gesetzt. Das heisst, dass bei einem Wechsel auf „Nationalität“ automatisch die Europakarte eingeschaltet wird.

### 5.5.3 Ablauf der Datenbankabfrage

Dem Benutzer werden verschiedene Selektionslisten vorgegeben, die er beliebig zusammenstellen kann und deren Auswahl in der Karte graphisch dargestellt wird. Für ihn nicht sichtbar, stellt er mit der Benützung der Selektionslisten im Hintergrund die Datenbankabfrage zusammen. Jede Selektionsliste übergibt eine Variable, die dann mit allen anderen Variablen an die Datenbankabfrage übergeben wird.

Im Folgenden werden die einzelnen Selektionslisten und deren Funktion erläutert. Weiter wird beschrieben, wie die Abfrage zu Stande kommt und es wird genauer auf die Übergabe an PHP, PHP selber, die Datenbankabfrage mit SQL und das Zurückgeben der Ergebnisse von PHP an JavaScript und deren Weiterverarbeitung in JavaScript eingegangen.

#### 5.5.3.1 Zusammenstellung der Datenbankabfrage

Die folgenden Selektionslisten und Objekte stellen die Variablen, die für die Datenbankabfrage zuständig sind, zusammen und übergeben diese einer Funktion. Die übergebenen Variablen müssen mit den Spaltennamen der Datenbanktabellen übereinstimmen, da sonst die Abfrage unzulässig ist und kein Resultat geliefert wird. Alle aufgeführten JavaScript-Codes für die Selektionslisten entstammen der Datei „main.js“.

- **Jahr**

In dieser Selektionsliste kann der Benutzer das darzustellende Jahr auswählen. Die Jahreszahl ist gekoppelt mit der Auswahl von Bestand / Neueintritte, da die Spaltennamen der Datenbanktabelle NE\_JAHR (Neueintritte) und JAHR (Bestand) heissen. Deshalb wird hier zuerst die aktuelle Auswahl der Selektionsliste Bestand / Neueintritte abgefragt und anschliessend der SQL-Befehl zusammengestellt:

```
JavaScript-Code:
    Jahr = arrayVal;           //Wert der Selektionsliste, z.B. 1990
    if (entryresult == "Bestand"){
        sql1 = "JAHR="+arrayVal;
    } else {
        sql1 = "NE_JAHR="+arrayVal ;
    }
}
```

- **Bestand / Neueintritte**

Hier kann der Benutzer zwischen dem Bestand und den Neueintritten einer bestimmten Jahreszahl wählen. Die Spaltennamen der Datenbanktabelle heissen NE\_JAHR (Neueintritte) und JAHR (Bestand). Die Variable kann so aber noch nicht übergeben werden, da die Jahreszahl noch fehlt. Die Jahreszahl wird aus der Selektionsliste „Jahr“ übernommen. Schlussendlich sieht der SQL-Befehl, der direkt an die Datenbankabfrage übergeben wird, z.B. so aus: JAHR =1990.

Der entsprechende JavaScript-Code lautet folgendermassen:

```
JavaScript-Code:
    entryresult = arrayVal;           //Wert der Selektionsliste
    if (entryresult == "Bestand"){
        sql1 = "JAHR="+Jahr;         //Jahr stammt aus der
                                     Selektionsliste "Jahr" (z.B. 1990)
    } else {
        sql1 = "NE_JAHR="+Jahr;
    }
}
```

- **Studienkategorie**

In dieser Selektionsliste ist die Studienkategorie Diplom-, Nachdiplomstudierende, oder Doktorierende auszuwählen. Die Unterscheidung der Kategorien ist in der Datenbank mit dem Attribut STUKAT gelöst. Für die drei Gruppen wurde eine Codierung mit den Nummern eins (Diplomstudierende), zwei (Nachdiplomstudierende) und drei (Doktorierende) gewählt. Dementsprechend wird der Variablenwert im JavaScript mit Hilfe einer Switch-Funktion zugeteilt:

```
JavaScript-Code:
    switch(selNr) { //selNr bezieht sich auf die Nummerierung
                   innerhalb der Selektionsliste
    case 0:
        sql2 = "STUKAT!=0";
        break;
    case 1:
        sql2 = "STUKAT=1";
        break;
    ...
    }
```

„case 0“ deckt den Fall ab, dass alle Studienkategorien zusammen dargestellt werden sollen. Der Ausdruck „!=0“ (ungleich 0) erlaubt die Selektion aller Studienkategorien.

- **Studiengang**

In dieser Selektionsliste ist der Studiengang auszuwählen. Auch in diesem Fall existiert in der Datenbank eine Codierung, die jedem Studiengang eine dreistellige Nummer zuteilt.

```
JavaScript-Code:
    switch(selNr) { //selNr bezieht sich auf die Nummerierung
                    innerhalb der Selektionsliste

    case 0:
        sql2 = "STURI_ID!=0";
        break;
    case 1:
        sql3 = "STURI_ID=751";
        break;
    ...
    }
```

Wiederum deckt „case0“ den Fall 'Summe aller Studiengänge' ab. Der Ausdruck „!=0“ (ungleich 0) erlaubt die Selektion aller Studiengänge.

- **Geschlecht**

Beim Geschlecht wird unterschieden zwischen „Männer und Frauen zusammen“, „nur Männer“, „nur Frauen“ und „Männer und Frauen im Vergleich“. Die Codenummer der Datenbank für Männer ist die Eins und die der Frauen ist die Null.

```
JavaScript-Code:
    switch(selNr) { //selNr bezieht sich auf die Nummerierung
                    innerhalb der Selektionsliste

    case 0:
        sql4 = "null";
        break;
    case 1:
        sql4 = "SEX=0";
        break;
    case 2:
        sql4 = "SEX=1";
        break;
    case 3:
        sql4 = "comp";
        break;
    }
```

Da der Code „0“ bereits für die Frauen besetzt ist, kann für die Darstellung der Summe von Männern und Frauen nicht mehr der Ausdruck „!=0“ verwendet werden. Auch für den Vergleich der beiden Geschlechter wird ein separater Ausdruck „comp“ (steht für Compa-

rierson) verwendet. Beide Variablen ziehen eine Fallunterscheidung im PHP-Code nach sich.

Mit den aufgelisteten Selektionslisten ist die Datenbankabfrage aber noch nicht vollständig. Die Haupttabelle der Datenbank enthält drei Attribute, die für die Unterscheidung „Wohnort vor Studienbeginn“ / „Nationalität“ und Schweiz / Ausland zuständig sind. Es sind dies:

- H\_LANDNR (Staatsnummer gemäss BfS nach Wohnort vor Studienbeginn)
- N\_LANDNR (Staatsnummer gemäss BfS nach Nationalität)
- H\_BEZNR4\_02 (Bezirksnummer gemäss BfS nach Wohnort vor Studienbeginn)

Anhand dieser Attribute kann die Datenbankabfrage vervollständigt werden. Diese Variablen werden bei den unter Kapitel 5.5.1 beschriebenen Switch-Schaltflächen der Kartenausschnitte in der Datei „main.js“ übergeben. Hier am Beispiel der Welt zu sehen:

```
JavaScript-Code:
    if (visirespoint == "visible"){ //visirespoint = Switch-Button von
                                    Wohnort vor Studienbeginn
        sql5 = "H_LANDNR!=8100";
    }else{
        sql5 = "N_LANDNR!=8100";
    }
}
```

Der Ausdruck „!=8100“ steht für „ungleich Schweiz“, da 8100 die Staatsnummer der Schweiz repräsentiert. Wenn die Werte der Schweiz mit in die Auswertung der Resultate genommen würden, wäre ein internationaler Vergleich nicht mehr möglich, da die meisten Studierenden der ETH aus der Schweiz kommen.

Mit den Variablen sql1-5 ist die Abfrage der Haupttabelle der Datenbank vollständig. Als Ergebnis wird die absolute Anzahl der ETH-Studierenden pro Bezirk bzw. pro Land herausgegeben.

Da für die Choroplethendarstellung Relativwerte (Anzahl Studierende / Bevölkerungszahl) von Nöten sind, muss zusätzlich zur ersten Abfrage eine zweite unabhängige Abfrage durchgeführt werden, die nur die Bevölkerungszahlen der Bezirke bzw. Länder extrahiert. Natürlich ist diese Abfrage wiederum mit der Jahreszahl gekoppelt. Für diese Abfrage wird in der JavaScript-Datei „main.js“ eine Variable sql6 erstellt, die beim Wechsel eines Kartenausschnittes und in der Selektionsliste „Jahr“ übergeben wird:

```
JavaScript-Code:
// für die Schweiz:
    sql6 = "BEZNR4_2002, POP_ "+JahrPop+" from POPULATION_SCHWEIZ";
// für die Welt:
    sql6 = "STAAT, POP_ "+JahrPop+" from POPULATION_WELT";
// für Europa
    sql6 = "STAAT, POP_ "+JahrPop+" from POPULATION_WELT where STAAT<8300";
```

Europa weist eine zusätzliche Einschränkung in der Abfrage auf: In der Landnummer-Codierung des Bundesamtes für Statistik sind alle Länder, die zu Europa zählen mit einem Code ausgerüstet, der zwischen 8100 und 8300 liegt. Die Staatsnummer der Schweiz ist mit 8100 die kleinste Landnummer, die in der Codierung aufgeführt ist. Die Abfrage kann auf Europa beschränkt werden, indem der Befehl „STAAT < 8300“ an die Abfrage, die für die Welt gemacht wird, angefügt wird.

### 5.5.3.2 Übergeben der Variablen an PHP

Die Variablen sql1-5 werden nun der Funktion „requeststud“ übergeben, die Variable sql6 der Funktion „requestpop“. Beide Funktionen rufen die zugehörige PHP-Datei (oraclestud.php und oraclepop.php) auf und übergeben dieser die entsprechenden Variablen. Nach der Ausführung des PHP-Codes werden die Resultate an die Funktion „parseFilestud“ bzw. „parseFilepop“ übergeben, die diese dann in JavaScript weiterverarbeitet.

```
JavaScript-Code:
function requeststud(sql1,sql2,sql3,sql4,sql5) {
    getURL('oraclestud.php?sql1='+sql1+'...'&sql5='+sql5,parseFilestud);
}

function requestpop(sql6){
    getURL('oraclepop.php?sql6='+sql6,parseFilepop);
}
```

### 5.5.3.3 Erstellung und Ausführung des PHP-Skriptes

In PHP existieren für jeden Datenbanktyp eine eigene Sammlung an Befehlsformen. Da in diesem Projekt mit einer Oracle-Datenbank gearbeitet wird, müssen oci-Funktionen verwendet werden. Je nach PHP-Version gibt es aber auch hier wieder Unterschiede zwischen den einzelnen Befehlen. Die Benützung der oci-Funktionen, die ohne Unterstrich aneinander geschrieben sind, sollte jedoch für jegliche PHP-Version problemlos sein.

Will man in PHP eine Datenbankabfrage machen, muss immer zuerst die Verbindung mit „oci-

logon“ hergestellt werden (und am Schluss des Skriptes wieder geschlossen werden). Dafür muss der Datenbanktyp, der Benutzername und das Passwort eingegeben werden:

```
PHP-Code:
$db = "his";      //Variablen werden in PHP mit $ gekennzeichnet.
$user = "username";
$pw = "password";

$conn = ocilogon($user, $pw, $db);    //Verbindung zur Datenbank
....
ocilogout($conn);                    //Beenden der Datenbank-Verbindung
```

Nun kann die Datenbankabfrage gestartet werden. Dafür muss zuerst der richtige Abfrage-Befehl gefunden werden.

Zuerst wird auf die Datei „oraclestud.php“ eingegangen, das die Anzahl Studierenden pro Bezirk bzw. Land extrahieren soll.

Wie bereits erwähnt, muss eine Fallunterscheidung für die übergebene Variable des Geschlechts unternommen werden. Dabei werden die Fälle „Frauen und Männer zusammen“ und „Frauen und Männer im Vergleich“ besonders behandelt:

```
PHP-Code:
if ($sql4 == "null"){
    ...
}
elseif ($sql4 == "comp") {
    ...
}
elseif ($sql4 != "comp" && $sql4 != "null") {
    ...
}
```

Diese Fallunterscheidung wird noch weiter differenziert, da es noch darauf ankommt, ob der Wohnort vor Studienbeginn im In- oder Ausland oder die Nationalität gefragt ist. Bei dieser Unterscheidung ändert sich lediglich der Spaltenname, der berücksichtigt werden soll:

```

PHP-Code:
    if ($sql4 != "comp" && $sql4 != "null"){
        if ($sql5 == "H_BEZNR4_02!=0"){
            $query = "select count(H_BEZNR4_02) AS COUNT, H_BEZNR4_02 from
                studierende where sql1 and sql2 ...";
        }
        if ($sql5 == "H_LANDNR!=8100" || sql5= = ...){
            $query = "select count(H_LANDNR) AS COUNT, H_LANDNR from
                studierende ... ";
        }
        if ($sql5 == "N_LANDNR!=8100" || sql5= =...){
            $query = "select count(N_LANDNR) AS COUNT, N_LANDNR from
                studierende ... ";
        }
    }
}

```

Diese sehr ausführliche Fallunterscheidung ist nicht abkürzbar, da die jeweiligen Spaltennamen angegeben werden müssen und diese von Fall zu Fall ändern.

