

# Diplomarbeit

## Räumliche Suche in einem internetbasierten grafischen Katalog

von Holger Schlenker

Fachhochschule Karlsruhe - Hochschule für Technik  
Fachbereich Geoinformationswesen  
Studiengang Kartographie und Geomatik  
Betreuer: Prof. Dr. Ch. Herrmann

- Fachhochschule Karlsruhe - Hochschule für Technik
- Fachbereich Geoinformationswesen
- Studiengang Kartographie und Geomatik
- Diplomarbeit: Holger Schlenker
-

## INHALTSVERZEICHNIS

<b>1 Einführung</b>	<b>7</b>
1.1 Einleitung	7
1.2 Ausführung	7
1.3 Anforderungen	7
1.4 Endprodukt	7
1.5 Zeitbedarf	8
 <b>2 Content Management</b>	 <b>9</b>
2.1 Content Management als Prozess	9
2.1.1 Planungsfaktoren	9
2.1.2 Beteiligte Gruppen	9
2.2 Funktionen von Content Management	10
2.2.1 Zugangskontrolle	10
2.2.2 Datensicherung	10
2.2.3 Multiuser-Fähigkeit	10
2.2.4 Fehlermanagement	11
2.3 Content Management Systeme	11
2.3.1 Web Content Management System	11
2.3.2 Geographisches Content Management System	12
2.4 Geographische Datenspeicher	12
2.4.1 Chorographischer Datenspeicher	12
2.4.1.1 Elektronisches Kartenarchiv	12
2.4.2 Chronologischer Datenspeicher	14
2.4.2.1 Geographisches Informations System	14
2.5 Urheberrecht	14
2.5.1 Veröffentlichungs-, Verwertungs- und Folgerecht	15
2.5.2 Einschränkungen des Urheberrechts	15
2.5.3 Verletzung des Urheberrechts	15
2.5.4 Digitale Publikationen	15
2.6 Fazit	16
2.6.1 Vorteile von Content Management	16
2.6.2 Nachteile von Content Management	16
2.6.3 Ausblick	16

<b>3 Datenbanken</b>	<b>17</b>
3.1 Von der Realität zum Modell	17
3.1.1 Geographische Koordinaten der Karten	17
3.1.2 Hauptkriterien der Karten-Recherche	17
3.1.2.1 Kartentitel	18
3.1.2.2 Herausgeber und Herausgeberort	18
3.1.2.3 Erscheinungsjahr	18
3.1.2.4 Kartenmaßstab	18
3.1.2.5 Kartenthema	18
3.1.3 Nebenkriterien der Karten-Recherche	20
3.1.3.1 Karten-ID	20
3.1.3.2 Bibliothek	20
3.1.3.3 Ausleih-Modus	20
3.1.3.4 Blattgröße	20
3.1.3.5 Sonstige Blattinhalte	20
3.1.3.6 Farben	21
3.1.3.7 Sprache	21
3.1.3.8 Sonstige Informationen	21
3.1.3.9 Bibliothekssignatur	21
3.2 Vom Modell zur Datenbank	21
3.2.1 Wahl der Datenbank	21
3.2.2 Spaltenreihenfolge und -Namensgebung	22
3.2.3 Datentypen und deren Speichergrößen	22
3.2.3.1 Ganze Zahlen	22
3.2.3.2 Gleitkommazahlen	22
3.2.3.3 Zeichenketten	23
3.3 Karten Datensatz	23
3.3.1 Quelldaten im Textformat	23
3.3.2 Beschaffenheit der Datenbank	25
3.3.3 Automatisiertes Einfügen der Daten in die Datenbank mit Hilfe eines Perl-Programmes	25
3.3.3.1 Anmelden an der Datenbank	25
3.3.3.2 Aufsplitten der Text-Datei	26
3.3.3.3 Auslesen und Überführen der Karteneinträge	27
3.3.4 Umrechnung der Koordinaten in ein metrisches System	28
3.4 Definition der Kartenfläche	31
3.4.1 Über die maximale Ausdehnung	31
3.4.2 Über die Eckpunkte	31
3.4.3 Über die Projektion	31
3.4.4 Fazit	32
3.5 Ausbaumöglichkeiten	32
3.5.1 Mehrsprachigkeit	32
3.5.2 Pixelvoransichten und digitale Rasterkarten	32
3.5.5 Datensätze	32
3.5.4 Digitale Karten	33

<b>4</b>	<b>Übersichtskarten</b>	<b>34</b>
4.1	Kartenauswahl und Format	34
4.1.1	Wahl der geeigneten Kartenprojektion	34
4.1.2	Festlegung der verschiedenen Karten	35
4.1.3	Erstellung der Übersichtskarten mit vektorbasierter Graphiksoftware	36
4.1.4	Bearbeitung der Karten im SVG-Format	36
4.2	Adaptives Zoomen der Karten	37
4.2.1	Prioritäts-Zuweisung der Ebenen in SVG	37
4.2.2	Darstellungswechsel der Generalisierungsstufen mit JavaScript	38
4.3	Wertübergabe und Umrechnung der Curserposition in Geographische Koordinaten	39
4.3.1	Voreinstellungen in JavaScript, Perl und SVG	39
4.3.2	Funktionen in JavaScript	41
4.3.2.1	Initialisierung	41
4.3.2.2	Zurückstellen bei Vergrößerung / Verschiebung	41
4.3.2.3	Umrechnung / Übergabe	41
4.4	Ausbaumöglichkeiten	42
4.4.1	Indexblätter	42
4.4.2	Interaktives Anzeigen der Kartenausmaße	42
4.4.3	Möglichkeit zum Druck	42
<b>5</b>	<b>Suchmaske</b>	<b>43</b>
5.1	Suchmaske im Allgemeinen	43
5.2	Sichtbare Oberfläche / Layout der Suchmaske	43
5.2.1	Seitenkopf	44
5.2.2	PopUp-Menüs	45
5.2.3	Übersichtskarte	45
5.2.4	Platzhalter	46
5.2.5	Trefferliste	46
5.3	Suche / Interaktion mit der DB	47
5.3.1	Aufbau der Seite mittels Perl	47
5.3.2	Deklaration und Übergabe der relevanten Variablen	48
5.3.2.2	Verbindung von JavaScript- und Perl-Dokument	48
5.3.2.3	Verbindung von Perl mit der MySQL- Datenbank	49
5.4	Ausgabe der Kartendaten	50
5.4.1	Anzeigen der Trefferliste	50
5.4.2	Detaillausgabe	51
5.5	Ausbaumöglichkeiten	52
5.5.1	Textsuche	52
5.5.2	Mehrere Sprachen wählbar	52
5.5.3	Reservation der Karten	52

<b>6 Schlusswort</b>	<b>53</b>
6.1 Erfolg / Misserfolg	53
6.1.1 Einschränkungen	53
6.1.2 Fazit	53
6.2 Unterstützung	53
6.3 Literaturverzeichnis	54
 <b>7 Anhang / Quelltext</b>	 <b>55</b>
7.1 kartensammlung.pl	55
7.2 map_search.pl	60
7.3 init.js (Auszug)	64
7.4 world.svg (Auszug)	73
7.5 navigation.svg	75

# 1 Einführung

## 1.1 Einleitung

Verteilte Datenserver, grafisch ausgelegte Webbrowser und schnelle Übertragungsraten ermöglichen heute den individuellen Zugriff auf Geodaten- und Kartenbestände, auf Geoinformations- und Kartenkonstruktionssysteme von jedem Punkt der Erde. Diese neuartigen Web.Mapping-Technologien haben zu einer unbekannten Verbreitung kartographischer Karten- und Datenbestände beigetragen. Contextabhängige Suchroutinen anhand von zum Beispiel geographischer Koordinaten und Kartenthemen helfen einen Überblick über die Karten - und Datenbestände zu behalten.

## 1.2 Ausführung

In Zusammenarbeit mit der Kartensammlung/Bibliothek der ETH Zürich/Schweiz ist ein internetbasierter grafischer Katalog, der sich zur räumlichen Suche eignet, als Prototyp zu erstellen. Im Browserfenster sollen in vektorbasierten Übersichtskarten die Karten der ETHZ-Kartensammlung mit Eckwerten und thematischen Attributen in verschiedenen Ebenen hinterlegt werden. Bei Aufzug eines Rahmens sollen die in ihm hinterlegten Karten auf einer Liste ausgegeben werden können. Der internetbasierte grafische Katalog soll als Teil einer digitalen Kartenbibliothek - also eines GCMS (Geo-Content-Management-System) - betrachtet werden können.

## 1.3 Anforderungen

Dafür sind:

- (a) ein inhaltliches Gesamtkonzept zu erarbeiten, das den Anforderungen multimedialer Informationsnutzung und -verarbeitung (klare Navigation, selbsterklärendes Interface, kurze Ladezeiten) gleichermaßen gerecht wird.
- (b) eine Auswahl von Systeminhalten vorzunehmen, diese zu bearbeiten und zu implementieren.
- (c) ein seitenorientiertes Gestaltungskonzept der Bildschirmanzeige (Screendesign) zu entwickeln und als Prototyp einer funktionsfähigen Website umzusetzen.
- (d) bei der Integration multimedialer Elemente in der Website HTML, SVG, JavaScript und Perl zur interaktiven Datenbankabfrage (Web-Server) einzusetzen. Die Schnittstelle zur vorhandenen TOPORAMA-Datenbank der ETH-Kartensammlung ist zu berücksichtigen.

## 1.4 Endprodukt

Beschreibender Text, Systemkonzeption Online, interaktiver Prototyp als CD-ROM Version und Installation der funktionsfähigen Internet-Version, aufrufbare Website, Sourcecode in Auswahl, Poster-Farbplot

- Fachhochschule Karlsruhe - Hochschule für Technik
- Fachbereich Geoinformationswesen
- Studiengang Kartographie und Geomatik
- Diplomarbeit: Holger Schlenker
- 

## **1.5 Zeitbedarf**

Dauer der Arbeit: 6 Monate

Ausgabe der Arbeit: 02.12.2002

Abgabe der Arbeit: 02.06.2003



## 2 Content Management

Content Management (CM) kann als strukturierte und systematische Beschaffung, Erzeugung, Aufbereitung, Verarbeitung und Publikation von Inhalten verstanden werden. [ROTHFUSS, RIED 2001]

### 2.1 Content Management als Prozess

Der ganze Prozess eines Content Managements beinhaltet im Prinzip mehrere verknüpfte Handlungen. Aus traditioneller Sicht können diese ganz ohne elektronische Hilfsmittel bewerkstelligt werden. Karteikartenbasierte Schlagwortkataloge haben historisch bedeutende Content Management-Aufgaben analog gelöst. Planung und Implementation sind dabei die wesentlichen Erfolgsfaktoren, die Technik übernimmt die operative Rolle. [ROTHFUSS, RIED 2001]

#### *2.1.1 Planungsfaktoren*

Der Content-Management-Prozess muss genau geplant und betreut werden. Eine Standard-Software kann im ganzen Umfang dies nicht bewerkstelligen. Die Hauptkriterien für ein gutes CM sind Effizienz und Effektivität. Sie entscheiden über Erfolg oder Misserfolg einer Lösung, die dem Problem genau angepasst werden muss. Der Blick in die Zukunft entscheidet dann über die Laufzeit und somit die Effizienz eines Systems.