Das „select count(z.B. H\_BEZNR4\_02) as COUNT ... group by H\_BEZNR4\_02“ bewirkt das Zusammenzählen aller Einträge, die z.B. dieselbe Bezirksnummer vorweisen. Die Anzahl der Einträge wird als Summe in einer neuen Spalte „neuer Spaltenname“ eingetragen. Der Output-Unterschied zwischen einer einfachen Abfrage und der COUNT-Abfrage sieht folgendermassen aus:

```

SQL-Befehl mit Output:

select * from studierende:                select count(H_BEZNR4_02) as COUNT ....
                                           group by ...:

    SEX  H_BEZNR4_02                        COUNT      H_BEZNR4_02
    ---  ---                                ---        ---
    0    101                                2          101
    1    101                                1          102
    0    102                                2          103
    1    103                                ...        ...
    1    103
    ...  ...

```

Falls die Männer und Frauen zusammen (\$sql4 == "null") dargestellt werden sollen, vereinfacht sich der Abfrageterm einfach durch das Weglassen der sql4-Bedingung. Somit wird die Summe der Männer und der Frauen zurückgeliefert.

Etwas komplizierter gestaltet sich die Abfrage für den Vergleich zwischen Männern und Frauen:

PHP-Code:

```
$query = "select count(...) AS COUNT, ... sum(SEX) as NRMALE from
studierende where ".$sql1." and ".$sql2." and ".$sql3." and
".$sql5." group by ..."; //"..." stehen für zugehöriges sql5
```

In diesem Fall erweitert sich der SQL-Befehl um den Term „sum(SEX) as NRMALE“. Dieser Ausdruck zählt zusätzlich zu der Gesamtanzahl Personen, die in die Spalte COUNT geschrieben wird, noch die Anzahl Männer pro Bezirk zusammen. Der Output sieht folgendermassen aus:

SQL-Befehl mit Output:

```
select count(...) AS COUNT, ... sum(SEX) as NRMALE:
```

COUNT	H_BEZNR4_02	NRMALE
2	101	1
1	102	0
2	103	2
..	...	..

Das Resultat ist zwar richtig, hängt aber zufälligerweise mit der Codierung des Geschlechtes in Zusammenhang. Alle Männer besitzen beim Attribut „SEX“ den Wert „1“ auf Grund der vorliegenden Codierung. Die Funktion sum() zählt immer alle Felder zusammen, die den Wert 1 enthalten. Wenn also die Codierung für die Männer z.B. 11 betragen würde, würde der Output nicht mehr die richtige Lösung bringen.

Nachdem der richtige SQL-Befehl gefunden wurde, kann nun die Abfrage durchgeführt werden. Hierfür benötigt es eine Reihe von Befehlen, die nacheinander ausgeführt werden müssen:

PHP-Code:

```
$resquery = ociparse($conn,$query);           //Bereitet den Befehl für
                                                die Ausführung vor

ociexecute($resquery);                        //Führt den Befehl aus

$studanz = ocifetchstatement($resquery,$results); //füllt die
                                                Resultate in den
                                                Array „results“
```

Der Array<sup>43</sup> „results“, der die Abfrageergebnisse enthält, kann nun weiter verarbeitet werden. Mit dem Ausgabebefehl „echo“ und den erhaltenen Resultaten wird ein XML-File erstellt, das später der JavaScript-Datei „main.js“ übergeben und von dieser eingelesen wird.

Zuerst wird eine Gruppe mit der id=“Students“ kreiert. In dieser Gruppe wird jede Zeile des Arrays als ein <stud>-Element aufgeführt. Dabei werden die Spaltennamen (COUNT, H\_BEZNR4\_02, etc.) als Attribute des <stud>-Elementes eingesetzt und die Zeilenwerte als deren Werte:

```
Output:
<g id="Students" xmlns:stud="http://www.ika.ethz.ch/schnabel">
  <stud COUNT="2" H_BEZNR4_02="101" />
  <stud COUNT="1" H_BEZNR4_02="102" />
  <stud COUNT="2" H_BEZNR4_02="103" />
  ...
</g>
```

Wenn der Geschlechter-Vergleich gefragt ist, wird die Ausgabe der <stud>-Elemente ein wenig komplexer. Da bei der Abfrage nur die Gesamtanzahl der Studierenden und die Anzahl der Männer berücksichtigt wird, muss die Anzahl der Frauen noch berechnet werden. Die Resultate werden zwar nicht mehr in den Array geschrieben, jedoch bei der Ausgabe in der „Students“-Gruppe angefügt. Der Array \$allkeys, der alle Attributnamen enthält, wird zuerst mit dem Befehl `array_push($allkeys,'NRFEMALE')` um den Attributnamen „NRFEMALE“ erweitert. Danach kann die Ausgabe der <stud>-Elemente auf einfache Weise um das neue Attribut und dessen Werte, die sich aus der Differenz der Gesamtanzahl der Studierenden und der Anzahl Männer ergeben, erweitert werden:

```
Output:
  <stud COUNT="2" H_BEZNR4_02="101" NRMAL="1" NRFEMALE="1" />
  <stud COUNT="1" H_BEZNR4_02="102" NRMAL="0" NRFEMALE="1" />
  <stud COUNT="2" H_BEZNR4_02="103" NRMAL="2" NRFEMALE="0" />
  ...
```

Die Datei „oraclepop.php“, das die Bevölkerungszahl pro Bezirk bzw. Land extrahieren soll, ist grundsätzlich gleich aufgebaut wie die eben behandelte Datei. Nur fallen die ganzen Fallunterscheidungen weg, da der definitive SQL-Befehl bereits in der JavaScript-Datei „main.js“ bestimmt wird. Die Ausgabe geschieht in diesem Fall in einer Gruppe mit der id=“Population“ und mehreren <pop>-Elementen. Sie fällt sehr einfach aus. Das Ergebnis ist ein Element, das zwei

---

<sup>43</sup>Siehe Glossar: Stichwort „Array“

Attribute, die Bezirks- oder Landnummer und die Einwohnerzahl, mit den entsprechenden Werten besitzt.

```
Output:
  <g id="Population" xmlns:pop="http://www.ika.ethz.ch/schnabel">
    <pop LANDNR="8600" POP_90="6830068" />      // "LANDNR" austauschbar mit
                                                "BEZNR4_2002"
    <pop LANDNR="9012" POP_90="875436208" />
    ...
  </g>
```

Die so erstellten XML-Files werden im nächsten Schritt in der JavaScript-Datei wieder eingelesen.

Bei der Arbeit in PHP ist zu beachten, dass PHP selber nicht vom Browser interpretiert werden kann. Um PHP-Abläufe zu überprüfen, müssen die Inhalte immer mit dem „echo“-Befehl und in Verbindung mit HTML- oder SVG-Syntax ausgegeben werden. Das Erschaffen von aussagekräftigen und nützlichen Fehlermeldungen ist ein sehr hilfreiches Mittel bei der Suche nach Fehlern.

#### 5.5.3.4 Verarbeitung der Ergebnisse in JavaScript

Die Erläuterungen in diesem Abschnitt beziehen sich alle auf die JavaScript-Datei „main.js“

- **Einlesen des XML-Files in JavaScript**

Das Einlesen des XML-Files in die JavaScript-Datei „main.js“ wird mit der Funktion „parseXML(obj.content)“ vorgenommen. Dieser Funktion liegen mehrere Schritte zugrunde. Zuerst werden alle <stud>-Elemente (oder <pop>-Elemente) im XML-File gesucht. Danach werden alle Attribute, die ein solches Element aufweist, herausgesucht und in einen Array geschrieben. Am Schluss werden die Attributwerte zusammen mit den Attributen in einem neuen, mehrdimensionalen Array abgespeichert.

Bei der Suche nach den <stud>-Elementen kann auf die Objekthierarchie, dem Document Object Model (DOM) von XML, zurückgegriffen werden. Dabei muss beachtet werden, dass nicht jeder DOM-Knoten ein <stud>-Element ist. Es werden mehrere leere Knoten kreiert, die nicht berücksichtigt werden dürfen.

Genauere Erläuterungen zu dieser Funktion finden sich im Anhang B1.

Mit dem Auslesen der XML-Files sind zwei Arrays, einer für die Anzahl Studierenden pro Bezirk bzw. Land (studETH) und einer mit der Bevölkerungszahl pro Bezirk bzw. Land (popu) erstellt

worden, mit denen in JavaScript weiter gearbeitet wird.

- **Berechnung der Relativwerte für die Darstellung**

Damit eine Choroplethendarstellung möglich ist, müssen Relativwerte vorhanden sein. Die durch die Abfrage erhaltenen Werte, welche in die Arrays gespeichert wurden, sind jedoch Absolutwerte. Somit ist eine Umrechnung dieser Absolutwerte in Relativwerte notwendig.

Dafür wird die absolute Anzahl Studierenden eines Bezirkes bzw. eines Landes durch dessen Bevölkerungszahl geteilt. Die Ergebnisse werden in einen neuen Array (ETHquote) gespeichert, der als erstes Attribut den Relativwert des Bezirkes bzw. Landes und als zweites Attribut die Bezirks- oder Staatsnummer enthält. Im Falle des Vergleichs „Männer-Frauen“ werden zusätzlich zur Gesamtanzahl der Studierenden noch die Relativwerte für die Männer und die Frauen gespeichert. Die Inhalte des Arrays sehen nun wie folgt aus:

```
Output des Arrays:
```

ETHQuote	H_BEZNR4_02	ev. NRMAL	ev. NRFEMALE
0.125	102	0.100	0.025
0.035	2603	0.023	0.012
0.065	2307	0.050	0.015
...	...	...	...

Dieser Array wird in einem nächsten Schritt nach dem Attribut „ETHQuote“ aufsteigend sortiert. Dies ist nötig, weil dem Benutzer die Klassierungsart „Quantile“ zur Auswahl steht. Die Grundidee der Quantile ist, dass in jeder Intervallklasse die gleiche Anzahl an Werten enthalten ist. Für die Berechnung der Intervallgrenzen müssen die für die Berechnung relevanten Werte der Größe nach sortiert werden. Im JavaScript-Code wird dies mit einer simplen Funktion „sortarray(a,b)“ durchgeführt, die immer zwei Werte miteinander vergleicht und den kleineren der beiden an die erste Stelle platziert. Der Inhalt des Arrays sieht nach der Anwendung der Funktion folgendermassen aus:

```
Output des Arrays:
```

ETHQuote	H_BEZNR4_02	ev. NRMAL	ev. NRFEMALE
0.035	2603	0.023	0.012
0.065	2307	0.050	0.015
0.125	102	0.100	0.025
...	...	...	...

- **Berechnung der Intervallgrenzen**

Der Benutzer kann in zwei Selektionslisten die Klassierungsart der Wertintervalle (gleiche Intervalle oder Quantile) und die Anzahl Klassen, die die Darstellung enthalten soll, wählen.

- *Klassierungsart und Anzahl Klassen*

Es wird ein Array erstellt, in dem die Werte der Intervallgrenzen gespeichert werden. Je nach Klassierungsart und Anzahl der Klassen unterscheiden sich die Intervallgrenzen.

- Gleiche Intervalle: Die Intervallbreite ist immer gleich.
- Quantile: Alle Intervalle enthalten die gleiche Anzahl Werte.

Weitere Erläuterungen zur Klassierungsart „Gleiche Intervalle“ und „Quantile“ sind bereits in Kapitel 2.1.2.2 beschrieben.

Im JavaScript-Code wird die Berechnung der Intervallgrenzen für die Methode „gleiche Intervalle“ wie folgt gelöst:

```
JavaScript-Code:
//Minimalwert = erster Wert in ETHQuote-Array;
dataminval = ETHquote[0][attrquote[0]]

//Maximalwert = letzter Wert in ETHQuote-Array;
datamaxval = ETHquote[ETHquote.length-1][attrquote[0]];

var borderarray = new Array(); //Erstellt Array, der Grenzwerte
                               enthalten soll

//Methode gleiche Intervalle
if (valcalculation=="gleiche Intervalle"){
    //Intervallbreite
    var intervallwidth = (datamaxval - dataminval) / valclasses;
    var firstborder = dataminval;
    borderarray.push(firstborder); //schreibt Wert in Array (an
                                   letzter Stelle)
    //füllt Array mit restlichen Werten; valclasses = Anzahl Klassen
    for (var i=0;i<valclasses;i++) {
        firstborder = firstborder + intervallwidth;
        borderarray.push(firstborder);
    }
}
```

Die Berechnung der Quantile ist etwas komplexer:

Zuerst wird die Anzahl der Werte, die in einer Klasse enthalten sind, berechnet. Dabei wird die

Länge des ETHQuote-Arrays<sup>44</sup> durch die Anzahl Klassen geteilt und gerundet. Falls ein Rest entsteht, wird dieser in einer Variablen gespeichert:

```
JavaScript-Code:  
    //Anzahl Werte pro Klasse  
    var nrPerClass = Math.floor(ETHquote.length / valclasses);  
  
    //Restwert  
    var modVal = ETHquote.length % valclasses;
```

In weiteren Berechnungsschritten werden die Intervallgrenzen gesetzt. Um eine unterbrochene Skala zu vermeiden, werden nicht wie üblich die Werte des ersten und des letzten Klassenmitgliedes, sondern die Mittelung der jeweils angrenzenden Unter-und Obergrenze als Intervallgrenzen gesetzt.

Beispiel:

unterbrochene Skala:	0-3	5-10	16-20
durchgehende Skala:	0-4	4-13	13-20

Wenn die Berechnung der Anzahl Werte pro Klasse einen Rest ergibt, müssen diese Restwerte auf die einzelnen Intervalle verteilt werden. Dies alles ist in der JavaScript-Funktion berücksichtigt worden. Den genauen Code und weitere Erläuterungen dazu findet sich im Anhang B.2

#### • Berechnung der Farbwerte

Nachdem die Intervallgrenzen berechnet sind, kann mit der Berechnung der Farbwerte der einzelnen Klassen begonnen werden. Anschliessend wird die Karte mit diesen Farben eingefärbt. Gleichzeitig wird eine Legende dazu erstellt.

Farben können generell entweder mittels dem RGB<sup>45</sup>- oder dem HSV<sup>46</sup>-Code angegeben und berechnet werden. In SVG werden Farben jedoch ausschliesslich mit dem RGB-Code definiert. Demzufolge muss eine Farbe, die im HSV-Code definiert ist, in den RGB-Code transformiert werden, damit sie in SVG dargestellt wird.

---

44 ETHQuote enthält die Relativwerte (Anzahl Studierende / Bevölkerungszahl) pro Bezirk bzw. Land

45 Siehe Glossar: Stichwort „RGB-Farbraum“

46 Siehe Glossar: Stichwort „HSV“

Für die Berechnung der Farbklassen wird die Sättigung verwendet. Damit wird erreicht, dass der Farbton immer derselbe bleibt, die Sättigung von Klasse zu Klasse aber abnimmt:



Abbildung 11: Farbverlauf

Der Anteil der Sättigung wird anhand der Anzahl Klassen berechnet. Beispiel: Bei 5 Klassen beginnt der Sättigungswert bei 0.2 und nimmt dann bei jeder Klasse um 0.2 zu, bis er am Schluss den Wert 1 erreicht.

Die erhaltenen HSV-Werte der einzelnen Klassen müssen in einem weiteren Schritt in RGB-Farben umgewandelt werden. Dafür wird eine von Andreas Neumann, Institut für Kartographie der ETH Zürich, erstellte Funktion „hsv2rgb“ verwendet, die die Transformationsparameter enthält. Das Ergebnis ist ein Array, mit den Attributnamen „red“, „green“ und „blue“ und den zugehörigen Werten zwischen 0 und 255.

- **Einfärbung der Karte**

Die Bezirke bzw. Länder sollen mit den entsprechenden Farben eingefärbt werden. Für die Zuweisung der richtigen Farbe zu den Bezirken bzw. Ländern werden der Farb- (myColorArray) und der Intervallgrenzen-Array (borderarray) benötigt.

```
JavaScript-Code:
for (var colIndex=0;colIndex<valclasses;colIndex++){ //colIndex = Index des
                                                    Farbarrays
if (borderarray[colIndex] <= ETHquote[i][attrquote[0]] && ETHquote[i]
    [attrquote[0]]< borderarray[colIndex+1]) {

    unitsId = ETHquote[i][attrquote[1]];

    //Einfärbung der Bezirke, wenn Schweiz gefragt
    if (sql5 == "H_BEZNR4_02!=0"){
        var myElement = document.getElementById("bez_"+unitsId);
        myElement.setAttributeNS(null,"fill","rgb(...,...,...)");
    }
}
```

```
//Einfärbung der Länder, wenn Welt oder Europa gefragt
else{
    var myElement = document.getElementById("world_"+unitsId);
    myElement.setAttributeNS(null,"fill","rgb(.....)");
}
}
}
```

Bei der ersten if-Abfrage wird gefragt, ob der Relativwert des ersten Arrayeintrages (ETHquote) zwischen der ersten und der zweiten Intervallgrenze liegt. Wenn dies zutrifft, wird in einer zweiten if-Anweisung gefragt, ob die Schweizer- oder die Weltkarte eingefärbt werden soll. In der entsprechenden Karte wird nach der ID des Bezirkes bzw.Landes gesucht und dessen Füllfarbe neu gesetzt. Während des ganzen Prozesses werden alle Intervallgrenzen und die zum Intervall gehörige Farbe durchlaufen.

Nicht zu vergessen ist eine weitere Differenzierung, die die Klassierungsart abfragt und die Einfärbung entsprechend anpasst.

Das Resultat sieht folgendermassen aus:

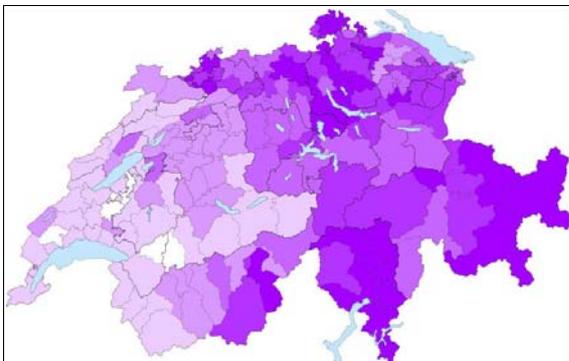


Abbildung 12: Einfärbung "Quantile"

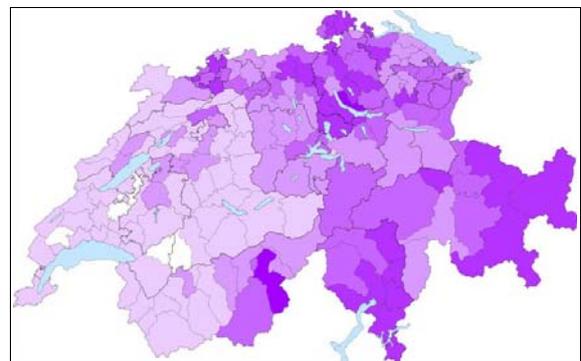


Abbildung 13: Einfärbung "Gleiche Intervalle"

#### • Erstellen der Legende

Bei der Erstellung der Legende wird ausschliesslich mit dem Befehl `createElementNS()` gearbeitet. Dabei wird mit JavaScript der DOM der SVG-Datei um die Elemente der Legende erweitert. Das heisst, dass der JavaScript-Code neue Elemente in die SVG-Datei einfügt.

Die Zahl der zu zeichnenden Legendenkästchen ist durch die Anzahl Klassen bestimmt. Für jede Klasse wird ein Rechteck erstellt, das mit der entsprechenden Farbe gefüllt wird.

Für die Bezirke bzw. Länder, von denen keine Studierenden kommen, die also weiss bleiben,

wird ein Legendenkästchen gezeichnet, das ebenfalls weiss bleibt. Die Rechtecke werden zusätzlich mit den entsprechenden Intervallgrenzen beschriftet. Auch bei diesem Verfahren wird der Befehl `createElementNS()` verwendet. Die fertige Legende sieht bei fünf Klassen folgendermassen aus:

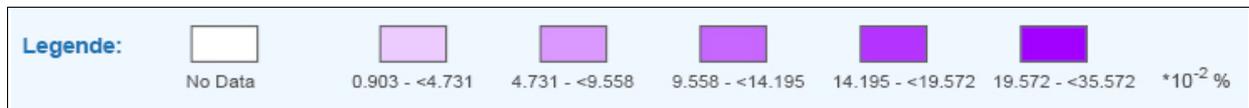


Abbildung 14: Legende

### • Erstellen des Diagramms

Damit der Benutzer die Werte unter sich vergleichen kann, wird ein Histogramm erstellt, das alle Werte der Grösse nach auflistet. Das Prinzip, das dahinter steckt, ist folgendes:

Für jeden Relativwert wird ein Rechteck erstellt, dessen Höhe (im `heightArray` gespeichert) direkt von der Grösse des Wertes und der Höhe des Diagramms (120) abhängt:

```
JavaScript-Code:
    for (var i=0;i<ETHquote.length;i++) {
        heightArray[i] = ((120/datamaxval)*ETHquote[i][attrquote[0]]);
        //datamaxval=grösster Wert
    }
```

Die Breite des Rechteckes wird durch die Breite des Diagramms und die Anzahl der Bezirke bzw. Länder bestimmt:

```
JavaScript-Code:
    xOffset = DiaWidth / popu.length; //für die Schweiz und die Welt
    xOffset = DiaWidth / 47; //für Europa (Europa hat 47 Länder)
```

Diese Zahl bestimmt auch gleich die X-Koordinate der Rechtecke, die jedesmal nachdem ein Rechteck gezeichnet wird, um die Breite des Rechteckes erhöht wird.

Auch für die Bezirke, die keine Daten aufweisen, werden Rechtecke gezeichnet. So bleibt die Rechtecksbreite immer konstant. Die Höhe der „NoData-Rechtecke“ ist abhängig von der Höhe des minimalen Wertes des `heightArrays`. Falls die Höhe dieses Wertes genügend klein ist, wird sie für die NoData-Rechtecke übernommen. Ansonsten wird ein fester Wert gesetzt.

Das Erstellen einer Gruppe, die ein Rechteck mit den Attributen „id“, „x“, „y“, „width“, „height“,

„fill“ und „stroke“ enthält, sieht so aus:

JavaScript-Code:

```
var groupe = document.createElementNS(svgNS, "g");
groupe.setAttributeNS(null, "id", "Diagram_rect");
var node = document.createElementNS(svgNS, "rect");
node.setAttributeNS(null, "id", "Diagramm_" + ETHquote[i][attrquote[1]]);
node.setAttributeNS(null, "x", xdia);
node.setAttributeNS(null, "y", ydia - heightArray[i]);
node.setAttributeNS(null, "width", xOffset);
node.setAttributeNS(null, "height", heightArray[i]);
node.setAttributeNS(null, "fill", "rgb(..., ..., ...)");
node.setAttributeNS(null, "stroke", "rgb(..., ..., ...)");
groupe.appendChild(node); //Hängt Rechteck an Gruppe
```

Bei der Erstellung des Diagramms wird wieder zwischen der Klassierungsart „Quantile“ und „gleiche Intervalle“ unterschieden:

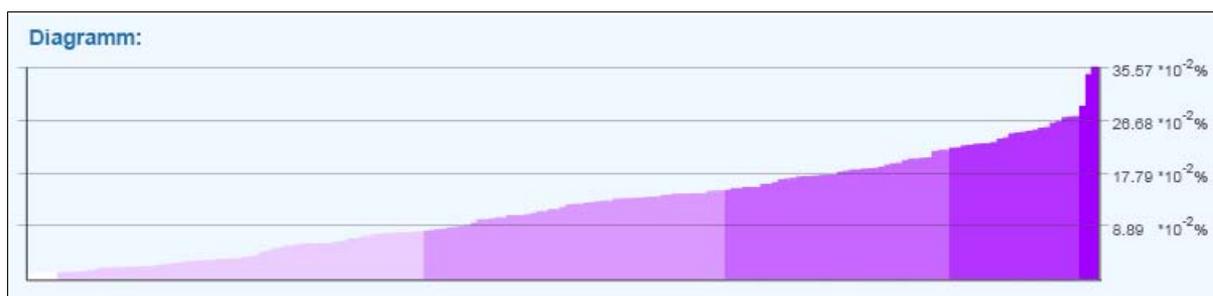


Abbildung 15: Diagramm "gleiche Intervalle"

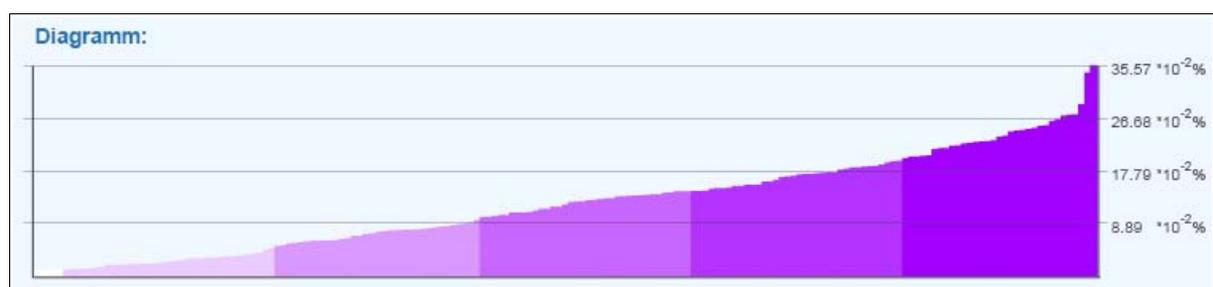


Abbildung 16: Diagramm "Quantile"

Ein Sonderfall stellt die Einstellung „Vergleich Männer-Frauen“ dar:

Bei dieser Einstellung werden die Werte der Männer und Frauen im Diagramm verglichen. Es wird je ein Rechteck für den Männer- und den Frauenanteil gezeichnet. Sie liegen übereinander und bilden zusammen die Summe aller Studierenden.