Eine Analyse der Aufgabenstellung beinhaltet dann folgende Planungsfaktoren:

- die Zielvorgabe - was soll erreicht werden?
- die Prozessbeschreibung - wie soll es erreicht werden?
- die Softwareentscheidung - welche Eigenschaften muss das Software-Werkzeug haben?
- der Implementierungsplan - was muss in welcher Reihenfolge getan werden, um erfolgreich zu sein?
- die Erfolgskontrolle - woran erkennt man den richtigen Weg?
- der Produktivbetrieb - wie stelle ich sicher, dass alles qualitativ in Ordnung ist?

[ROTHFUSS, RIED 2001]

#### *2.1.2 Beteiligte Gruppen*

An einem solchen Prozess sind vereinfacht drei verschiedene Gruppen beteiligt.

- Die Autoren, Redakteure und Gestalter liefern die Inhalte und entscheiden über das gestalterische Aussehen.
- Die Programmierer sind im Vorfeld verantwortlich für die benötigten Softwarebestandteile, die Erstellung und die spätere Prozess- und Infrastruktur des ganzen Systems.
- Die Arbeit der Administratoren schlussendlich koordiniert und überwacht den laufenden Prozess. Die Administratoren kontrollieren nicht nur die technische Funktionalität, sondern auch den wirtschaftlichen Nutzen. [ROTHFUSS, RIED 2001]

## 2.2 Funktionen von Content Management

### *2.2.1 Zugangskontrolle*

Grundsätzlich ist ein Datenspeicher mit den Funktionen 'Eintrag lesen' und 'schreiben', 'bestehenden Eintrag ändern' und 'löschen' schon ein minimales Content Management. Aus heutiger Sicht ist es jedoch unverzichtbar, die Daten nur den dafür autorisierten Gruppierungen zugänglich zu machen. So wird dem Nutzer nie der gesamte Datenbestand zur Verfügung gestellt, sondern die Informationen nur nach Anfrage und Erlaubnis übermittelt. So kann zu jeder Person oder Personengruppe z.B. ein Teil der Daten zur Bearbeitung bereitgestellt werden, anderen wiederum nur zur Einsicht. Dies ist auch das Grundprinzip eines normalen Computer-Netzwerkes größerer Ordnung. Ein Schutz der Daten besteht nur wenn dieses Konzept vom Administrator konsequent eingerichtet und gepflegt wird, besteht ein Schutz der Daten.

Zu den schützenswerten Ressourcen eines Unternehmens gehören in erster Linie firmeninterne Daten jeglicher Art, IT-Infrastrukturen, Dienstleistungen oder Produktionsleistungen. [HEGERING, ABECK, NEUMAIR 1999]

### *2.2.2 Datensicherung*

So wie die Datensicherung vor Unbefugten wichtig ist, darf eine Absicherung aber auch vor physikalischen Einflüssen nicht vernachlässigt werden. Jedes Digitale System funktioniert auf unterster Ebene mit elektronischen Geräten und der dazugehörigen Stromversorgung. Diese Fehlerquellen können oft nicht beeinflusst, müssen aber einkalkuliert werden. So ist es ratsam, seine Daten auf physikalisch und örtlich voneinander getrennten Speichermedien zu sichern. Ein modernes CM implementiert mehrere Sicherungsmechanismen, die unter anderem die Speicherung von parallelen Backups zulässt.

So kann beispielsweise bei automatisiert-periodisch gemachten Sicherheitskopien ein möglicher Datenverlust begrenzt werden. Es gibt auch eine zeitversetzte Speicherung von Daten, um sich gegen Fehlmanipulation der Benutzer zu schützen. Die ausfallende Stromversorgung lässt sich durch einfache Kondensatoren lösen, die dann die Zeitspanne bis zu den anlaufenden motorbetriebenen Notstromagregaten überbrücken. Man spricht im Fachjargon von einer unterbrechungsfreien Stromversorgung (UPS). [HEGERING, ABECK, NEUMAIR 1999]

### *2.2.3 Multiuser-Fähigkeit*

Einerseits muss ein solches System gleichzeitig mehreren Mitarbeitern Daten zur Verfügung und zur Bearbeitung bereitstellen können, andererseits wiederum manche Daten sperren, die von einem User gerade verändert werden. Je nach Bedarf, Beschaffenheit und administrativen Rechten muss dies jeweils eindeutig geklärt sein. Wenn z.B. Mitarbeiter A einen Datensatz von der Content Base (Datenbank) auf seinen lokalen Rechner lädt, um diese zu aktualisieren, sollte Mitarbeiter B zeitgleich höchstens die Möglichkeit bekommen, den ursprünglichen Datensatz anzusehen, ohne diesen zu manipulieren. [HEGERING, ABECK, NEUMAIR 1999]

#### *2.2.4 Fehlermanagement*

Fehlermanagement beschäftigt sich mit dem Entdecken, Eingrenzen und Beheben von einerseits abnormalen Systemverhalten oder andererseits von nicht zueinander passenden Daten. Als Fehler kann die Abweichung von gesetzten Betriebszielen, Systemfunktionen oder Diensten definiert werden, die Fehlerquellen sind hierbei recht vielschichtig. So kann es bei der Datenübertragung (Kabel), den Netzkomponenten (Hardware) oder auch bei der Software zu Fehlern kommen, die jedoch mit verschiedensten Protokollen überwacht und auch minimal gehalten werden können.

Außerdem wird, falls es der Zusammenhang der Daten auch ermöglicht, bei der Eingabe bzw. Verarbeitung durch den Menschen ein gewisser Parameter festgelegt, der bei einer Überschreitung eine erneute Kontrolle erfordert. Ein einfaches Beispiel ist die Erstellung einer Europakarte, bei der geographische Breiten im südlichen Bereich nicht vorkommen dürfen und daher entweder automatisch oder auch von Hand korrigiert werden. [HEGERING, ABECK, NEUMAIR 1999]

### **2.3 Content Management Systeme**

Ein CM ist eine dem Problem genau angepasste Lösung, individuell und vergleichsweise aufwendig. Bei einem CMS handelt es sich um eine schon vorhandene Software, die dann die Aufgabe des CM übernimmt.

Bei einem CMS kann die Effektivität (Leistungsfähigkeit) unter dem allgemeinen Nutzen leiden. Andererseits kann die Effizienz (Wirtschaftlichkeit) durch die wesentlich geringeren Anschaffungs- und Unterhaltskosten wiederum gesteigert werden. Solche Systeme sind auch ein Zusammenspiel aus miteinander gut harmonisierenden Bausteinen (z.B. Editor, DTD-Tool, Workflow, Programmier-Komponenten, etc.), die jedoch von den Herstellerfirmen oftmals leider nur im Gesamten angeboten werden. [ROTHFUSS, RIED 2001]

#### *2.3.1 Web Content Management System*

Die Web Content Management Systeme (WebCMS) bilden mittlerweile die stärkste Gruppe aller CMS. Mit ihnen lassen sich Publikationen gerade im Internet schnell und leistungsfähig realisieren. [ROTHFUSS, RIED 2001]

Ein Außenmitarbeiter einer Zeitung kann beispielsweise über den Browser Texte, Bilder oder andere Daten durch einfaches einloggen im System auf die Datenbank überspielen. So wird den Lesern der Homepage ohne Zeitverzögerung diese Informationen zugänglich gemacht.

Der große Pluspunkt eines WebCMS ist die nicht an Rechner und Standpunkt gebundene Verfügbarkeit für den Benutzer, insofern er einen Rechner mit Internetzugang in seiner Nähe hat. Nachteil ist eine normalerweise geringe Datenübertragungsrate, die bei großen Datenmengen die Nutzbarkeit wiederum stark beeinträchtigt. [BÜCHNER, TRAUB, ZAHARADKA, ZSCHAU 2000]

### *2.3.2 Geographisches Content Management System*

Ins deutsche Übersetzt ist ein CMS nichts anderes als ein Verwaltungssystem von Inhalten. Im Falle eines Geographisches Content Management System (GeoCMS) handelt es sich um Geographische Inhalte, die nach verschiedensten Kriterien abgelegt, zusammengestellt und weiterverwendet werden können.

Am Beispiel einer benutzerdefinierten Karte lässt sich dies näher erläutern. Wie auch schon von der Herstellung einer herkömmlichen Karte bekannt, werden hier angefangen von der Topographie bis hin zu der Schrift und Thema alles getrennt voneinander abgespeichert und später nach einem Baukastensystem zu einer Einheit verschmolzen. So kann z.B. eine herkömmliche Bevölkerungsdichtekarte mit administrativen Grenzen zusammengestellt werden, erfordert jedoch der wissenschaftliche Zweck die Einwohnerdaten mit einem digitalen Höhenmodell zu vereinen, wäre dies dann auch möglich.

Ansätze zu solchen Baukastenlösungen gibt es zwar schon seit einigen Jahren mit den Geographischen Informations Systemen (GIS), jedoch immer mit der Problematik, dass es sich um Spezialsoftware handelt und die Kompatibilität der Daten darunter dementsprechend leidet.

## **2.4 Geographische Datenspeicher**

Geographische Daten können durch ihre Struktur von Grund auf verschieden sein. Zwei große Gruppierungen sind einerseits gespeicherte Karten und andererseits geographische Datensätze. Wo darin die Unterschiede liegen, folgt auf den nächsten Seiten.

### *2.4.1 Chorographischer Datenspeicher*

Der Name "chorographisch" oder auch "chorologisch" kommt aus dem Griechischen und betrifft die Raum- oder Ortswissenschaft. Demnach sind in einem Chorographischen Datenspeicher die Daten mit einer gewissen Nachbarschaftsbeziehung.

Ein Beispiel hierfür sind Karten. Gleichzeitig und nebeneinander die "Daten" wie Straßen, Namen oder Häuser auf einem Bogen Papier visualisiert. Für den Betrachter ergibt dieses Zusammenwirken einen schnellen Eindruck, die Information ist verständlich und optisch sichtbar.

Bisher war die Rede von herkömmlichen Karten, also analogen und auf Papier gedruckten. Wird eine solche Karten nun eingescannt, so ändert sich zwar das Speichermedium, die Art der Information bleibt trotzdem gleich.

#### *2.4.1.1 Elektronisches Kartenarchiv*

In diesem Kapitel geht es um Kartenarchive, also unter anderem um einen Chorographischen Datenspeicher. Grundsätzlich sollte bei der Verwendung des Begriffes "Elektronisches Kartenarchiv" unterschieden werden, ob es sich um ein "elektronisches Archiv von Karten" (a) oder ein "Archiv elektronischer Karten" (b) handelt. Beide unterscheiden sich sowohl in der Erstellung als auch im späteren Nutzen.

*(a) elektronisches Archiv von Karten*

Das "elektronische Archiv von Karten" bezieht sich auf Karten im Allgemeinen. Dabei ist es nicht Relevant, ob es sich um gedruckte oder digital vorliegende Karten handelt. Der Schwerpunkt liegt in der Archivierung, die hier elektronisch erfolgen muss. Es handelt sich also um eine Datenbank mit klar definierten Einträgen zu den Karten, also weniger über die Karteninhalte selber. Der Nutzen eines solchen Archivs hat viele bibliothekarische Gründe, wie z.B. die Verwaltung, Bestellung und Weiterführung von ganzen Kartenwerken oder auch der besseren Übersichtlichkeit und gezielteren Suchfunktion wegen. Die Möglichkeit einer schnellen Kartenrecherche stellte früher hohe Ansprüche an das Sortiersystem der Kartensammlung und die große Erfahrung der Bediensteten, da eine Suche über Karteikarten oder diverse Indexblätter nicht schnell und vor allem nicht vollständig erfolgen konnte. Diese digitale Art der Archivierung hat sich mittlerweile in allen Bibliotheken durchgesetzt, bedeutende Kartensammlungen ziehen nach. [GRELL 2001]

Die Komplexität einer solchen Datenbank ist äußerst variabel, sollte aber von Grund auf möglichst hoch sein, da sie später die eigentliche Identität einer Karte bildet. So sollte jede Umstellung bzw. Erstellung eines digitalen Systems gut geplant und vor allem mit den Anforderungen der Nutzer und der Suchfunktionen in Einklang gebracht werden, denn diese Faktoren bestimmen die Erfolgchancen des elektronischen Archivs. Oft spielt bei solchen Überlegungen die Kostenfrage eine große Rolle, da meistens auch die Kartensammlungen in ihrem Etat eng bemessen sind. Trotzdem ist mittlerweile auch hier eine elektronische Lösung den herkömmlichen analogen Archiven weit überlegen, da es viel mehr und umfangreichere Ausbaustufen zulässt und somit auch als nicht abgeschlossenes Projekt über mehrere Jahre hinweg ausgearbeitet und finanziert werden kann.

*(b) Archiv elektronischer Karten*

Ein "Archiv elektronischer Karten" ist eine Sammlung von ausschließlich digitalen Karten. Auch hier steht das Speichermedium nicht im Mittelpunkt, da elektronische Karten sowohl auf einem portablen Datenträger, wie z.B. die momentan sehr häufig benutzte CD, als auch global erreichbar auf einem Internet-Server abgelegt sind. Ein solches Archiv ist nicht zwingend ein elektronisches, wie oben schon näher erläutert, jedoch eine analoge Archivierung würde sich allein schon wegen der möglichen Interaktivität von Karte und System nicht anbieten. [GRELL 2001] Es wäre theoretisch sogar möglich, sämtliche Text- und Koordinatenangaben in den Karten in die Suche mit einfließen zu lassen, was eine direkte Beziehung zu den Karteninhalten bedeuten würde.

Diese Art von Archiven bekommen eine immer größere Bedeutung, da digitale Karten durch downloaden oder kopieren schnell, einfach und billig reproduzierbar sind. Außerdem betreten ganze Atlanten wie der "Atlas der Schweiz - interaktiv" durch ihre individuelle Darstellungsmöglichkeiten einen neuen Bereich der Kartographie, die jedoch durch die meist noch in sich abgeschlossenen Systeme nur im Ganzen bezogen und verwendet werden können. Ebenso problematisch ist hier noch die Frage nach dem Urheberrecht (siehe 2.5), da diese natürlich auch bei den digitalen Karten gelten und nicht verletzt werden dürfen.

### *2.4.2 Chronologischer Datenspeicher*

"Chronologisch" bedeutet zeitlich geordnet. Ist ein Datensatz zeitlich geordnet, so besteht nicht mehr wie bei Chorologischen Daten die Gleichzeitigkeit. Auch wenn ein Datensatz geographische Daten beinhaltet, ist dies keine Karte mehr. Nacheinander ist so zu verstehen, dass zuerst die (Lage-)Informationen zu allen Straßen kommen, danach zu allen Häusern, Namen, usw., also strikt getrennt voneinander.

Für ein Digitales System ist ein solche Trennung notwendig, damit auch eine Manipulation ganz gezielt vorgenommen werden kann. Nachteil ist aber, dass diese Daten dem Benutzer zu abstrakt sind und man im Vergleich zu den Karten keine Rückschlüsse daraus ziehen kann. Es liegt also daran, diese Daten wieder zu visualisieren.

Scalable Vector Graphics ist ein gutes Beispiel für ein System, das mit Hilfe des PlugIn im Browser aus einem inhaltlich getrennten Datensatz ein Gesamtbild präsentiert. Mehr dazu im Kapitel 4.

#### *2.4.2.1 Geographisches Informations System*

Ein Geographisches Informations System (GIS) baut genau auf diese oben erklärte Art der Datenspeicherung auf.

Ein GIS dient der Erfassung, Speicherung, Analyse und Darstellung aller Daten, die einen Teil der Erdoberfläche und die darauf befindlichen technischen und administrativen Einrichtungen sowie geowissenschaftliche, ökonomische und ökologische Gegebenheiten beschreiben. [BARTHELME 1995]

Aus diesen abstrakten Datensätzen kann dann mit Hilfe geeigneter Software wieder eine Karte erstellt werden.

## **2.5 Urheberrecht**

Da in den vorangegangenen Kapiteln mehrfach von Publikation die Rede war, sollte an dieser Stelle auch das sogenannte Urheberrecht angesprochen werden.

Dieses Recht steht dem Urheber (Autor, Verfasser, Bearbeiter, Übersetzer) eines Werkes der Wissenschaft zu. Das gegen jedermann wirkende absolute Recht an seiner geistigen Schöpfung ist im Gesetz über das Urheberrecht und verwandte Schutzrechte (Urheberrechtsgesetz) von 1965 geregelt. [MAYERS Lexikon 1997]

Bei Karten oder Plänen sind die Urheberrechte jedoch nicht ganz so einfach zu definieren. Es stellt sich immer die Frage, in wie weit die Karte eine "geistige Schöpfung" ist. Gleiches gilt für digitale Datensätze wie beispielsweise ein Digitales Höhenmodell (DHM). Man spricht auch von der "geistigen Schöpfungshöhe" eines Werkes.



### *2.5.1 Veröffentlichungs-, Verwertungs- und Folgerecht*

- Das Veröffentlichungsrecht liegt beim Urheber. Nur er bestimmt, ob, wann und wie sein Werk der Öffentlichkeit vorgestellt wird.
- Im Mittelpunkt der materiellen Interessen am Geistigen Eigentum steht das alleinige Verwertungsrecht. Das Verwertungsrecht des Autors erstreckt sich auch auf Bearbeitungen oder Umgestaltungen des Werkes (z.B. Übersetzungen).
- Ein Urheber ist vielfach weder willens noch in der Lage, seine Rechte selbst auszuwerten. Hier greift die Vermittlungsfunktion des Verlegers ein (Verlagsrecht). Will der Autor dem Vermittler ein begrenztes Werknutzungsrecht überlassen, so räumt er ihm eine begrenzte Nutzungserlaubnis, die sogenannte Lizenz, ein.

[MAYERS Lexikon 1997]

Die niedrige geistige Schöpfungshöhe eines kartographischen Produkts garantiert jedoch nicht den vollen Umfang der oben genannten Rechte. Dennoch ist es nicht erlaubt, Karten oder ähnliches ohne entsprechender Lizenz kommerziell zu verwerten oder zu vervielfältigen.

### *2.5.2 Einschränkungen des Urheberrechts*

Nach dem Ablauf einer Schutzdauer von 70 Jahren, gerechnet vom Ende des Todesjahres des Urhebers, wird sein Werk gemeinfrei. Bei öffentlichen Reden und Vorträgen, insbesondere im Parlament und vor Gericht, besteht jedoch weitgehend Wiedergabefreiheit. Auch sind Presse und Medien bei der Berichterstattung über Tagesereignisse von einer Rücksichtnahme auf etwaige Urheberrechte befreit. [MAYERS Lexikon 1997]

### *2.5.3 Verletzung des Urheberrechts*

Die Verletzung geschützter Urheberrechte gilt als unerlaubte Handlung im Sinne des bürgerlichen Rechts mit den sich daraus ergebenden Rechtsfolgen, z.B. Ansprüche auf Unterlassung, Beseitigung und Schadenersatz. [MAYERS Lexikon 1997]

Gerade bei Karten sind solche Rechtsstreitigkeiten meist problematisch, da eine Verletzung erst nachgewiesen werden muss. Manche kartographische Verlage bauen "Fehler" in ihr Produkt ein, um eine illegale Weiterverarbeitung des Werkes später vor Gericht beweisen zu können.

### *2.5.4 Digitale Publikationen*

Allein die Digitalisierung begründet nicht den Schutz als solches zugunsten desjenigen, der analoges Material digitalisiert. Für den Schutz von Multimediawerken besteht soweit es sich um eine Datenbank im Sinne der EU-Richtlinie handelt ein urheber- und nachbarrechtlicher Datenbankschutz. Eine Datenbank ist nach Art. 1 Abs. 2 der EU-Datenbankrichtlinie eine "Sammlung von Werken, Daten oder anderen unabhängigen Elementen, die systematisch oder methodisch angeordnet und einzeln mit elektronischen Mitteln oder auf andere Weise zugänglich sind".