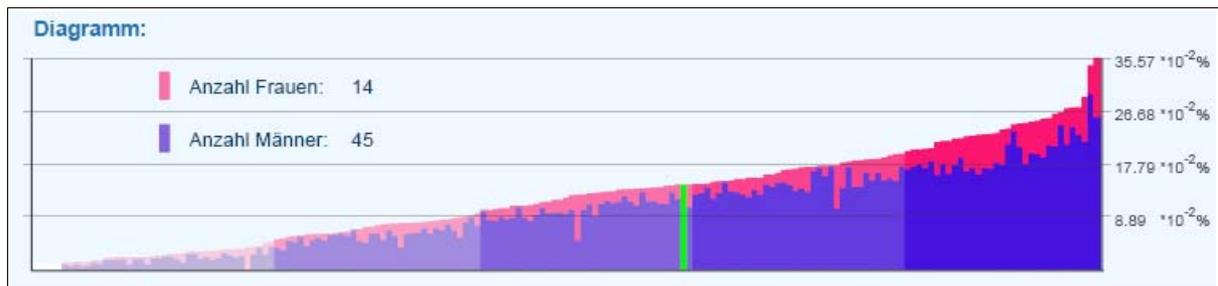


Abbildung 17: Diagramm "Vergleich der Geschlechter"

## 5.5.4 Funktionen der Karte

Die Karte verfügt über weitere Interaktionen, die keine neue Datenbankabfrage generieren, sondern auf die bereits zurückgelieferten Ergebnisse zugreifen.

### 5.5.4.1 Dynamisches Anzeigen von Zusatzinformation

Beim Überfahren eines Bezirkes bzw. Landes mit dem „Maus-Cursor“ wird dieser mit einer anderen Farbe markiert und dem Benutzer dessen Name, die Einwohnerzahl und die Anzahl Studierenden angezeigt. In der Legende wird ebenfalls die entsprechende Klasse markiert. Auch im Diagramm wird das entsprechende Rechteck markiert.

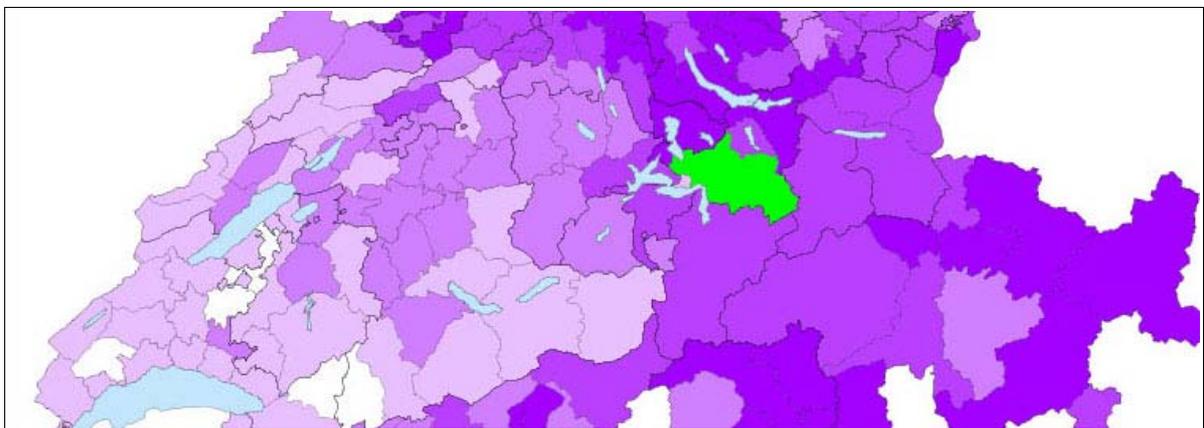


Abbildung 18: Anzeigen des Pfades

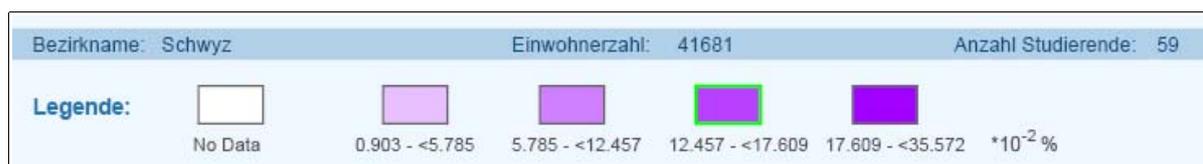


Abbildung 19: Anzeigen von Information in der Legende

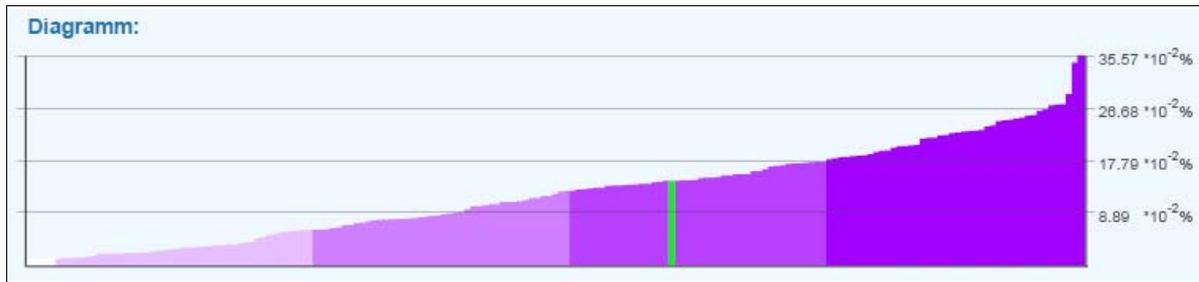


Abbildung 20: Anzeigen von Information im Diagramm

Im SVG-Code der Datei „index.svg“ wird über die Pfade eine Funktion gelegt, die beim Überfahren mit der Maus ausgeführt wird. Dabei wird zwischen „mouseover“ und „mouseout“ unterschieden, da beim Verlassen des Pfades der ursprüngliche Darstellungszustand wieder hergestellt werden soll.

SVG-Code:

```
<g id="bezirke" onmouseover="showDistrictname(evt, 'Bezirke');"
  onmouseout="showDistrictname(evt, 'Bezirke');">
```

Im JavaScript-Code der Datei „main.js“ wird die Funktion definiert und die verschiedenen Aktionen beschrieben.

Das gleiche Prinzip herrscht beim Überfahren der Diagrammrechtecke. Der dazugehörige Bezirk und die Legendenklasse wird hervorgehoben und die Detailinformation wird angezeigt.

#### 5.5.4.2 Dynamischer Selektionslisteninhalt der Anzahl Klassen

Dem Benutzer werden in der Selektionsliste „Anzahl Klassen“ defaultmässig vier verschiedene Wertklassen zur Auswahl gestellt. Es sind dies zwei, drei, vier und fünf Klassen (vgl. Abb.21).

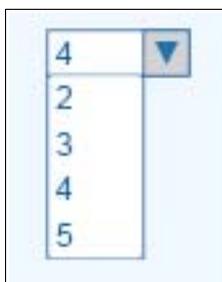


Abbildung 21:  
Default

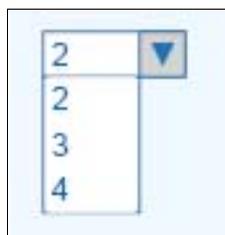


Abbildung 22: vier  
Werte

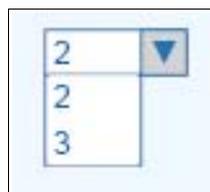


Abbildung 23:  
drei Werte

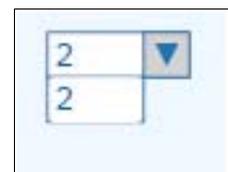


Abbildung 24:  
zwei Werte

Es gibt aber Einstellungen, bei denen die Datenbankabfrage weniger als fünf Werte zurück liefert. Das ist beispielsweise der Fall bei der Anzahl der neu eintretenden Studierenden des Studiengangs „Interdisziplinäre Naturwissenschaften“ in den Jahren 1992 und 1993. Im Jahre 1992 haben sich nur zwei Studierende eingeschrieben und im Jahre 1993 nur deren vier. Die Werte könnten also nicht mehr nach einem logischen Prinzip auf fünf Klassen verteilt werden. Daher muss der Inhalt der Selektionsliste der Anzahl Werte, die die Datenbankabfrage zurück liefert angepasst werden.

Dies wird mit einer Fallunterscheidung erreicht. Wenn die Anzahl der zurückgelieferten Werte kleiner als fünf ist, wird die bestehende Selektionsliste mit dem Befehl „selClass.removeSelectionList()“ gelöscht, wobei der Ausdruck vor dem Punkt (hier: selClass) angibt, welche Selektionsliste gelöscht werden soll. Anschliessend wird der Inhalt der Selektionsliste neu erstellt. Dabei werden so viele Klassen angegeben, wie es Werte gibt (vgl. Abb. 22 – 24). Anschliessend wird die Selektionsliste mit dem neuen Inhalt wieder gesetzt. Zu beachten ist, dass eine nächste Einstellung wieder mehr als fünf Werte aufweisen kann. Der Inhalt der Selektionsliste muss dementsprechend wieder angepasst werden.

### 5.5.4.3 Color-Picker

Dem Benutzer wird in der Karte eine zusätzliche Funktion zur Verfügung gestellt, mit der er die Darstellungsfarbe nach seinem Belieben ändern kann. Für diese Funktion wird das Color-Picker-Werkzeug<sup>47</sup> von Andreas Neumann, Institut für Kartographie der ETH Zürich, verwendet.

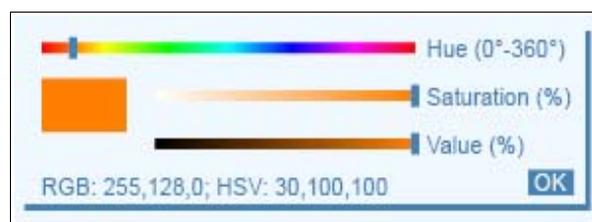


Abbildung 25: Color-Picker

Der Farbton (Hue), die Sättigung (Saturation) und die Helligkeit (Value) kann gewählt werden.

Die Grundeinstellung der Darstellungsfarbe hängt von der Selektionsliste „Geschlecht“ ab. Die Frauen werden in roter, die Männer in blauer und die Summe der Geschlechter in violetter Farbe dargestellt. Wenn dem Benutzer diese Farbgebung nicht entspricht, kann er die Farbe beliebig ändern. Wenn die Einstellung „Geschlechtervergleich“ ausgewählt ist, wird aus Darstellungsgründen der Color-Picker ausgeblendet.

---

47 [URL: Color-Picker]

## 5.6 Resultat des Projektes

Das Ergebnis des gesamten Projektes erscheint auf dem Bildschirm wie folgt:

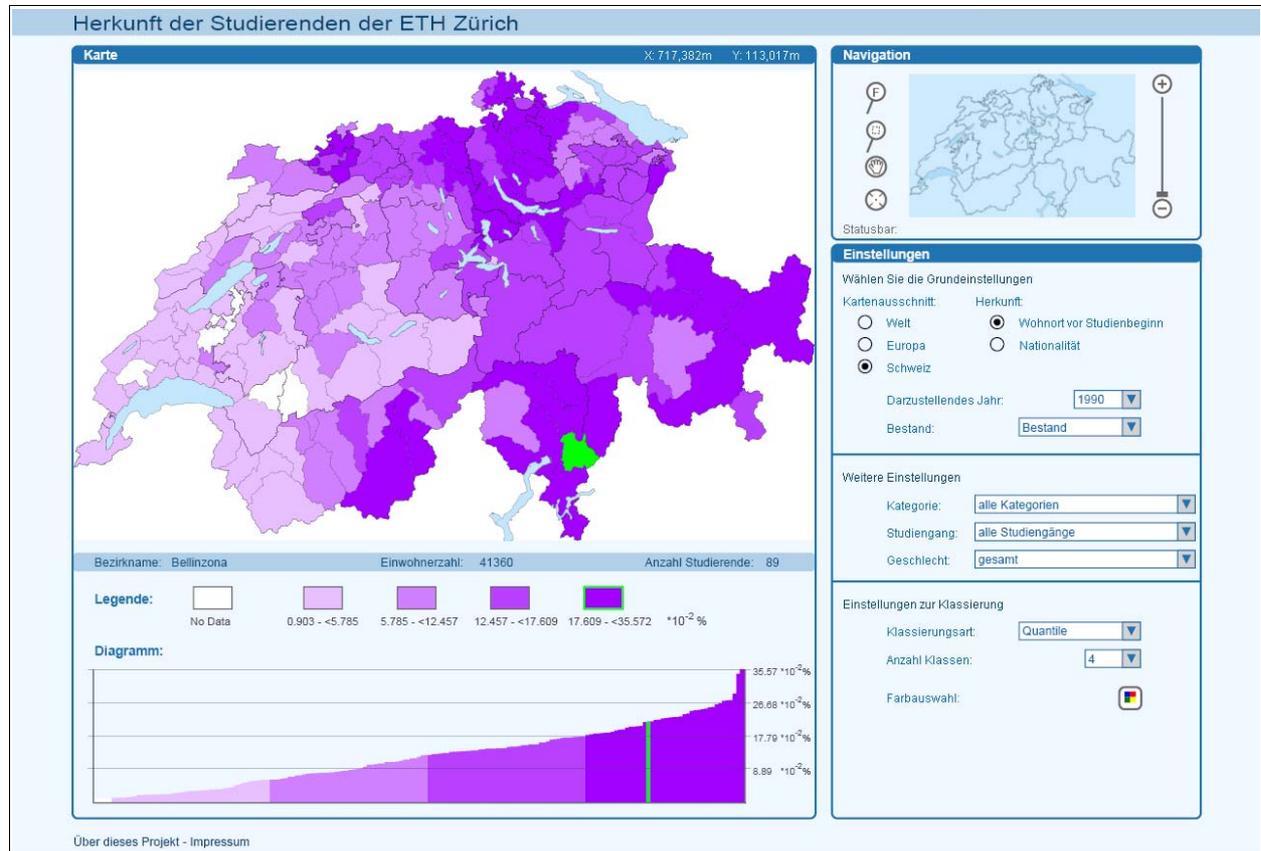


Abbildung 26: Resultat des Projektes

Abbildung 26 zeigt in allen Einstellungen mit Ausnahme der Klassierungsart die Grundeinstellungen, die beim Laden des Projektes angezeigt werden. In der Selektionsliste „Klassierungsart“ wurde die Einstellung „Quantile“ gewählt, was im Diagramm sichtbar ist.