Es sollte klargestellt werden, dass auch Datenträger der Definition der Bild- und Tonträger unterfallen. Darüber hinaus wird empfohlen, dass kombinierte Werke (z.B. Bild und Ton) in Bezug auf das Multimediawerk als Ganzes geprüft werden. Digitale Bildträger fallen schon jetzt unter die gesetzliche Definition in §16 Abs. 2 UrhG ("Vorrichtungen zur wiederholbaren Wiedergabe von Bild- oder Tonfolgen"). [DREIER 1997]

## 2.6 Fazit

### *2.6.1 Vorteile von Content Management*

Durch einen skalierbaren Ausbau eines CMS bleibt der Umfang des Systems unbegrenzt. Mit wachsenden Ansprüchen kann durch hinzufügen von neuen Komponenten der ganze Prozess erweitert werden. [BÜCHNER, TRAUB, ZAHARADKA, ZSCHAU 2000]

So können auch Daten gleichen Formats mehrere Anwendungen ermöglichen. Eine maximale Kompatibilität (z.B. zwischen Verwaltung und Produktion) kann somit erreicht werden. [ROTHFUSS, RIED 2001]

### *2.6.2 Nachteile von Content Management*

Es soll hier angemerkt sein, dass solche komplexe Systemlösungen nicht ohne ausreichender Dokumentation weiterführbar bzw. übernehmbar sind. Auch die Wartung und Weiterführung muss von geschultem und eingelerntem Fachpersonal sehr sorgfältig gewählt sein - geschieht dies nicht, so droht ein irreparables Scheitern des ganzen Systems. Die Folgen wären in der Bilanz verheerend. [ROTHFUSS, RIED 2001]

### *2.6.3 Ausblick*

Die Software-Unterstützung von Content Management-Prozessen erfordert komplexe Strukturen und Funktionalitäten. Für leistungsfähige Mechanismen braucht es ein Format, das leicht zu verarbeitende, flexible, machtvoll und gut transformierbare Strukturen besitzt. Ebenso sollte es leicht mit Datenbankstrukturen und eingeführten Programmiermethoden vereinbart und über Schnittstellen zwischen einzelner Software-Modulen transportiert werden können.

Darüber hinaus sollte die Frage geklärt werden, ob und wie Inhalte und Strukturen von Software möglichst automatisiert verarbeitet werden können, ohne dass ein Mensch verstehend und ordnend eingreifen muss. [ROTHFUSS, RIED 2001]



### 3 Datenbanken

Eine Datenbank i.A. ist heute das Kernstück jedes Content Management Systems und auch jeder Suchmaschine. Für dieses Projekt stellte sich die Frage nach dem Ausmaß, der Beschaffenheit und der Programmiertheit der Datenbank.

Der Eintrag einer Karte in der Datenbank konnte über eine Zeile definiert werden. Daher wurde hier auf eine weitaus kompliziertere relationale Datenbank verzichtet.

Eine relationale Datenbank zeichnet sich durch mehrere Tabellen aus, die getrennt voneinander existent sind, aber bei Abfragen miteinander verknüpft werden können. Je nach Umfang und Komplexität der Daten muss entschieden werden, was dem Bearbeiter als am sinnvollsten erscheint.

#### 3.1 Von der Realität zum Modell

Um eine Datenbank aufzubauen definiert man zuerst ein Abbild der realen Welt, ein sogenanntes Datenmodell. Diese Miniwelt wird gebildet aus Objekten (Entitäten), die Relationen geben die Beziehungen wieder.

Beim Errichten einer Datenbank stellen sich prinzipiell immer zwei Fragen - nach dem Was und dem Wie. Was letztendlich in einer Datenbank aufgenommen werden kann oder muss, wird vom Nutzen vorgegeben. Die Kosten-Nutzen-Frage ist normalerweise das A und O, denn nichts ist so arbeitsintensiv bzw. so teuer wie die Daten selbst.

Wie man nun die Daten in die Datenbank eingibt oder überführt, kommt wiederum auf die Beschaffenheit der Daten an. Es gilt so einfach und automatisiert wie nur möglich die Daten einzugeben. Dabei darf die Qualität dadurch nicht das gesetzte Minimum unterschreiten. Die reale Welt darf also im Datenmodell nicht zu weit verfälscht werden. [SAUER 1991]

Diese allgemeinen Gedanken zu einer Datenbank sind auch der Leitfaden für das hiesige Projekt. Es gilt eine Datenbank zu erstellen, die der Recherche nach Karten gerecht wird. Im Folgenden die verschiedenen Daten, die dazu benötigt werden.

##### *3.1.1 Geographische Koordinaten der Karten*

Geographische Koordinaten sind eindeutige Angaben und somit auch die beste Schnittstelle von der gedruckten Karte zum Suchinstrument. Bisher wurde ausschließlich, wie z.B. bei einer herkömmlichen Bibliothek mit Büchern, über Titel, Herausgeber oder ähnliche "Text"-Eingaben gesucht. Jedoch sind solche Arten von Informationen nur näherungsweise beschreibend und über die tatsächliche Gebietsausdehnung eines Kartenblattes meist nichtssagend. Es gibt aber auch hier Unterschiede, wie die Koordinaten in die Suche miteinfließen. Unter 3.4 sind diese Unterschiede noch näher erläutert.

##### *3.1.2 Hauptkriterien der Karten-Recherche*

Es ist naheliegend, dass nicht nur die geographische Ausbreitung zu einer komfortablen Kartensuche gehört, bedenkt man die vielen anderen Merkmale, die die Individualität von Karten ausmacht. Zu den Hauptmerkmalen und somit Hauptkriterien einer Recherche gehören sowohl der Titel wie auch der Maßstab, das Thema, Herausgeber und das Jahr.

Aus bibliothekarischer Sicht unerlässlich sind die Signatur und der damit verbundene Standort der Karte in der Sammlung.

#### 3.1.2.1 Kartentitel

Der Titel wird vom Herausgeber festgelegt und beschreibt meistens sowohl die Region/Land/Erdteil als auch das Thema und den Maßstab und bietet somit einen groben ersten Eindruck. Es ist aber unerlässlich, den Maßstab und Thema zusätzlich in einer eigenen Spalte abzulegen.

#### 3.1.2.2 Herausgeber und Herausgeberort

Bei den Herausgebern gibt es zwei große Gruppierungen, die staatlichen und die privatwirtschaftlichen. Fast jeder Staat dieser Erde hat die hoheitsbedingte Aufgabe sein Land kartographisch zu erfassen und auf dem neusten Stand zu halten, wobei die qualitativen (z.B. Maßstab) und quantitativen Unterschiede entweder vom Budget oder von der Größe und Besiedlungsdichte des Landes abhängen. Prinzipiell kann gesagt werden, dass von staatlicher Seite her die topographische Situation flächendeckend bearbeitet wird. Zudem sind mit nicht kommerziell tragenden Karten und Atlanten sämtliche Grundbedürfnisse abgedeckt sind.

Die Privatwirtschaftlich orientierten Kartenverlage decken die kommerziell rentable Seite der Kartographie ab. Straßen- und Freizeitkarten oder Schulatlanten sind Beispiele hierfür.

Mit der Information über den Herausgeber kann sich der Kartensuchende ein Bild davon machen, ob es sich vielleicht um einen Teil eines Kartenwerks oder um eine Einzelkarte handeln könnte.

Die Angabe des Ortes ist zu der eindeutigen Identifizierung des Herausgebers gedacht und wird immer als Zusatz mitangegeben.

#### 3.1.2.3 Erscheinungsjahr

Das Herausgabejahr ist meistens der Spiegel einer Situation zu dem momentanen Zeitpunkt (z.B. Wirtschafts-, Bevölkerungskarten), in wenigen Fällen wiederum eher nebensächlich, da sich die Inhalte nicht weiter verändert haben (z.B. Zeitonenkarten).

Da die Aktualität eine sehr bedeutende Rolle in der Auswertung von Kartenmaterial darstellt, ist das Erscheinungsjahr auch das spätere Sortierkriterium in der Trefferliste. Üblicherweise wird das bei herkömmlichen (wissenschaftlichen) Suchmasken zumindest auch so voreingestellt.

#### 3.1.2.4 Kartenmaßstab

Mit dem Maßstab kann fast immer eine mehr oder weniger genaue Aussage über den Detaillierungsgrad einer Karte getroffen werden. Auch ohne größeres Fachwissen kann sich der Benutzer dann aus eigener Erfahrung intuitiv ein Urteil über die Karte bilden und entscheiden, ob sie seinen Bedürfnissen entsprechen könnte.

#### 3.1.2.5 Kartenthema

Das Thema ist die Information über die Art der georeferenzierten Daten und demnach für wissenschaftliche Auswertungen, für die die Thematischen Karten eigentlich auch erstellt wurden, die wohl entscheidende Aussage. Georeferenziert bedeutet, dass abstrakte Daten wie z.B. das Bruttosozialprodukt sich auf eine administrative Fläche der realen Welt beziehen und somit die Informationen nicht nur als Tabelle auswerten sondern auch untereinander optisch vergleichen lassen.

Karten lassen sich nach verschiedenen Methoden klassifizieren: einerseits in drei Hauptgruppen (a) oder andererseits in neun Klassen mit zugehörigen Unterklassen (b). [ASCHE, HERRMANN 2002]

(a)

1. Karten mit allgemeinem Nutzen (z.B. Topographische Karten, Atlaskarten)
2. Karten mit speziellem Nutzen (z.B. Katasterpläne, Luftverkehrskarten)
3. Thematische Karten (z.B. Gefahrenkarten, politische Wahlergebniskarten)

(b)

1. Physikalische Umgebung: Geologie / Klima / Vegetation
2. Demographische / Kulturelle Umgebung: Bevölkerung / Bildung / Tourismus
3. Wohnen / Siedlung: Wohnen / städtische Entwicklung / Hinterland, Umland
4. Wirtschaft / Kommunikation: Bergbau / Landwirtschaft / Transport
5. Politik / Verwaltung / Geschichte: Geologie, Geographie / Verwaltung / Geschichte
6. Finanzverwaltung: Wirtschaftskraft / Verwaltungsinfrastruktur / Preiszonen
7. Umweltplanung / -Management: Umweltpotential / Regionalplanung / Spezialplanung
8. Geographische Verteilung: Naturregion / Wirtschaftsregion / Kulturregion
9. Weltraum: Planeten / Sonnensysteme / Satellitenbilder

In dem im Rahmen dieser Diplomarbeit entwickelten Prototyp einer graphischen Kartensuche musste aber auf ein schon bestehendes System der Zuordnung zurückgegriffen werden, da es die Quelldaten so verlangten. Es handelt sich um eine Einteilung in 72 alphabetisch sortierte Themengebiete. Hier ein kleiner Auszug daraus:

01. Allgemeine Atlanten
02. Alte Karten
03. Anthropologische Karten
04. Archäologische Karten
05. Architekturkarten
06. Astronomische Karten
07. Bathymetrische Karten
08. Bergbaukarten
09. Bevölkerungskarten
10. Bildhafte Kartendarstellungen
11. Bildungswesen
12. Bodenkarten
- ...
72. Zeitzonenkarten

### *3.1.3 Nebenkriterien der Karten-Recherche*

Prinzipiell kann gesagt werden, dass es sich bei den Nebenkriterien um zusätzliche Angaben handelt, die den Charakter einer Karte unterstreichen, auch wenn durch die Hauptmerkmale eine Eindeutigkeit schon gegeben ist.