## 6 Diskussion des Resultats

In diesem Kapitel wird das Resultat des Projektes diskutiert. Es sollen Vor- und Nachteile der Karte aufgezeigt werden. Die Karte wurde in verschiedenen Browsern und in den zwei Versionen 3.0 und 6.0 des Adobe SVG Viewers getestet. Das Ergebnis dieses Tests wird ebenfalls aufgeführt.

Das Hauptziel der Karte, nämlich „die Visualisierung der Herkunft der Studierenden der ETH Zürich in der Schweiz und im Ausland“, ist erreicht worden. Die Herkunft der Studierenden wird in einer Schweizerkarte, die in Bezirke aufgeteilt ist, und in einer Weltkarte, die in Länder aufgeteilt ist, dargestellt. Für die Darstellung wurde die Choroplethendarstellung gewählt. Dabei müssen die zu visualisierenden Werte als Relativwerte vorliegen. Das heisst, dass nicht die absolute Anzahl Studierender, sondern die Anzahl Studierender im Verhältnis zur Bevölkerungszahl dargestellt wird. Da sich die Relativwerte der Schweiz, bezogen auf die Bezirke, massiv von denen der Weltländer abheben, drängt es sich auf, das Resultat in Bezug auf die Schweiz und das Ausland getrennt zu diskutieren.

### 6.1 Resultat für die Schweiz

Für die Schweiz fällt das Resultat der Karte sehr positiv aus. Die Werte sind sehr gut sichtbar, sogar dann, wenn eine Auswahl getroffen wird, bei der nur sehr wenige Studierende visualisiert werden.

Das Gegenüberstellen der einzelnen Relativwerte im Diagramm kommt in der Schweizerkarte in jedem Fall sehr gut zum Ausdruck.

Da der Inhalt der Selektionsliste „Anzahl Klassen“ dynamisch generiert wird, sind auch Einstellungen, die weniger als fünf Werte aufweisen, gut darstellbar. Im Diagrammfenster werden bei solchen Einstellungen nur wenige (<5) farbige Rechtecke gezeichnet. Die Rechtecke der Bezirke, die keine Studierenden aufweisen erscheinen in weisser Farbe.

Bei der Gegenüberstellung der Relativwerte im Diagramm kommt auch gut zum Ausdruck, dass die einzelnen Werte ohne Probleme miteinander vergleichbar sind. Bei jeder beliebigen Zusammenstellung der Abfrage ist die Abstufung zwischen den einzelnen Rechteckshöhen und somit der Vergleich der einzelnen Relativwerte sehr gut sichtbar.

## 6.2 Resultat für die Welt und Europa

Die Choroplethendarstellung ist für die Weltländer nicht immer optimal. Verschiedene Probleme kommen hier zum Vorschein:

Die Relativwerte sind meistens so klein, dass sie für den Benutzer nicht mehr erfassbar sind. Sehr viele Länder weisen bei bestimmten Einstellungen so kleine Relativwerte auf, dass mehr als drei Nachkommastellen nötig sind (0.00001). Da die Legende aber nur Werte bis drei Nachkommastellen darstellt, weisen die Intervallgrenzen dieser Länder dann Nullwerte (0.000) auf.

Ein weiteres Problem bilden kleine Länder wie beispielsweise das „Fürstentum Liechtenstein“ oder „Luxemburg“, die eine kleine Einwohnerzahl aufweisen. Ihre Relativwerte sind im Vergleich zu den grösseren Ländern sehr hoch.

Das Fürstentum Liechtenstein wird in der Schweiz oft als 27. Kanton bezeichnet, besitzt aber richtigerweise keine Schweizer Bezirksnummer. Viele Schülerinnen und Schüler aus dem Liechtenstein absolvieren das Gymnasium in der Schweiz, weshalb sie auch ohne Eintrittsprüfung an der ETH studieren können. Liechtenstein weist darum eine verhältnismässig hohe Anzahl an ETH-Studierenden auf. Die Einwohnerzahl von Liechtenstein kommt der eines Bezirks in der Schweiz nahe, was den relativen Anteil an Studierenden in die Höhe schnellen lässt. Wird dieser Wert in einem Diagramm den Werten der anderen Länder gegenüber gestellt, entsteht eine grosse Diskrepanz:

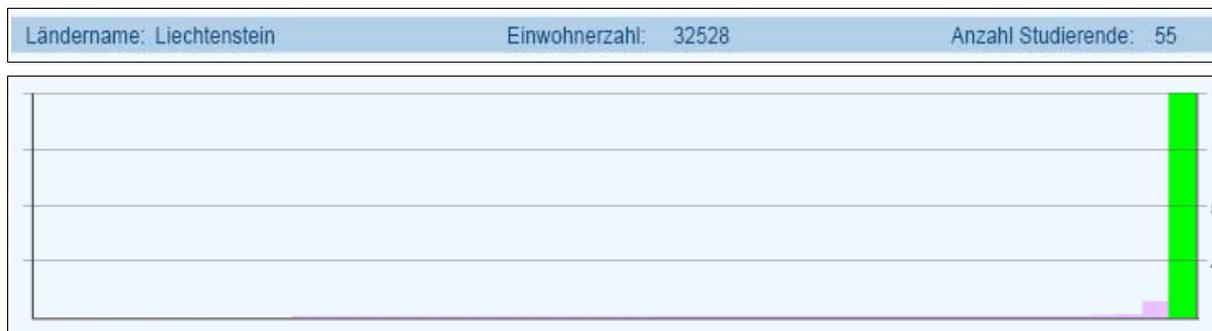


Abbildung 27: Diagramm mit hohem Wert Liechtenstein

Der Vergleich zwischen den anderen Werten wird so praktisch verunmöglicht, da die Höhen der anderen Rechtecke so klein sind, dass sie kaum mehr feststellbar sind. Das Problem liegt aber nicht einfach an Liechtenstein und dessen hoher Studierendenzahl, sondern vielmehr am Umstand, dass kleine Länder immer einen hohen Relativwert aufweisen. Wenn Liechtenstein in einer Einstellung keine Studierenden aufweist, nimmt meistens ein anderes kleines Land, wie zum Beispiel Luxemburg, den Platz des Spitzenreiters ein, wie folgende Abbildung zeigt:

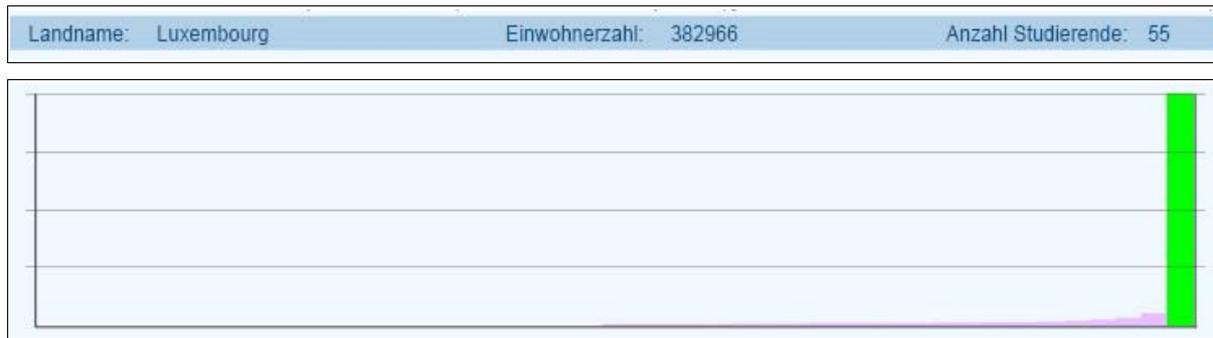


Abbildung 28: Diagramm mit Wert Luxemburg

Hinzu kommt, dass Liechtenstein auf der Karte eine so kleine Fläche einnimmt, dass das Land erst bei einem grossen Zoomfaktor sichtbar wird. Wenn die Einstellung „Gleiche Intervalle“ ausgewählt ist, ist Liechtenstein meist der einzige Wert in der obersten Klasse. Im Diagramm nimmt der Benutzer diesen hohen Wert wahr (siehe Abb. 27) und sieht, dass er in die oberste Klasse eingeteilt ist. Da aber die Fläche von Liechtenstein in der Karte verschwindend klein ist, sieht der Benutzer die Einfärbung des Pfades in der Karte nicht. Erst wenn er mit der Maus über das Diagrammrechteck fährt und so der Ländername angezeigt wird, stellt er fest, dass es sich beim Spitzenreiter der Diagrammeinträge um Liechtenstein handelt. Beim anschliessenden Zoomen auf das gesuchte Land, kann dann festgestellt werden, dass es mit der richtigen Farbe eingefärbt ist.

Optimierungsvorschläge zur Darstellung der Herkunft der Studierenden aus dem Ausland finden Sie im Kapitel 7.1 „Optimierungsmöglichkeiten“.

### 6.3 Browserkompatibilität

Die Karte wurde in den Browsern „Internet Explorer“, „Mozilla“ und „Opera“ getestet und funktioniert problemlos in allen dreien. Dies entspricht den Erwartungen der Zielsetzung, da das Zielpublikum die Karte hauptsächlich mit diesen Browsern betrachten wird.

Der Aufbau der Karte in Internet Explorer dauert einiges länger als im Mozilla und im Opera. Für eine komfortable Benutzung der Karte empfiehlt es sich in jedem Fall, eine schnelle Internetverbindung zu haben.

Die Grösse aller Dateien, die für dieses Projekt benötigt werden (SVG-, JavaScript- und PHP-Dateien), können durch Zippen reduziert werden. Dies ist vor allem für die Haupt-SVG-Datei entscheidend. Deren Dateigrösse kann dadurch von 1.7 MB auf 600 KB verringert werden. Die Ladezeit der Karte verringert sich somit um einiges, vor allem, wenn es sich um einen analogen Internetanschluss über das Telefonkabel handelt.

Eine Verbesserung der Ladezeit könnte mit der Speicherung der Geometriedaten in einer Datenbank erreicht werden. So würden immer nur die Geometriedaten geladen, die gerade dargestellt werden (siehe auch Kapitel 7.2.4).

## 6.4 Kompatibilität mit SVG-Viewern und Betriebssystemen

Das Projekt wurde mit den aktuellen Adobe<sup>48</sup> SVG-Viewern der Version 3.0 und 6.0 getestet. Es funktioniert in beiden Versionen einwandfrei. Zu erwähnen ist hier, dass der SVG-Viewer der Version 3.0 im Mozilla-Browser nicht funktioniert. Für das Betrachten der Karte in Mozilla muss demnach die Version 6.0 installiert werden.

Das Projekt wurde sowohl mit Windows, als auch mit Macintosh und Linux getestet. In all diesen Betriebssystemen kann die Karte ohne Fehler in allen getesteten Browsern (Internet Explorer, Mozilla und Opera) betrachtet werden. Die einzelnen Interaktionen werden fehlerfrei ausgeführt.

## 6.5 Flexibilität des Projektes

### 6.5.1 Hinzufügen von weiteren Jahrgängen

Die Flexibilität des Projektes ist so ausgelegt, dass ohne Probleme weitere Jahrgänge in das Projekt integriert werden können. Hierfür sind folgende Anpassungen nötig:

- Integration der neuen Daten in der Datenbank
- Erweitern der Tabellen, die die Bevölkerungszahlen pro Land bzw. Bezirk für jedes Jahr enthalten. Die Tabellen können in der Excel-Datei ergänzt werden und anschliessend kann die neue Datei wieder in die Datenbank eingelesen werden.
- Erweitern des Inhalts der Selektionsliste „Jahr“ in der Datei „main.js“ um den neuen Jahrgang:

```
JavaScript-Code:  
var contentYear = new Array("1990", "1991", ..., "2004", "2005");
```

Die Übergabe des darzustellenden Jahres an die Abfrage ist so gelöst, dass der Inhalt der Selektionsliste übernommen wird. Das heisst, wenn ein neuer Jahrgang (z.B. 2005) integriert wird, werden sofort die richtigen Resultate zurückgegeben, ohne dass andere Bereiche des Skriptes abgeändert werden müssen.

---

48 PC-Softwarehersteller

### 6.5.2 Ändern der Spaltennamen der Datenbank

Da in der SQL-Abfrage direkt die Spaltennamen der Datenbanktabellen angegeben werden müssen, ist das Ändern dieser Spaltennamen in der Datenbank zu vermeiden. Änderungen hätten zur Folge, dass die Abfrage nicht durchgeführt werden kann und damit als logische Folge die Karte nicht eingefärbt würde.

Falls trotzdem einmal eine Änderung der Spaltennamen erfolgen sollte, müsste das Projekt auf die neuen Attributnamen angepasst werden. Diese Anpassung ist an sich einfach durchzuführen. Es muss lediglich darauf geachtet werden, dass kein Eintrag vergessen wird, der auf die Attributnamen verweist. Es sind dies konkret die Übergabe der Variablen in der JavaScript-Datei „main.js“ und die SQL-Befehle in den PHP-Dateien.

### 6.5.3 Ändern der Codierung der Attribute

Auch die Codierung gewisser Attribute, beispielsweise der Studienkategorie (1=Diplomstudierende, 2=Nachdiplomstudierende, 3=Doktorierende), sollte nicht verändert werden, da das Projekt auf diese zugreift. Wenn eine Änderung der Codierung durchgeführt würde, müsste auch in diesem Fall das Projekt den neuen Gegebenheiten angepasst werden. Auch in diesem Fall wären die Anpassungen mit wenig Aufwand verbunden. Es müsste lediglich die Übergabe der Variablen in der JavaScript-Datei „main.js“ geändert werden.

### 6.5.4 Änderung der Geometrie

Wenn in der Schweiz Bezirksmutationen oder in der Weltkarte Ländermutationen vorkommen, müssen neue Geometrien in das Projekt implementiert werden. Die Geometriedaten müssten in einem solchen Fall auf die gleiche Weise aufbereitet werden, wie die Geometriedaten im vorliegenden Projekt aufbereitet wurden. Konkret müssen folgende Schritte durchgeführt werden:

- Generalisierung des Weltkartendatensatzes und Aggregation der Schweizerdaten auf Bezirksebene
- Erweitern der Attributtabelle um die nötigen Attribute
- Konvertieren der Shapefiles in SVG-Dateien.

Eine Mutation der Geometrie bringt vielfach auch eine Änderung der Bezirksnummern oder der Landnummern mit sich. Falls dies zutrifft, müsste auch in der Datenbank die neue Codierung übernommen werden. Ebenfalls ist das File mit den Bevölkerungszahlen an die neue Codierung anzupassen, was wahrscheinlich den grössten Aufwand mit sich bringt.

## 7 Ausblick

In diesem Kapitel werden Gedanken dargelegt, die zur Erweiterung des vorliegenden Projektes dienen. Es werden einige Aspekte, die beim Erstellen der Karte als optimierungsbedürftig aufgefallen sind, erläutert und die Optimierungsvorschläge aufgelistet. Es werden auch Elemente der Karte aufgeführt, die aus Zeitgründen nicht mehr realisiert werden konnten.

### 7.1 Optimierungsmöglichkeiten des Projektes

Die Optimierungsvorschläge beziehen sich auf die Visualisierung der Herkunft im Ausland und auf eine allfällige Diagrammanpassung.

#### 7.1.1 Visualisierung der Auslandskarten mit Absolutwerten

Wie im Kapitel 6.2 „Resultat für die Welt und Europa“ bereits festgestellt wurde, kann die Choroplethendarstellung in der Weltkarte zu Problemen führen. Eine Lösung für dieses Problem wäre die Darstellung der Werte in absoluter Form. Das heisst die Anzahl der Studierenden wird nicht mehr durch die Bevölkerungszahl geteilt, sondern direkt in einem Kreisdiagramm dargestellt. Die Grösse der Kreise wäre abhängig von der Zahl der Studierenden. Der Vergleich zwischen dem Frauen- und dem Männeranteil könnte direkt im Kreisdiagramm dargestellt werden.

Die Visualisierung der Werte in absoluter Form bringt neue Aspekte mit sich, die bei der Choroplethendarstellung nicht berücksichtigt werden mussten. So müsste die Platzierung der Kreise gut überlegt werden. Auch ein adaptives Zooming wäre unumgänglich. Das Beispiel Liechtenstein zeigt die Problematik deutlich: Liechtensteins kleine Fläche und deren hohe Studierendenzahl würde in der Weltübersichtskarte ein grosses Kreisdiagramm generieren, dessen Linien mehrere Landesgrenzen verdecken würde, so dass diese Länder nicht mehr sichtbar wären. Die Grösse der Kreise müsste deshalb der Zoomstufe angepasst werden.

Es gibt noch weitere Möglichkeiten, dieses Problem zu lösen. Es soll hier jedoch nur darauf hingewiesen werden, dass neue Aspekte berücksichtigt werden müssten.

### 7.1.2 Modifizierung des Diagramms

Da das Umstellen der Relativwertdarstellung auf die Absolutdarstellung sehr aufwändig ist, kann auf eine einfachere Lösung des Problem zurückgegriffen werden. Diese sieht die Anpassung des Diagramms in der Welt- und Europakarte vor. Damit auch die kleinen Werte des Diagramms sichtbar werden, müsste die Höhe der Rechtecke nicht den grössten, sondern den kleinsten Werten angepasst werden. Für die wenigen Ausreisser wird eine Darstellung gewählt, die einen Unterbruch in der Rechtecksfläche erlaubt.

Abbildung 29 zeigt Möglichkeiten auf, wie solche Ausreisser einer Wertreihe dargestellt werden können:

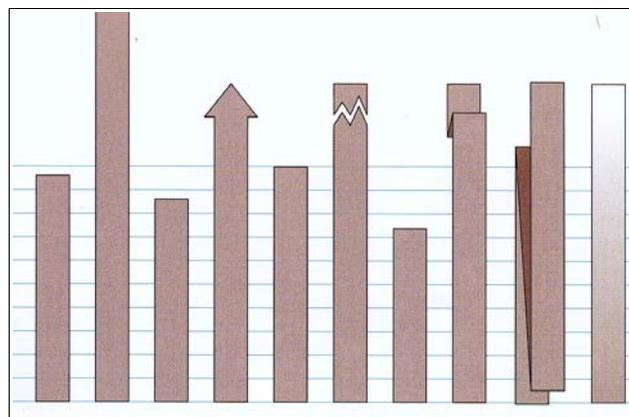


Abbildung 29: Mögliche Diagrammanpassungen  
(Quelle: [Boundford, 2000], S.68)

Um das Diagramm in dieser Form zu realisieren, müsste die Differenz der Rechteckshöhen zwischen den Ausreissern und den anderen Rechtecken berechnet werden. Wenn diese Differenz einen gewissen Betrag überschreitet, würde auf die neue Diagrammdarstellung gewechselt.

## 7.2 Erweiterung des Projektes

Die Sprachauswahl und die Animation sind Elemente der Karte, die aus Zeitgründen nicht mehr in das Projekt integriert werden konnten. Das Einbauen eines adaptiven Zoomings ist ein Element, für dessen Ausführung noch einmal beträchtlich Zeit aufgewendet werden müsste. Auch das Speichern der Geometrien in der Oracle-Datenbank würde einige Vorteile mit sich bringen.

### 7.2.1 Sprachauswahl

Die Sprachauswahl, Englisch oder Deutsch, ist ein Element, das die internationale Verwendung der Karte fördern würde. Ein Erweitern der Karte um dieses Element ist sehr zu empfehlen.

Bei der Ausführung müssten folgende Aspekte berücksichtigt werden:

- Alle Textelemente der Karte berücksichtigen
- Inhalte der Selektionslisten anpassen
- Hilfeanweisungen in der Statusbar der Navigation berücksichtigen
- Ländernamen in deutscher und englischer Ausführung
- Zweisprachiges Impressum

Zu beachten ist, dass die englischen Ländernamen bereits in der Attributtabelle der Shapefiles aufgeführt sind. Diese Tabelle müsste um ein neues Attribut ergänzt werden, das die deutschen Namen der Länder beinhaltet. Wichtig ist, dass die Attribute unterschiedliche Bezeichnungen aufweisen, damit später spezifisch auf die englischen oder die deutschen Namen zugegriffen werden kann.

### 7.2.2 Animation

Die Animation ermöglicht die Visualisierung der Herkunft über einen bestimmten Zeitabschnitt. Veränderungen können so hervorgehoben werden. Dabei soll der Benutzer das Anfangs- und das Endjahr selber bestimmen und die Animation mit einer Start-, Pause- und Stoppschaltfläche steuern können. Während der Animation kann der Benutzer die Verschiebungen der Werte in der Karte beobachten.

Ein Problem dabei könnte die Geschwindigkeit der Animation darstellen. Für jedes Jahr muss eine neue Abfrage gestartet werden, deren Resultate für die Darstellung in der Karte durch die einzelnen Funktionen des JavaScripts laufen müssen. Bei einer langsamen Internetverbindung könnte die Animation zu langsam sein, so dass dieses Tool seine Wirkung verfehlt.

### 7.2.3 Dynamische Selektionslisteninhalte

Ein Vorteil der Karte wäre das dynamische Generieren der Inhalte der beiden Selektionslisten „Studiengang“ und „Jahr“. Die Namen der Studiengänge könnten aus der Datenbank extrahiert und anschliessend in die Selektionsliste geschrieben werden. Dies hat zur Folge, dass allfällige Änderungen der Benennung der Studiengänge nur in der Datenbank vorgenommen werden und diese dann direkt in die Selektionsliste übertragen werden. Auch die Jahreszahlen der

darzustellenden Jahre könnte dynamisch generiert werden. Dies hätte den Vorteil, dass wenn die Datenbank um neue Jahrgänge erweitert wird, sich der Inhalt der Selektionsliste automatisch anpasst.

#### **7.2.4 Speichern der Geometrie in der Datenbank**

Die Speicherung der Geometriedaten in der Oracle-Datenbank würde einige Vorteile mit sich bringen: Die Grösse der Haupt-SVG-Datei würde verringert, da die Geometriedaten nicht mehr in dieser abgespeichert würden. Die Ladezeit der Karte würde sich optimieren, weil immer nur die Geometrie aus der Datenbank gelesen und in die SVG-Datei geladen würde, die der Benutzer anzeigen lassen will.

Weil der Linienverlauf der Geometrieobjekte (Länder, Bezirke, Gemeinden, etc.) ständigen Veränderungen unterliegt, müssten im Verlaufe der Zeit immer wieder Grenzmutationen in Form von beispielsweise Gemeindeaggregationen vorgenommen werden. Würden diese Mutationen ebenfalls in der Datenbank aufgenommen, könnte immer nur gerade die zum Jahrgang gehörige, aktuelle Geometrie geladen werden. Natürlich müssten auch die Dateien mit den Bevölkerungszahlen und die Sachdaten entsprechend in der Datenbank vorliegen.

#### **7.2.5 Adaptives Zooming**

Die Erstellung eines adaptiven Zoomings ist sehr sinnvoll. Bis jetzt werden die Daten in der Schweizerkarte nur auf Bezirksebene visualisiert. Ein adaptives Zooming würde die Darstellung der Daten auf Gemeindeebene ermöglichen. Je nach Zoomstufe würde die Geometrie angepasst und die Daten dementsprechend visualisiert. Die Gemeindeebene würde erst eingeschaltet, wenn der Benutzer einen bestimmten Zoomfaktor erreicht hat. Ein Überladen des Kartenbildes mit zu viel Information kann so verhindert werden.

Die Umsetzung des adaptiven Zoomings bedingt, dass alle Geometrien in die Oracle-Datenbank gespeichert werden. Ansonsten würde die Grösse der Haupt-SVG-Datei ein Ausmass erreichen, das ein effizientes Arbeiten mit der Karte verhindern würde, da die Ladezeit der Karte zu viel Zeit beanspruchen würde.

### 7.3 Entwicklungspotenzial des Projektes

Das Ziel dieser Diplomarbeit war die Visualisierung der Herkunft der Studierenden. Die Karte ist somit von einer Geometrie abhängig. Die Entwicklung eines Tools, das von der Geometrie unabhängig ist und das nur absolute Daten der ETH visualisiert, wäre vorstellbar. Die einzelnen Werte könnten in verschiedenen Diagrammen visualisiert werden. Ein Beispiel hierfür wäre der Vergleich der einzelnen Studiengänge untereinander. Die Zahlen der Studierenden verschiedener Studiengänge eines bestimmten Jahres könnten in einem Diagramm direkt verglichen werden. Die Möglichkeiten sind hier unbeschränkt und der Phantasie kann freien Lauf gelassen werden.

Die Verknüpfung des vorliegenden Projektes mit einem allfälligen neuen Projekt, das ohne Raumbezug auskommt, würde naheliegen, da beiden Projekten statistische Daten der ETH zugrunde liegen würden.