Die Nebenkriterien sind in der Testdatenbank nicht alle lückenlos angegeben und z.T. in der Suchfunktion nicht mit inbegriffen. So können zum Herausgeber, der Sprache oder der Bibliothek Einstellungen gemacht werden, Farben, Blattgröße oder der Ausleihmodus sind nur informativ mit angegeben.

#### *3.1.3.1 Karten-ID*

Eine Nummer zur eindeutigen Identifikation eines Eintrags in die Datenbank ist für jedes datenbankbasierte System lebensnotwendig. Selbstverständlich braucht es auch in einer Kartensammlung eine sinnvolle Nummerierung, jedoch ist diese ID nicht mit der Bibliotheks-Signatur (3.1.3.9) zu verwechseln. Sie ist also eher systemintern und wird zur besseren Nachvollziehbarkeit mit angegeben.

#### *3.1.3.2 Bibliothek*

Dieses Kriterium ist im erstellten Prototyp momentan nebensächlich, da es sich fast ausschließlich um Karten der ETH-Kartensammlung handelt. Gedacht ist jedoch, dass weitere Kartenbestände anderer Bibliotheken integriert werden und somit gezielt in Kartensammlungen gesucht werden kann. Auch kleinere institutseigene Sammlungen können somit miteinbezogen werden. Eine bibliotheksübergreifende Lieferfunktion, wie es bei Büchern schon möglich ist, ist ebenfalls denkbar.

#### *3.1.3.3 Ausleih-Modus*

Je nach Karte oder Atlas sind verschiedene Modalitäten angegeben. So können manche Karten nach Hause genommen werden, die einen länger, die anderen vielleicht nur über Nacht. Andere wiederum stehen dem Betrachter nur an Ort und Stelle im Lesesaal zur Verfügung. Ausbaufähig wäre hier noch eine Online- Bestell- oder Reservierungsfunktion, ist aber in dieser Arbeit nicht mit implementiert.

#### *3.1.3.4 Blattgröße*

Mit dieser Angabe kann man schon einen Eindruck bekommen, ob es sich beispielsweise um eine kleine Atlasseite handelt oder um einen großen Kartenbogen. In Verbindung mit dem Maßstab kann hier bereits ein Eindruck über das Aussehen der Karte gewonnen werden.

#### *3.1.3.5 Sonstige Blatinhalte*

Hier werden Legenden, Graphiken oder Statistiken aufgeführt, die vielleicht zum Thema oder einfach nur zum näheren Verständnis erwähnenswert sind.

#### 3.1.3.6 *Farben*

Handelt es sich beispielsweise um eine schwarz/weiß-Karte, so lässt sie sich ohne größeren Qualitätsverlust mit einem normalen Kopiergerät leicht reproduzieren. Ist die Angabe von sechs oder acht Farben bei einer Karte, so ergibt dies den Rückschluss, dass es sich hierbei um ein qualitativ hochwertiges Blatt handelt.

#### 3.1.3.7 *Sprache*

Manchmal eher unbedeutend, manchmal aber unabdingbar, ist die Angabe der Sprache von der Karte. Zu bedenken ist, dass es nicht nur um Probleme beim Verständnis der Legende sondern vielmehr um die Lesbarkeit der Karte geht. So ist ein Blatt mit kyrillischer oder asiatischer Schrift für die meisten Mitteleuropäer nicht lesbar.

#### 3.1.3.8 *Sonstige Informationen*

Sonstige Informationen sind z.B. das existieren eines Begleitheftes oder die Zusammengehörigkeit von mehreren Karten in einem Kartenwerk.

#### 3.1.3.9 *Bibliothekssignatur*

Über diesen Identifikations-Code, meist eine Kombination von Buchstaben und Ziffern, lässt sich für bibliothekarisch eingelerntes Personal eine Karte im Archiv lokalisieren oder evtl. auch für Suchende selbst das gewünschte Blatt in der Sammlung finden. Eine systematische Anordnung, vielleicht nach Themen, Kartenwerken oder Ländern sortiert, ist in größeren Sammlungen unverzichtbar, um ein gewisses Maß an Überblick zu behalten.

### 3.2 **Vom Modell zur Datenbank**

Da es sich wie schon erwähnt um eine nicht-relationale Datenbank handelt, ist der Sprung vom Modell zu der endgültigen Datenbank nicht besonders kompliziert, da auf eventuelle Überschneidungen nicht akribisch geachtet werden muss. Es wird hier vom einhalten der Normalformen (NF) gesprochen. Normalformen sind bei Datenbanken Regeln, die logisch richtige Abfragen sichern. So darf in einer Zelle beispielsweise nur ein Wert stehen (2.NF), der sich zudem nicht durch weitere errechnen lässt (3.NF).

Außerdem muss darauf geachtet werden, dass eine Tabelle immer einen fest strukturierten Aufbau aus Zeilen (Datensätze) und Spalten hat. Der Aufbau wird einmal beim Anlegen der Tabelle definiert und ändert sich während der gesamten Lebensdauer nie. [SAUER, Herrmann 1991]

#### 3.2.1 *Wahl der Datenbank*

Bevor die Datenbank erstellt wird, sollte natürlich entschieden werden, welche Software dafür verwendet wird.

Von Microsoft wird beispielsweise Access angeboten, eine Datenbank mit eigener Abfragesprache. Theoretisch hätte dies auch ausgereicht, zeigt aber Schwächen beim Export oder der Sicherheit des Zugriffs.

Datenbanken wie MySQL, Postgres oder Oracel werden von SQL unterstützt. SQL ist eine vergleichsweise mächtigere Abfragesprache. Zum Beispiel können damit auch größere Datenmengen gespeichert werden. Auf MySQL-Datenbanken kann mit einer DBD/DBI (SQL-Interface) zugegriffen werden. Ein weiterer Vorteil ist, dass Script-Sprachen (z.B. Perl oder PHP) eine Schnittstelle bilden können, also Aufrufe daraus gemacht werden können. Hier fiel die Wahl auf eine MySQL-Datenbank, die Abfragen dazu stammen aus dem Perl-Script.

### *3.2.2 Spaltenreihenfolge und -Namensgebung*

Eine sinnvolle Reihenfolge der einzelnen Spalten ist eher der Übersichtlichkeit dienlich als für das System notwendig. Inhaltlich zusammengehörende Spalten sollten nebeneinander liegen, angefangen von den Hauptkriterien (s. 3.1.2) bis hin zu den Nebenkriterien (s. 3.1.3).

Ebenfalls sollte darauf geachtet werden, dass die Namensgebung eindeutig und vor allem syntaktisch richtig ist. Groß- und Kleinschreibung kann gezielt eingesetzt werden. Auf Sonderzeichen sollte wenn möglich verzichtet werden, da dies für die Abfragesoftware eine Fehlerquelle darstellen kann. Die Abfragesoftware kann ganz empfindlich auf solche kleine Unterschiede reagieren und die Suche nach solchen Syntaxproblemen ist sehr mühsam und zeitraubend.

### *3.2.3 Datentypen und deren Speichergrößen*

MySQL arbeitet wie einige andere Programmiersprachen auch mit Datentypen, die im Vorfeld schon angegeben werden müssen. Mit diesen Angaben legt man bei Numerischen Formaten den Wertebereich fest, bei Zeichenketten die Länge (Anzahl der Zeichen). Dies hat den Sinn, Spalten nicht unnötig viel Speicherplatz zuzuweisen und somit das System schneller und platzsparender zu machen. [MOOS, DAUES 1991]

#### *3.2.3.1 Ganze Zahlen*

Für Ganze Zahlen bietet sich das INTEGER-Format an. Elf Zeichen (inkl. Vorzeichen) und einen Wertebereich von -9 999 999 999 bis 99 999 999 999 ist für die meisten Nummerierungen und Maßstabszahlen völlig ausreichend. [MOOS, DAUES 1991]

#### *3.2.3.2 Gleitkommazahlen*

Gleitkommazahlen werden in der Regel mit dem Datentyp FLOAT angegeben, da dieser einen theoretischen Wertebereich von  $-0.1 \cdot 10^{307}$  bis  $0.9 \cdot 10^{308}$  hat. Der zu benötigende Speicherplatz wird durch 2 Werte angegeben, die erste Zahl gibt die gesamte Ziffern-Länge inkl. Vorzeichen und Dezimalpunkt an, der zweite Wert die Nachkommastellen. Mit "FLOAT (7,2)" können dann beispielsweise Zahlen von -999.99 bis 9 999.99 gespeichert werden. [MOOS, DAUES 1991]

#### 3.2.3.3 Zeichenketten

Zeichenketten, auch Strings genannt, werden im CHAR-Format abgelegt. CHAR steht für Character (Buchstabe) und bietet maximalen Platz für 255 Zeichen. Die benötigte Länge wird ebenfalls mit einer Zahl in Klammern angegeben. "CHAR (20)" erlaubt also nur eine Gesamtlänge von 20 Buchstaben, Ziffern oder Sonderzeichen. Mit im CHAR-Format abgespeicherten Zahlen kann jedoch nicht gerechnet werden, lediglich verglichen. [MOOS, DAUES 1991]

### 3.3 Karten Datensatz

#### 3.3.1 Quelldaten im Textformat

Von der ETH-Kartensammlung konnte aus einem schon bestehenden Datenbanksystem die Kartendaten als Text-Datei exportiert werden. Diese auf lokalen Rechnern in der Kartensammlung laufende TOPORAMA-Datenbank basiert auf der Software des GIS-Herstellers ESRI. Die Text-Datei war dann auch die Grundlage für das hier beschriebene Suchsystem, musste aber noch in die eigens erstellte MySQL-Datenbank überführt werden.