## 8 Glossar

- Apache-Server** Der Apache-Server ist ein weitverbreiteter Open-Source-Web-Server im Internet. [URL: net-lexikon]
- Array** Mit Hilfe eines Arrays können Daten eines einheitlichen Datentyps geordnet und so im Speicher eines Computers abgelegt werden, dass ein Zugriff auf die Daten über einen Index möglich wird. [URL: net-lexikon]
- Auszeichnungssprache** Eine Auszeichnungssprache dient zur Beschreibung von Daten oder des Verfahrens oder der Schritte, die zur Darstellung nötig sind. [URL: net-lexikon]
- Choroplethen** Choroplethen sind flächenhafte Darstellungen flächenbezogener quantitativer Daten. Es werden Einzelflächen mit jeweils der Fläche einheitlich zugeordneten gleichen Werten scharf voneinander abgegrenzt und durch Schraffuren, Farbabstufungen u.a. dargestellt. Die an den Flächen hängenden Sachdaten beziehen sich nicht auf einen einzelnen Ort, sondern auf eine mehr oder weniger gefasste Fläche, wie z.B. Gemeinden oder Wahlbezirke. [Bill et al., 2001]
- CSS** **Cascading Style Sheets**. 1996 eingeführt, um Formatanweisungen in HTML einbauen zu können. Vom W3C verabschiedeter Standard zur seitenabhängigen Zuweisung von Eigenschaften wie Schriftattribute und Positionierungen an HTML- oder XML-Objekte. [Bill et al., 2001]
- Datei** Unter einer Datei versteht man eine zusammenhängende, abgeschlossene Einheit von Daten, auf heutigen Systemen üblicherweise eine beliebige Anzahl eindimensionaler adressierter Bytes. [URL: net-lexikon]
- Datenbank** Eine Datenbank ist die elektronische Form eines Karteikastens. Eine Sammlung von zusammenhängenden (in Beziehung zueinander stehenden) Daten, die in einem computerunterstützten Informationssystem gespeichert ist, um eine oder mehrere Anwendungen zu bedienen, und die unabhängig von den Computerprogrammen ist, die sie verwenden. Eine Datenbank kann aus mehr als einer Datei bestehen. [Bill et al., 2001]
- GIS** Ein Geo-Informationssystem ist ein rechnergestütztes System, das aus Hardware, Software, Daten und den Anwendungen besteht. Mit ihm

können raumbezogene Daten digital erfasst und redigiert, gespeichert und reorganisiert, modelliert und analysiert sowie alphanumerisch und graphisch präsentiert werden. [Bill et al., 2001]

- HSV** Die Abkürzung HSV steht für das Farbmodell, bei dem man die Farbe mit Hilfe des Farbtons (**H**ue), Sättigung (**S**aturation) und der Helligkeit (**V**alue) definiert. Dabei wird der Farbton als Winkel auf dem Farbkreis (z.B. Rot=0°, Grün =120°, Blau=240° und die Sättigung und die Helligkeit als Prozentwert (zwischen 0 und 1) angegeben. [URL: net-lexikon]
- Internet** Das Internet ist ein weltweites Computernetzwerk einzelner voneinander unabhängiger Computer, die alle über ein System von Datenverbindungen miteinander gekoppelt sind. Es dient der elektronischen Kommunikation und dem Austausch von Informationen. Jeder Rechner des Netzes kann dabei prinzipiell mit jedem anderen kommunizieren. [URL: net-lexikon]
- Plugin** Einzelne Softwaremodule, die eine bestehende und funktionsfähige Software um zusätzliche Features erweitern und zu dieser kompatibel sind. [Bill et al., 2001]
- RGB-Farbraum** **R**ot **G**rün **B**lau ist ein additives Farbmodell, bei dem sich die Grundfarben zu weiss addieren. Eine Farbe wird durch drei Werte beschrieben: den Rot-, den Grün- und den Blauanteil. Jeder Farbanteil kann zwischen 0 und 255 variieren. [URL: net-lexikon]
- Server** Ein Server ist in der Informatik ein Dienstleister, der in einem Computersystem Daten oder Ressourcen zur Verfügung stellt. Das Computersystem kann dabei aus einem einzelnen Computer oder einem Netzwerk mehrerer Computer bestehen. [URL: net-lexikon]
- SGML** SGML ist eine Metasprache, mit deren man verschiedene Auszeichnungssprachen für Dokumente definieren kann. [URL: net-lexikon]
- Shapefile** Von der Firma ESRI eingeführtes Format für Geodaten. Neben der shp-Datei zur Speicherung der Geometriedaten wird eine .shx-Datei zur Speicherung der Indexinformation für die Geometrie und eine .dbf-Datei zur Speicherung der Sachdaten gleichzeitig erzeugt. [Bill et al., 2001]
- Skriptsprache** Skriptsprachen sind Programmiersprachen, die die Ausführung des Pro-

- grammcodes ohne getrennte Übersetzungsphase ermöglichen. Programme die in Skriptsprachen geschrieben sind, werden auch Skripte genannt. Es handelt sich immer um Textdateien oder Textfragmente, die mit einem normalen Editor bearbeitet werden können. [URL: net-lexikon]
- Tag** Neben dem eigentlichen Text enthalten z.B. XML-Dateien spezifische Markierungen, den so genannten „tags“. Die Tags werden im Quelltext des Dokumentes durch spitze Klammern markiert. Fast alle Markierungen bestehen aus einem einleitenden und einem abschliessenden Tag. Der Text dazwischen ist der „Gültigkeitsbereich“ für die betreffenden Tags. Beispiel: `<svg>...</svg>`. [Bill et al., 2001]
- Viewer** Viewer sind eigenständige Programme, die vom Webbrowser gestartet werden und in einem eigenen Fenster auf dem Bildschirm z.B. eine Animationsdatei abspielen. Viewer werden im Browser aufgerufen, sobald bestimmte Dateiformate empfangen werden, die vom Browser selbst nicht interpretiert werden können. Der Browser erkennt den Dateityp und ruft den zugehörigen Viewer auf. Allerdings müssen diese vorab installiert werden. Sie sind jedoch meist kostenfrei erhältlich. [Bill et al., 2001]
- Webbrowser** (oder auch nur Browser): sind Computerprogramme, die zum Betrachten verschiedener Inhalte benutzt werden. Sie werden vorwiegend für das Anzeigen von Websites im World Wide Web verwendet. [URL: net-lexikon]
- Webserver** Ein Webserver ist ein Server-Programm, welches Dateien über das HTTP-Protokoll zur Verfügung stellt. Jedes Mal, wenn eine Webseite angefordert wird (beispielsweise durch Anklicken eines Links), wird vom Browser eine HTTP-Anfrage an einen Webserver gerichtet. Dieser Webserver wird dann die gewünschte Seite zurücksenden. [URL: net-lexikon]
- World Wide Web** Ist ein Hypertext-System, das über das Internet abgerufen werden kann. Hierzu benötigt man einen Webbrowser, um die Daten vom Webserver zu holen und z.B. auf dem Bildschirm anzuzeigen. [URL: net-lexikon]
- WWW-Consortium** Das **World Wide Web Consortium**, oder auch W3C, ist das Gremium zur Standardisierung des Internet betreffender Techniken. [URL: net-lexikon]

## 9 Bibliographie

### 9.1 Literatur

- [Arnberger, 1997]** Arnberger, E. (1997): Thematische Kartographie. Westermann Schulbuchverlag GmbH, Braunschweig.
- [Boundford, 2000]** Boundford, T. (2000): Digital diagrams – effective design and presentation of statistical information. Watson-Guption Publications, New York.
- [Bill et al., 2001]** Bill, R., Zehner, M.L. (2001): Lexikon der Geoinformatik. Herbert Wichmann Verlag, Heidelberg.
- [Cartwright, 2003]** Cartwright, W. (2003): Maps on the Web. In M. Peterson. Maps and the internet. Elsevier Science Ltd., Oxford.
- [Hake et al., 2002]** Hake, G., Grünreich, D., Meng, L., (2002): Kartographie. Walter de Gruyter & Co., Berlin-New York.
- [Held et al., 2003]** Held, G., Schnabel, O., Neumann, A. (2003): Advanced webmapping with SVG. Proceeding SVG Open 2003, Vancouver.
- [Hurni, 2002]** Hurni, L. (2002): Thematische Kartographie. Unterrichtsunterlagen der Vorlesung GIS und Kartographie. ETH Zürich.
- [Imhof, 1972]** Imhof, E. (1972): Thematische Kartographie. Walter de Gruyter & Co., Berlin-New York.
- [Schnabel, 2004]** Schnabel, O. (2004): Web-Kartographie: Einführung in SVG: Unterrichtsunterlagen der Vorlesung Multimediakartographie, Institut für Kartographie, ETH Zürich.
- [SGK, 2002]** Schweizerische Gesellschaft für Kartographie SGK (2002): Topografische Karten – Kartengrafik und Generalisierung. Kartografische Publikationsreihe, Nr. 16. Bern
- [Sieber, 2003]** Sieber, R. (2003): Multimedia- und Webkartographie, Multimedia in Karten und Atlanten. Unterrichtsunterlagen der Vorlesung Multime-

diakartographie, Institut für Kartographie, ETH Zürich.

**[swisstopo, 2004]** Bundesamt für Landestopographie (swisstopo) (2004): Produktinformation VECTOR200, das kleinmassstäbliche digitale Landschaftsmodell der Schweiz.

## 9.2 Quellen aus dem WWW

- [URL: Adobe]** Adobe Systems Incorporated - <http://www.adobe.com/svg/viewer/install/beta.html> (23.02.2005)
- [URL: Apache]** Apache Webserver: <http://httpd.apachefriends.org/> (18.2.2005)
- [URL: BfS]** Bundesamt für Statistik – [www.statistik.admin.ch](http://www.statistik.admin.ch) (20.2.2005)
- [URL: Color-Picker]** Neumann, A.: Color-Picker-Tool - <http://www.carto.net/papers/-svg/gui/> (18.3.2005)
- [URL: ESRI]** ESRI – GIS and Mapping Software - <http://www.esri.com> (23.2.2005)
- [URL: ETH Zürich]** ETH Zürich - [www.ethz.ch](http://www.ethz.ch) (24.2.2005)
- [URL: Navigation]** Neumann, A.: Navigationstool - <http://www.carto.net/papers/svg/navigationTools/> (24.1.2005)
- [URL: net-lexikon]** Net-Lexikon – Online-Lexikon- <http://www.net-lexikon.de> (Januar – April 2005)
- [URL: Ogis2svg]** Neumann, A.: Shapefile to SVG-Converter - <http://www.carto.net/papers/svg/utills/shp2svg/> (13.1.2005)
- [URL: Oracle]** Oracle – Oracle-Instant-Client - <http://www.oracle.com/technology/software/tech/oci/instantclient/index.htm> (22.2.2005)
- [URL: PHP]** Achor, M. Betz, F., Doval, A., Lopes, N., Olson, P. et al.: PHP Handbuch - <http://www.php.net/manual/de/> (Februar – April 2005)
- [URL: Selektionsliste]** Neumann, A.: Selektionslistentool - <http://www.carto.net/papers/-svg/gui/selectionlist/> (4.2.2005)

- [URL: shp2svg] Rogers, N.: Shapefile to SVG-Converter - [http://www.carto.net/projects/shp2svg/index\\_old.html](http://www.carto.net/projects/shp2svg/index_old.html) (12.1.2005)
- [URL: TOra] TOra Oracle-Werkzeug - [http://www.sofotex.com/TOra---Toolkit-For-Oracle-download\\_L2090.html](http://www.sofotex.com/TOra---Toolkit-For-Oracle-download_L2090.html) (21.2.2005)
- [URL: U.S.C.B.] United States Census Bureau (U.S.C.B.)– Online Datenbank mit demografischen und sozio-ökonomischen Daten - <http://www.census.gov/ipc/www/idbprint.html> (21.2.2005)
- [URL: W3C] WWW-Consortium – [www.w3c.org](http://www.w3c.org)(14.2.2005)

### 9.3 Weiterführende Literatur

- Demarmels, S.** (2004): Interaktive, SVG-basierte Flugplatzhinderniskarte. Diplomarbeit, Institut für Kartographie, ETH Zürich.
- Fibinger, I.** (2002): SVG - Scalable Vector Graphics. Markt + Technik Verlag, München.
- Flanagan, D.** (1997): JavaScript – Das umfassende Referenzbuch. O'Reilly Verlag & Associates Inc., Köln.
- Gilmore, W.J.** (2001): PHP professionell – Das Handbuch für Umsteiger und Fortgeschrittene. Galileo Press GmbH, Bonn.
- Kline, K., Kline, D.** (2001): SQL In a Nutshell. O'Reilly Verlag & Associates Inc., Köln.
- Münz, St.:** SELFHTML - HTML-Dateien selbst erstellen - <http://de.selfhtml.org/> (Januar – April 2005)
- Neumann, A., Winter, A.M., et al.** - <http://www.carto.net/> (Januar – April 2005)
- Schröder, H.** (1985): Diagrammdarstellung in Stichworten. Verlag Ferdinand Hirt, Unterägeri.
- Ullrich, Th.** (2003): Dynamische Online-Visualisierung thematischer Karten im Internet. Fachhochschule Karlsruhe.
- World Wide Web Consortium:** W3 Schools – Online Tutorials - <http://www.w3schools.com/> (Januar – März 2005)

# **Anhang**

## Inhaltsverzeichnis

<b>A</b>	<b>Verwendete Instrumente</b>	<b>79</b>
A.1	XML	79
A.2	SVG	79
A.2.1	Aufbau einer SVG-Datei	79
A.2.2	SVGViewer	80
A.3	PHP	80
A.4	SQL	81
A.5	JavaScript	81
A.6	ArcGIS	81
A.7	Oracle Database	82
<b>B</b>	<b>JavaScript-Code Erläuterungen</b>	<b>83</b>
B.1	ParseXML	83
B.2	Berechnung Quantile	84
<b>C</b>	<b>Daten-CD</b>	<b>87</b>

## A Verwendete Instrumente

### A.1 XML

XML steht für eXtensible Markup Language (erweiterbare Auszeichnungssprache<sup>49</sup>). XML ist ein offener Standard des WWW-Consortiums<sup>50</sup>, welcher für die Definition, Validierung und den Austausch von Dokumentenformaten in Web entworfen wurde. XML bezeichnet einen Standard zur Definition von Auszeichnungssprachen und ist als vereinfachte Teilmenge von SGML<sup>51</sup> konzipiert worden. Ein Grundgedanke von XML ist die Trennung von Daten und Repräsentation. Weiter ist XML plattform- und softwareunabhängig, wird von allen Programmiersprachen unterstützt und ist textbasiert. Auf Grund der Textbasierung kann ein XML-Dokument nicht nur von der Maschine, sondern auch vom Menschen gelesen werden. [URL: net-lexikon]

### A.2 SVG

SVG steht für Scalable Vector Graphics und ist eine Sprache zur Beschreibung und Integration von Vektorgrafiken (aber auch Rastergrafiken und Text) für das Internet. SVG ist ein Standard des WWW-Consortiums und basiert auf XML. Das Format ist rein textbasiert und kann so in jedem Texteditor gelesen, verändert und geschrieben werden.