Auf der folgenden Seite zeigen wir einen Auszug der Textdatei, die wir von der ETH-Bibliothek erhalten haben und erläutern ihn.



```

000445137 FMT      L MP
000445137 LDR      L 01650nem--2200337uu-4500
000445137 003      L ETHICS
000445137 005      L 19880218000000.0
000445137 007      L aj|||||
000445137 008      L 880218s1970----sz-u---|-----|00|||ger|d
000445137 0341     L $$aa$$b300000$$dE0055700$$eE0103000$$fN0474800$$gN0454800
000445137 040      L $$aETH-BIB$$bger$$cETH-BIB$$eETHICS-ISBD
000445137 041      L $$agerfre
000445137 090      L $$a000479273$$bBHR$$n2
000445137 090      L $$a83065318$$bMIKAS
000445137 099      L $$aKONV$$c19880218$$lE01$$h10:15:18
000445137 245      L $$aKARTE DER MASSGEBENDEN GRUNDWASSERLEITER MIT DEN WICHTIGSTEN
    FASSUNGS- UND ANREICHERUNGSGEBIETEN 1:300 000$$hKartenmaterial$$b[SCHWEIZ]$$dCARTES
    DES COUCHES AQUIFERES AVEC INDICATION DES PLUS IMPORTANTES ZONES DE CAPTATION
    ET D`ENRICHISSEMENT DES NAPPES SOUTERRAINES$$b[SUISSE]$$bTeilleitbild der
    Siedlungswasserwirtschaft$$cAlfred Werner, Karl Tschudin; ORL, ETH-Zürich &
    Eidg.Amt für Gewässerschutz, Bern
000445137 250      L $$aAusg.1970
000445137 255      L $$a1:300 000$$cE 05°57-E 10°30/N 47°48-N 45°48
000445137 260      L $$aZürich$$bInstitut für Orts-, Regional- und Landesplanung$$c1970
000445137 300      L $$a1 Karte$$c80 x 120 cm
000445137 500      L $$aQuellen: Kantonale und kommunale Amtsstellen
000445137 691E1    L $$aHYDROLOGISCHE, HYDROGRAPHISCHE, HYDROGEOL. KARTEN$$9ger$$8912%556
000445137 691E1    L $$aHYDROGEOLOGIE + GRUNDWASSERHYDROLOGIE$$9ger$$8556.3
000445137 691E1    L $$aSCHWEIZ (MITTELEUROPA). SCHWEIZERISCHE EIDGENOSSENSCHAFT$$9ger$$8(494)
000445137 691E1    L $$aSIEDLUNGSKARTEN (WASSER, KEHRRICHT, LICHT)$$9ger$$8912%628
000445137 691E1    L $$aWASSERAUFBEREITUNG + WASSERVERSORGUNG + WASSERNUTZUNG$$9ger$$8628.1
000445137 691E1    L $$aWASSERKREISLAUF + WASSERHAUSHALT DER ERDE$$9ger$$8556.1
000445137 700      L $$aWerner, A.
000445137 700      L $$aTschudin, K.
000445137 906      L $$cCM Karte = Carte
000445137 CAT      L $$aCONV$$b00$$c19990819$$lEBI01$$h1101
000445137 CAT      L $$c20000930$$lEBI01$$h0248
000445137 CAT      L $$c20010626$$lEBI01$$h2102
000445137 Z30-1    L $$lE01$$2KS$$3K 680221: ED.1970$$f41$$Fkeine Ausleihe
000445137 691E1    L $$aHYDROGEOLOGICAL MAPS + HYDROLOGICAL MAPS + HYDROGRAPHIC
    MAPS$$9eng$$8912%556
000445137 691E1    L $$aCARTES HYDROGEOLOGIQUES, HYDROLOGIQUES,
    HYDROGRAPHIQUES$$9fre$$8912%556
000445137 691E1    L $$aSUISSE (EUROPE CENTRALE). CONFEDERATION SUISSE$$9fre$$8(494)
000445137 691E1    L $$aSWITZERLAND (CENTRAL EUROPE). SWISS CONFEDERATION$$9eng$$8(494)
000445137 691E1    L $$aSETTLEMENT INFRASTRUCTURE (MAPS)$$9eng$$8912%628
000445137 691E1    L $$aCARTES/INFRASTRUCTURE DE BASE (DISTRIBUTION D EAU, ORDURES,
    COURANT)$$9fre$$8912%628
000445137 691E1    L $$aWATER SUPPLY + WATER TREATMENT + WATER UTILIZATION$$9eng$$8628.1
000445137 691E1    L $$aAPPROVISIONNEMENT EN EAU, TRAITEMENT, UTILISATION$$9fre$$8628.1
----1----- --2-- -----3-----

```

-1- ist die Karten-ID. Jeder darauf folgende Eintrag einer Zeile gehört zu der Karte mit dieser ID. Zum besseren Verständnis können die Einträge mit der gleichen Karten-ID später als eine Zeile in der Datenbank gesehen werden.

-2- ist der Schlüssel oder auch die Leitzahl, die durch ihre unterschiedlichen Ziffern quasi als Spaltenangabe zu verstehen ist. Welche Information sich hinter welchem Schlüssel verbirgt, ist in der Tabelle unter 3.3.2 aufgeführt.

-3- sind die eigentlichen Einträge, meist beginnend mit "\$\$" und anschl. Buchstaben oder Zahl. Es ist ein weiterer Schlüssel zur Angabe der Daten.



### 3.3.2 Beschaffenheit der Datenbank

Sind nun sämtliche Faktoren wie Namensgebung, Datentyp und Quelldaten berücksichtigt, ergibt sich diese Struktur der Datenbank:

Name	Format	Leitzahl	Beispiel von 3.3.1 (z.T. schon umgewandelt)
karten_id	Int (10)		000445137
karten_titel	Char (255)	245 \$a	KARTE DER MASSGEBENDEN GRUNDWASSERLEITER MIT DEN WICHTIGSTEN FASSUNGS- UND ANREICHERUNGSGEBIETEN 1:300 000
herausgeber	Char (255)	260 \$b	Institut für Orts-, Regional- und Landesplanung
herausgeber_ort	Char (100)	260 \$a	Zürich
jahr	year (4)	260 \$c	1970
massstab	Int (10)	255 \$a (ohne 1:)	300 000
thema	Char (255)	691E1 \$\$a	HYDROLOGISCHE, HYDROGRAPHISCHE, HYDROGEOL. KARTEN
west	Float (8.3)	255 \$c	5.95
ost	Float (8.3)	255 \$c	10.50
nord	Float (8.3)	255 \$c	47.80
sued	Float (8.3)	255 \$c	45.80
signatur	Char (15)	Z30-1 \$3	K 680221
bibliothek	Char (25)	040 \$a	ETH-BIB
ausleihe	Char (50)	Z30-1 \$F	keine Ausleihe
blattgroesse	Char (30)	300 \$c	80 x 120 cm
sonst_blattinhalte	Char (255)	300 \$e	-
farben	Char (100)	300 \$b	-
sprache	Char (10)	008 (Ende)	ger
sonst_info	Char (255)	500 \$a	Quellen: Kantonale und kommunale Amtsstellen

[TABELLE 3.3.2: Datenbank-Struktur]

### 3.3.3 Automatisiertes Einfügen der Daten in die Datenbank mit Hilfe eines Perl-Programmes

In den folgenden Kapitel ist schrittweise aufgeführt, wie ein Perl-Programm diese Textdatei im ASCII-Format systematisch bearbeitet und die Einträge in die richtige Spalte überführt. Bedenkt man die große Anzahl der in der Text-Datei aufgenommenen Karten (ca. 10 000) wird klar, wie wichtig eine Automatisierung für diese Testdatenbank wurde.

#### 3.3.3.1 Anmelden an der Datenbank

Nachdem die SQL-Datenbank mit dem Namen "karten2" erstellt wurde, meldet sich zu aller erst das Perlscript dort an. Das ist notwendig, schließlich sollen die Daten in dieser dann auch gespeichert werden. Wie schon in 3.2.1 erwähnt, kann mit Perl eine Schnittstelle von Text- und Datenbank geschaffen werden.

### 3.3.3.2 Aufsplitten der Text-Datei

- Im ersten Schritt liest das Programm die Text-Daten ein (Zeile 48) und beginnt mit der Aufteilung einer Zeile (ab Zeile 51).
- So gibt es einen Teil vor [0] und nach [1] den doppelten Dollarzeichen (Zeile 54).
- Der vordere Teil davon wird an den Leerzeichen nochmals gesplittet und unter dem Variablennamen "preamble" gespeichert (Zeile 55).
- "preamble [0]" ist dann die Karten-ID (Zeile 58)
- Im Anschluss wird die ID an die Datenbank übergeben (Zeile 68).

Im obigen Beispiel wäre das dann die '000445137'

```

46      #----- lese aus ascii dokument -----
47
48      open(DAT, $datenfile) or die("Could not open file!");
49      @rohdaten=<DAT>;
50
51      foreach $linie (@rohdaten) {
52          #----- aufteilen der linie
53          chop($linie);
54          @nachdollar = split(/\$\$/, $linie);
55          @preamble = split(/ /, $nachdollar[0]);
56          @nachbalken = split(/\|\\|\\|\\|/, $linie);
57
58          $karten_id = $preamble[0];
59
60          #----- schauen, ob ein eintrag existiert
61          $sth = $dbh->prepare("SELECT ('karten_id') FROM karten2 WHERE (karten_id
= '$karten_id') ");
62          $resp=$sth->execute;
63          $stable = $sth->fetchall_arrayref;
64
65          $counter=0;
66          for $i ( 0 .. $#{$stable} ) { $counter++;}
67          if($counter==0){
68              $src = $dbh->do("INSERT INTO karten2 (karten_id) VALUES ('$karten_id') " )
or die "Problems with karten_id $karten_id entry...";
69          }
70
71          #----- falls nein, eintrag erstellen
72

```

Auszug aus "kartensammlung.pl"

### 3.3.3.3 Auslesen und Überführen der Karteneinträge

- "preamble [1]" ist die schon mehrfach erwähnte Leitzahl, die die Art der folgenden Daten angibt. "Z30-1" steht für die Signatur in der Kartensammlung, also die Bibliotheks-Standortnummer (Zeile 77).
- Da aber mehrere Einträge mit diesem Schlüssel verbunden sind, muss noch die direkte Zahl nach dem "\$\$" berücksichtigt werden, hier die "3" (Zeile 80).
- Beim Beispieleintrag wäre das die 'K 680221', die in der Datenbank in die Spalte "signatur" gespeichert wird (Zeile 84).

```

73      #----- untersuchen der leitzahlen
74
75      #----- signatur -----
76
77      if($preamble[1] eq 'Z30-1') {
78          foreach $wort (@nachdollar) {
79              $ifwert = substr($wort, 0, 1);
80              if($ifwert eq '3') {
81                  $laenge = length($wort);
82                  $signatur = substr($wort, 1, $laenge);
83                  #----- einfüegen in datenbank -----
84                  $src=$dbh->do("UPDATE karten2 SET signatur = '$signatur' WHERE karten_id
= '$karten_id' " ) or die "Problems with signatur $signatur entry";
85
86              }
87          }
88      }
Auszug aus "kartensammlung.pl"

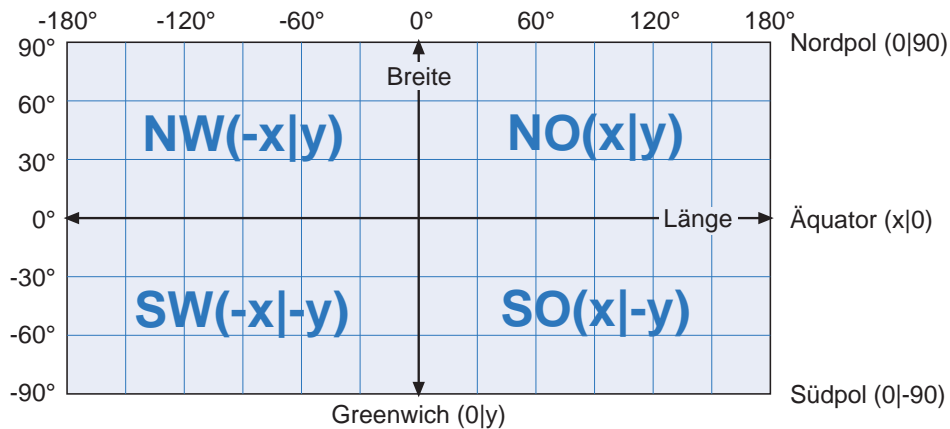
```

Nach diesem Prinzip konnten auch sämtliche weitere Einträge (z.B. Thema, Maßstab, etc.) ausgelesen und ordnungsgemäß übergeben werden. Somit wurde die ganze Datenbank aufgebaut. Angesichts der recht großen Datenmenge von 45 MB blieb die Fehlerquote relativ gering und nahezu 10 000 reell in der Kartensammlung existente Karten standen mit 19 Spalten der späteren Suche zur Verfügung.

### 3.3.4 Umrechnung der Koordinaten in ein metrisches System

Etwas Problematisch gestaltete sich das Einlesen der Koordinaten. Hinter der Leitzahl "255 \$\$\$" steckt die geographische Angabe der Karte. Bei den Koordinaten wurden nicht die Eckkoordinaten sondern die maximale Ausbreitung (siehe 3.4.1 und 3.4.2) erfasst. Die Zahlen sind nicht in einem metrischen Koordinatensystem (siehe Abb. 3.3.4). In dem unter 3.3.1 gezeigtem Beispiel sieht der Eintrag so aus: 'E 05°57'-E 10°30'/N 47°48'-N 45°48'.

Das Ziel war es, die Koordinatenangaben in ein Format zu bringen, das eine Suche ermöglicht, damit also wie folgt die einzelnen Werte in die Spalten übergeben werden: west: '5.95' / ost: '10.50' / nord: '47.80' / sued: '45.80'. Unter Tabelle 3.3.4 dann die genauere Erläuterung zeilenweise



[ABBILDUNG 3.3.4: Koordinatensystem]

```

182 #----- koordinaten -----
183
184 if($preambel[1] eq '255') {
185     foreach $wort (@nachdollar) {
186         $ifwert = substr($wort, 0, 1);
187         if($ifwert eq 'c') {
188             $laenge = length($wort);
189             $koordinaten = substr($wort, 1, $laenge);
190             @richtung = split(/\//,$koordinaten);           # trennt 'koordinaten'
in länge und breite
191             @laengengrade = split(/\-/,$richtung[0]);       # trennt 'länge' in
von und bis
192             @breitengrade = split(/\-/,$richtung[1]);       # trennt 'breite' in
von und bis
193 #----- west -----
194             @teil_west = split(/ /,$laengengrade[0]);       # trennt 'von länge'
in vorzeichen und werte
195             @koord_west = split(/\°/,$teil_west[1]);        # trennt 'von
länge werte' in grad und minuten
196             $west = $koord_west[0] + ($koord_west[1]/60);    #
verrechnet 'von länge grad' und 'von länge minuten'
197             if($teil_west[0] eq 'W') {                       # setzt
vorzeichen
198                 $west = $west*(-1);
199             }
200             $src=$dbh->do("UPDATE karten2 SET west = '$west' WHERE karten_id =
'$karten_id' " ) or die "Problems with west $west entry";
201
202 #----- ost -----
203             @teil_ost = split(/ /,$laengengrade[1]);        # trennt 'bis länge'
in vorzeichen und werte
204             @koord_ost = split(/\°/,$teil_ost[1]);           # trennt
'bis länge werte' in grad und minuten
205             $ost = $koord_ost[0] + ($koord_ost[1]/60);       # verrechnet 'bis
länge grad' und 'bis länge minuten'
206             if ($teil_ost[0] eq 'W') {                       # setzt
vorzeichen
207                 $ost = $ost*(-1);
208             }
209             $src=$dbh->do("UPDATE karten2 SET ost = '$ost' WHERE karten_id =
'$karten_id' " ) or die "Problems with ost $ost entry";
210
...

```

Auszug aus "kartensammlung.pl"

Schritt	Zeile	Erklärung	Ergebnis
1	184 - 188	Einlesen der Koordinaten bei der Leitzahl "255" und "\$\$3"	'E 05°57'-E 10°30'/N 47°48'-N 45°48'
2	190	Trennung von "koordinaten" in [0] ("laengengrade") und [1] ("breitengrade") bei "/"	'E 05°57'-E 10°30' und 'N 47°48'-N 45°48'
3	191	Trennung von "laengengrade" (= "koordinaten [0]") in [0] ("teil_west") und [1] ("teil_ost") bei "-"	'E 05°57' und 'E 10°30'
4	192	Trennung von "breitengrade" (= "koordinaten [1]") in [0] ("teil_nord") und [1] ("teil_sued") bei "-"	'N 47°48' und 'N 45°48'
5	194	Trennung von "teil_west" (= "laengengrade [0]") in [0] (Vorzeichen) und [1] ("koord_west" = Werte) bei " "	'E' und '05°57'
6	195	Trennung von "koord_west" (= "teil_west [1]") in [0] (Grad) und [1] (Minuten) bei "°"	'05' und '57'
7	196	Errechnung von "west" mittels Addition von Grad ("koord_west [0]") und Minuten (in Dezimalform) ("koord_west [1] / 60")	'05.95'
8	197 - 199	Setzen des Vorzeichens: nur wenn "teil_west [0]" = "W" Multiplikation von "west" mit "-1", sonst keine Veränderung	'05.95'
9	200	Übergabe von "west" an die Datenbank ("karten2") in die Spalte "west"	
10	203	Trennung von "teil_ost" (= "laengengrade [1]") in [0] (Vorzeichen) und [1] ("koord_ost" = Werte) bei " "	'E' und '10°30'
11	204	Trennung von "koord_ost" (= "teil_ost [1]") in [0] (Grad) und [1] (Minuten) bei "°"	'10' und '30'
12	205	Errechnung von "ost" mittels Addition von Grad ("koord_ost [0]") und Minuten (in Dezimalform) ("koord_ost [1] / 60")	'10.50'
13	206 - 208	Setzen des Vorzeichens: nur wenn "teil_ost [0]" = "W" Multiplikation von "ost" mit "-1"; sonst keine Veränderung	'10.50'
14	209	Übergabe von "ost" an die Datenbank ("karten2") in die Spalte "ost"	
...	210 - 230	Umrechnung und Übergabe von "nord" und "sued" nach gleichem Prinzip	

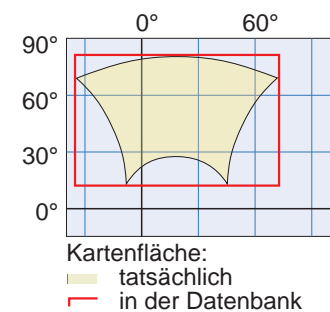
[TABELLE 3.3.4: Schrittweise Umrechnung und Übergabe der Koordinaten]

### 3.4 Definition der Kartenfläche

In der Kartennetzlehre beschäftigt man sich mit dem Problem der Projektion des dreidimensionalen Raums auf eine zweidimensionale Fläche. Es ist problematisch, das auf einer meist rechtwinkligen Karte abgebildete Gebiet durch einen einfachen, mathematischen Rahmen zu definieren.

#### 3.4.1 Über die maximale Ausdehnung

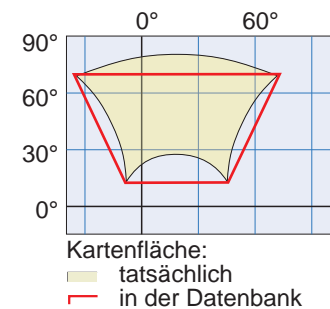
In der von der ETH vorgenommenen digitalen Kartenerfassung hat man sich bei den Karten auf die maximale Ausbreitung beschränkt. So wurde nur der nördlichste, südlichste, östlichste und westlichste Punkt der Karte erfasst. Dadurch entsteht ein Rechteck, das nur näherungsweise dem tatsächlichen Kartengebiet entspricht. Das hat aber den Vorteil, dass vier Werte (maximale und minimale Breite und Länge) genügen (siehe Abbildung 3.4.1). Problematisch ist dies bei sehr kleinmaßstäbigen Azimutalprojektionen und bei nicht genordneten Karten. Für die Kartensuche im graphischen Katalog erwiesen sich diese Angaben aber als völlig ausreichend.



[ABBILDUNG 3.4.1: Kartenfläche über maximale Ausdehnung]

#### 3.4.2 Über die Eckpunkte

Wird die Kartenfläche über ihre Eckpunkte angegeben, so entsteht bei der Verbindung ein meist nicht rechtwinkliges Viereck, sodass das tatsächliche Gebiet bei den meisten Karten schon ziemlich genau diesem Rahmen entspricht. Dieses Verfahren hat jedoch den Nachteil, dass es insgesamt acht Werte erfordert (4 Eckpunkte, jeweils mit Länge und Breite) (siehe Abbildung 3.4.2), was sowohl bei der Erfassung als auch bei der weiteren Verarbeitung einen Mehraufwand bedeutet. Das zieht dadurch auch eine größere Fehlerquote nach sich.