SVG benutzt für die Darstellung von Vektorobjekten Tags. So können beliebige zweidimensionale Grafiken gezeichnet werden. Für Standardformen wie Rechtecke oder Kreise etc. existieren vordefinierte Elemente, welchen lediglich die Grösse und die Platzierung auf dem Bildschirm zugewiesen werden muss. Weiter ist die Einbindung von Rastergrafiken, CSS, Sound, Interaktionen (z.B. mittels JavaScript), Animationen und Filtern möglich. Wie der Name schon sagt, können SVG-Dateien verschoben, vergrössert und verkleinert werden, ohne dass ein Informationsverlust eintritt. [Schnabel, 2004]

#### A.2.1 Aufbau einer SVG-Datei

SVG-Dokumente sind hierarchisch gegliedert, deren oberstes Glied die WWW-Seite ist. Die Glieder bestehen immer aus einem Start- und Endtag. Im Falle des obersten Gliedes ist dies das Dokument-Starttag <svg> und das Dokument-Endtag </svg>. Zwischen diesen Gliedern wird der Inhalt der Karte aufgeführt.

Die klare Hierarchie der Objekte ermöglicht die Ansprechbarkeit jedes einzelnen Elementes und ist somit Grundlage für die Interaktionen einer Karte. Diese Objekthierarchie nennt man

---

49 Siehe Glossar: Stichwort „Auszeichnungssprache“

50 Siehe Glossar: Stichwort „WWW-Consortium“

51 Siehe Glossar: Stichwort „SGML“

Document Object Model (DOM)<sup>52</sup>.

Der Browser liest die Inhaltselemente von oben nach unten aus und stellt sie dar. Die ersten Elemente werden also auch zuerst gezeichnet. Die Elemente selber können ebenfalls ineinander verschachtelt sein. [Schnabel, 2004]

Die Gruppierung verschiedener Elemente ist ein häufig angewendetes Verfahren, um mehrere Elemente gleichzeitig anzusprechen. Den einzelnen Elementen, aber auch den übergeordneten Gruppen können verschiedene Attribute zugewiesen werden, auf die mittels JavaScript zugegriffen werden kann.

### **A.2.2 SVGViewer**

Der Browser kann das SVG-Format nicht von sich aus interpretieren. Für die korrekte Darstellung einer SVG-Datei wird deshalb ein Plugin (z.B. SVGViewer von Adobe) benötigt, das zusätzlich installiert werden muss. Das Plugin kann kostenlos auf der Internetseite von Adobe heruntergeladen werden. Es gibt verschiedene Versionen dieses Viewers. Die aktuellste Version ist der SVGViewer 6.0. Da es sich aber um eine Version handelt, die für Entwickler vorgesehen ist, empfiehlt es sich, die Dreier-Version zu installieren. [URL: Adobe]

## **A.3 PHP**

PHP steht für PHP-Hypertext Preprocessor und ist eine serverseitige Skriptsprache, wobei es sich um OpenSource-Software handelt. PHP läuft auf vielen Plattformen (Windows, Linux, Unix, etc.) und auf fast allen Servern. PHP wurde speziell für das Erstellen dynamischer Internetseiten entwickelt und erlaubt einfache Datenbankabfragen und Dateizugriffe.

PHP muss auf einem Webserver installiert sein (z.B. Apache-Server).

Der PHP-Code kann zwar in einem normalen Texteditor geschrieben, jedoch nicht ausgeführt werden. Dazu muss die Datei auf dem Server liegen. Um die Ausgabe einer PHP-Datei zu betrachten, muss der PHP-Code in ein HTML oder SVG-Code eingebettet werden oder aber auch in einem separaten File gespeichert werden.

---

52 Siehe Glossar: Stichwort „DOM“

## A.4 SQL

SQL steht für Structured Query Language und ist eine ANSI<sup>53</sup> Standardcomputersprache für den Zugriff und die Manipulation von Datenbanksystemen. Mit SQL können verschiedene Interaktionen zwischen der Datenbank und dem Benutzer ausgeführt werden. Daten einer Datenbank können abgefragt, eingebettet, gelöscht und aktualisiert werden. [URL: W3C]

In dieser Arbeit wird SQL in das PHP-File eingebettet, um die gewünschten Daten der Oracle-Datenbank abzufragen.

## A.5 JavaScript

JavaScript ist eine von Netscape entwickelte Skriptsprache, die es erlaubt, Interaktionen in Anwendungen zu erstellen. JavaScripts sind unkomplizierte Programmanweisungen, die in einer HTML- oder SVG-Seite eingeschlossen werden und von jedem eingesehen werden können. Ohne einen zusätzlichen Serverkontakt können JavaScript-Anweisungen vom Client selbst direkt ausgeführt werden. JavaScript ermöglicht ein dynamisches Gestalten von Webseiten. Jegliche Interaktionen einer Internetseite werden mittels JavaScript ausgeführt. [Bill et al., 2001]

In SVG werden JavaScript-Skripte mit dem Tag `<script>` integriert. Sie können im SVG selber (intern) definiert werden, oder extern gespeichert werden. Bei der externen Skriptverwaltung muss jedoch im SVG auf den Speicherort und den Namen der JavaScript-Datei hingewiesen werden.

## A.6 ArcGIS

ArcGIS ist ein GIS Programm der Firma ESRI. „ArcGIS ist der Name einer Produktfamilie von ideal aufeinander abgestimmten GIS Produkten und Entwicklungswerkzeugen. Die ArcGIS-Strategie sieht vor, dass alle Daten in Zukunft in objektrelationalen Datenbanken gespeichert werden [URL: ESRI]“.

ArcGIS dient zur Darstellung und Bearbeitung räumlicher Daten und trennt dabei die Daten von ihrer Repräsentation. Ein ESRI-Shapefile besteht immer aus drei Komponenten: „.shp“ (enthält die Thematik), „.shx“ (enthält die Koordinatengeometrie der Thematik) und „.dbf“ (enthält die Attribute in Form einer dBase-Tabelle).

ArcGIS besteht aus drei Komponenten: ArcMap, ArcToolbox und ArcCatalog. ArcMap ist die zentrale Anwendung innerhalb ArcGIS. Für alle interaktiven Arbeiten mit Karten, sei es die Datenerfassung, die Analyse, die Vorbereitung der Ausgabe oder die Verknüpfung mit anderen In-

---

53 American National Standards Institute

formationen und Medien, steht ArcMap. ArcCatalog ist ähnlich wie der Windows Explorer aufgebaut und hilft die Geodaten zu strukturieren, zu verwalten und zu dokumentieren. ArcToolbox ist ein Werkzeugkasten, in dem die in ArcGIS enthaltenen Werkzeuge übersichtlich zusammengefasst und angeordnet sind. [URL: ESRI]

## **A.7 Oracle Database**

Oracle ist ein sehr leistungsfähiges, objekt-relationales Datenbankverwaltungssystem. Oracle ist eine Sammlung von Programmen, die die anwendungsunabhängige dauerhafte Speicherung von Daten in einer Datenbank ermöglicht und die damit verbundene Verwaltung übernimmt. Weitere Aufgabe von Oracle ist die Bereitstellung von verschiedenen Sichten auf die Daten (im englischen als Views bezeichnet), die Konsistenzprüfung der Daten (Integritätssicherung), die Autorisationsprüfung, die Behandlung gleichzeitiger Zugriffe verschiedener Benutzer (Synchronisation) und das Bereitstellen einer Datensicherungsmöglichkeit für den Fall von Systemausfällen. [URL: net-lexikon]

Da Oracle eine relationale Datenbank ist, werden die Daten in Form von zweidimensionalen Tabellen verwaltet, die über Schlüssel (Primärschlüssel, Fremdschlüssel) miteinander verknüpft werden können.

Zur Arbeit in Oracle wird die Transformationssprache SQL verwendet.

## B JavaScript-Code Erläuterungen

### B.1 ParseXML

In diesem Kapitel wird die Funktion „parseXML“ der Datei „main.js“ genauer erläutert, die das XML-File, das von PHP kreiert wurde, ausliest und die Inhalte in einen Array schreibt. Die Kommentare, jeweils mit „//“ gekennzeichnet, sind für die Erläuterungen der einzelnen Schritte genau zu studieren.

```
JavaScript-Code:
function parseFilestud(obj) {

//falls ein Objekt von PHP zurückgegeben wird:
if(obj.success) {

    attr = new Array();      //erstellt neues Array für Attributnamen
    var newData = parseXML(obj.content);    //parseXML-Funktion
    var root = newData.childNodes.item(0); //sucht den Kindknoten des
                                           //ersten Objektes
    var stud = root.getElementsByTagName("stud"); //sucht alle Elemente mit
                                                  //dem Tag-Name <stud>
    studanzahl = stud.length;    //Anzahl der <stud>-Elemente

// Erstellen eines Arrays mit allen Attributnamen
    for(var i=0; i<studanzahl;i++){

//falls das i-te Objekt den Knotennamen (hier <stud>) aufweist
        if(stud.item(i).nodeName){

            var attranz = stud.item(i).attributes.length; //Anzahl Attributnamen
                for(var j=0; j<attranz;j++){

//schreibe alle Attributnamen in das Array „attr“
                    attr[attr.length]=stud.item(i).attributes.item(j).name;
                }
                break;
            }
        }

//Schreibe alle zu den Attributnamen gehörigen Attributwerte in ein Array
        for (var i=0;i<studanzahl;i++){
//falls das i-te Objekt den Knotennamen (hier <stud>) aufweist
            if (stud.item(i).nodeName){
                studETH[i] = new Array();    //erstellt ein Array, das alle Attribut-
                                             //namen inklusive Werte enthält
            }
        }
    }
}
```

```
//schreibe alle Attributwerte in das Array „studETH“ mit den Attributnamen  
des Array „attr“  
    for (var j=0;j<attranz;j++) {  
  
        studETH[i][attr[j]] = stud.item(i).attributes.item(j).value;  
    }  
    }  
    }  
    } else {  
    alert('Fehler bei Verbindung zum Server');  
    }  
}
```

## B.2 Berechnung Quantile

Die Berechnung der Intervallgrenzen der Methode „Quantile“ wird im folgenden JavaScript-Code der Datei „main.js“ näher erläutert. Die wichtigste Fallunterscheidung der vorliegenden Berechnung ist das Vorhandensein eines Restwertes bei der Berechnung der Anzahl Werte, die in einer Klasse zu liegen kommen. Weitere Erläuterungen sind direkt im Text als Kommentare (mit „//“ gekennzeichnet) aufgeführt.

JavaScript-Code:

```
//Methode Quantile  
if (valcalculation == "Quantile"){  
    var nrPerClass = Math.floor(ETHquote.length / valclasses); // Anzahl  
                                                                Werte pro Klasse  
    var modVal = ETHquote.length % valclasses; //modVal = Restwert von  
                                                                nrPerClass  
  
    var bordercalc;  
    var plusVal = 0;  
  
    //Berechnung mit Rest  
    if (modVal > 0) {  
        plusVal = 1;  
  
        //speichert ersten Grenzwert in Array:  
        borderarray.push(dataminval);  
  
        //nächster Grenzwert = alter Grenzwert plus ein Restwert  
        var tempBorder = nrPerClass + plusVal;  
        var NewTempBorder = tempBorder; //Definition neuer Variable  
  
        //Durchlaufen aller Werte von ETHquote  
        for (var i=0;i<ETHquote.length;i++) {
```

```
//wenn der i-te Wert erreicht ist , nehme den Mittelwert der
angrenzenden Unter- und Obergrenze als Invertvallgrenze
if (i == NewTempBorder) {

bordercalc= ETHquote[i-1][attrquote[0]]+(ETHquote[i][attrquote
[0]]-ETHquote[i-1][attrquote[0]])/2;

//Schreibe Intervallgrenze in Array
borderarray.push(bordercalc);
modVal--; //Restwert -1

//Wenn der Rest bereits null ist
if (modVal == 0){

//nehme die normale Intervallbreite für den nächsten Wert
NewTempBorder = NewTempBorder + nrPerClass;
}

//Ansonsten
else{

//nehme die Intervallbreite plus den nächsten Restwert.
NewTempBorder = NewTempBorder + tempBorder;
}
}
borderarray.push(datamaxval);
}

//Berechnung ohne Rest (gleiches Prinzip wie mit Rest, aber ohne
Restwert)
else{
borderarray.push(dataminval);
tempBorder = nrPerClass;
NewTempBorder = tempBorder;

for (var i=0; i<ETHquote.length;i++){

if (i==NewTempBorder){

bordercalc= ETHquote[i-1][attrquote[0]]+(ETHquote[i][attrquote
[0]]-ETHquote[i-1][attrquote[0]])/2;
borderarray.push(bordercalc);

NewTempBorder = NewTempBorder + tempBorder;
}
}
}
```

```
borderarray.push(datamaxval);  
    }  
}
```

Der gesamte JavaScript-Code kann direkt in der main.js-Datei angesehen werden.

## **C      Daten-CD**