[ABBILDUNG 3.4.2: Kartenfläche über Eckpunkte]

#### 3.4.3 Über die Projektion

Wird die Kartenfläche über die Projektion beschrieben, so bekommt man das mathematisch wirklich korrekte Ergebnis. Der "Preis" ist jedoch meist eine eher komplizierte Formel. Für den zu erzielenden "Nutzen", hier ein Suchsystem, ist das viel zu hoch.

#### *3.4.4 Fazit*

Der Weg über Absolutwerte, also Koordinaten, ist wesentlich einfacher und verständlicher. Sowohl für den Datenerfasser als auch den für Programmierer ist eine einheitliche Aufnahme der Koordinaten notwendig. Bei großen Maßstäben ist die Abweichung bei der maximalen Ausbreitung recht gering. Bedenkt man, dass der größte Teil aller Karten der ETH-Bibliothek einen Maßstab von 1 : 500 000 und größer haben, fällt die Wahl schnell auf die einfachste Variante. Der nördlichste, südlichste, westlichste und östlichste Wert auf der Karte ist schnell gefunden und eingegeben.

#### *3.5 Ausbaumöglichkeiten*

Diese vorangegangenen Überlegungen zu der Datenbank einer Suchmaschine sind nicht nur theoretisch, sondern auch praktisch noch vielseitig erweiterbar. Inspirativ kommen nun noch einige Punkte, welche Features oder Möglichkeiten zum Ausbau dieses Suchsystem noch hinzugefügt werden können. Auch im Hinblick auf ein modernes GeoCMS sind Erweiterungen möglich.

##### *3.5.1 Mehrsprachigkeit*

Da die Datenbank automatisiert aus der Textdatei aufgebaut wurde (siehe 3.3), können ohne größeren Aufwand noch weitere Informationen aus dem vorhandenen Datensatz geholt werden. Es wurde beispielsweise der Titel und das Thema meist noch in englischer und französischer Sprache aufgenommen. Somit könnte selbst die Suchmaske multilingual gestaltet oder besser noch wählbar gemacht werden.

##### *3.5.2 Pixelvoransichten und digitale Rasterkarten*

Es wäre denkbar, von den Karten eine gescannte oder abfotografierte Pixelversion auf einem Bilddaten-Server abzulegen. So könnte sich der Benutzer durch grob aufgelöste Teilbereiche einen schnellen Eindruck über das tatsächliche Aussehen der Karte machen. Bei einer hohen Auflösung und guter Datenübertragungsrate wäre es möglich, dass man eine druckfähige Version der Karte herunterladen darf.

##### *3.5.5 Datensätze*

Datensätze wie beispielsweise ein Digitales Höhenmodell (DHM) oder aus der Statistik könnten in einem immer aktuellen Zustand den Nutzern zugänglich gemacht werden. Erforderlich hier aber einiges an Fachkenntnis, da solche Datensätze meist in einem Rohdaten-Format sind und erst mit geeigneter Software visualisiert werden können. Der Vorteil wäre aber die Variabilität, da man ähnlich eines Baukastensystems nur die tatsächlich benötigten Daten bekommt und auf Nutzloses verzichten kann.



#### *3.5.4 Digitale Karten*

Einen Schritt weiter wären schon fertige digitale Karten in einem Vektorformat. So könnten qualitativ hochwertige Karten mit einem recht geringen Speicherplatzbedarf und somit guten Download-Eigenschaften der Masse zur Verfügung gestellt werden.

Geeignete Formate zur Weiterverarbeitung sind momentan noch Adobe Illustrator (AI) oder Macromedia Freehand (FH), zur Visualisierung PDF und zur Publikation im Internet das auch hier später verwendete Open-Source-Format SVG (Scalable Vector Graphics) bzw. noch Flash von Macromedia.

SVG kann in Zukunft AI oder FH ersetzen. Es bietet weitaus mehr Möglichkeiten, beispielsweise der Einbau von Interaktion oder Animation. SVG ist vom World Wide Web Consortium (W3C) als Standard definiert worden und die Viewer sind für jeden Gratis erhältlich. Somit ist man nicht mehr von einer Firma und deren Geschäftspolitik abhängig.

## 4 Übersichtskarten

Dass die Gebietsauswahl und die damit verbundene Suche grafisch über Karten möglich ist, ist für das Internet neu. In Hinblick auf die Entwicklung eines modernen Tools wurde von den herkömmlichen pixelbasierten Karten Abstand genommen und auf qualitativ bessere vektorbasierte Karten zurückgegriffen. Dafür stehen momentan zwei Datenformate zur Verfügung. Das schon sehr weit von Macromedia entwickelte Flash (aktuelle Version MX) oder aber das, was die PlugIns angeht noch etwas in den Kinderschuhen steckende Scalable Vector Graphics (SVG).

SVG basiert auf XML und ist somit Webstandard. Es hat den großen Vorteil, dass der Quellcode wie bei HTML angezeigt und in jedem Text-Editor auch bearbeitet werden kann (OpenSource). Bewegung und Interaktion sind einfach beispielsweise mit JavaScript zu realisieren. Nachteil ist das noch fehlende gute Interface, um die Bearbeitung grafisch zu vereinfachen. Außerdem ist das benötigte PlugIn in den Browsern noch nicht enthalten und muss extra runtergeladen werden.

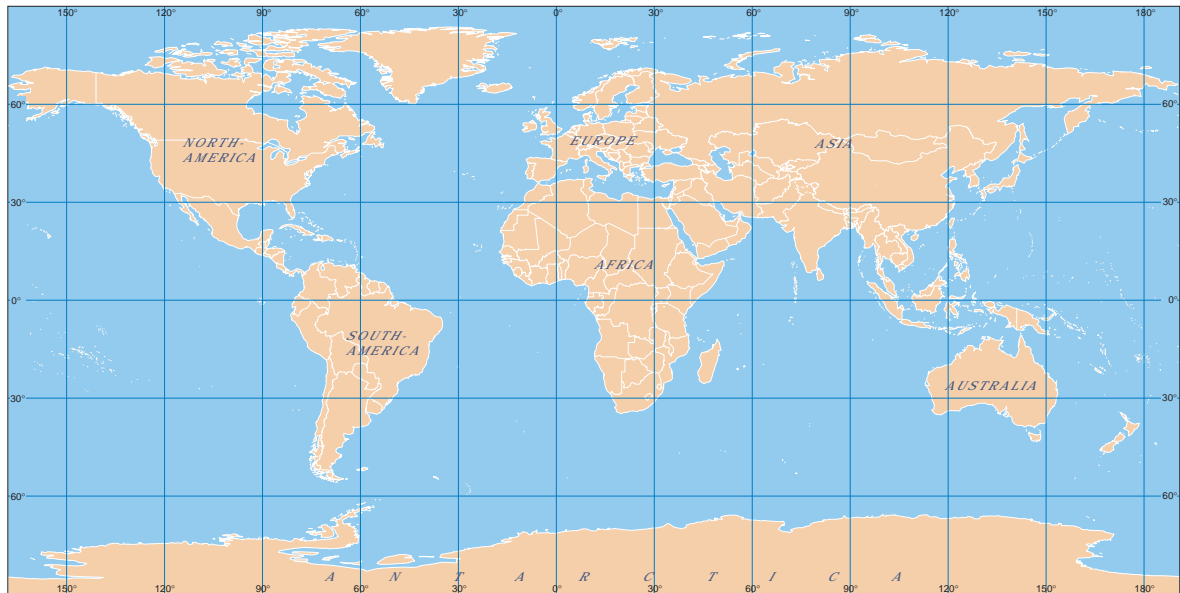
Das Flash-PlugIn ist schon seit geraumer Zeit in jedem neuen Browser integriert. Die damit erstellten Seiten stehen im World Wide Web somit einer viel größeren Anzahl von Benutzern zur Verfügung. Nachteil ist, dass die Erstellung und Bearbeitung nur mit der hauseigenen Software möglich ist. Die Animation und Interaktion läuft über das implementierte ActionScript. Das ist eine eigene Scriptsprache, die in der Handhabung aber dem JavaScript sehr ähnlich ist.

### 4.1 Kartenauswahl und Format

Wie der Name schon sagt, ist SVG skalierbar, jedoch auch nicht unendlich. Daher sollte bei der Wahl der Übersichtskarten Vorüberlegungen in punkto Detailierungsgrad oder Inhalte stattfinden. Um später den Anforderungen auch gerecht zu werden empfehlen sich diesbezüglich Tests, um die Datenmenge der Karte nicht mit zu vielen Inhalten unnötig groß werden zu lassen.

#### *4.1.1 Wahl der geeigneten Kartenprojektion*

Da das mit der Maus auf einer solchen Übersichtskarte aufgespannte Gebiet dem Suchgebiet in der Datenbank entsprechen soll, bieten sich Karten mit einem gleichmäßigen rechtwinkligen Gitternetz an. Die sogenannten quadratischen Plattkarten (quadratisches Gitternetz, siehe Abb. 4.1.1) erscheinen dem Betrachter möglicherweise etwas ungewöhnlich, da sich die Fläche zu den Polen hin stark vergrößert. Sie sind aber die einzigen, bei denen ohne aufwendige Matrix die geographische Position bestimmt werden kann. Bei einer einzigen Übersichtskarte hätte man den schöneren aber aufwendigeren Weg über solche Matrizen einschlagen können. Nun sollte es aber bei dieser einen Karte nicht bleiben. Sind alle Karten in einem quadratischen Netzgitter projiziert, so gilt auch für alle die selbe Abfragemethode. Siehe auch 4.3.



[ABBILDUNG 4.1.1: Quadratische Plattkarte]

#### 4.1.2 Festlegung der verschiedenen Karten

Anfangen mit einer Weltkarte ist es im SVG nur bedingt möglich, am Ende daraus eine sehr detaillierte, großmaßstäbige Karte zu erhalten. Über das PlugIn ist nur eine vordefinierte maximale Vergrößerung möglich, jedoch unbegrenzt verkleinerbar. Eine einzige Übersichtskarte wäre aber deshalb schon nicht ratsam, da eine solche Karte selbst im Vektorformat einen sehr großen Datenumfang bekommen würde (>10 MB). Solche Datenmengen würden die Vorteile von Vektorkarten gegenüber den datenintensiven Rasterkarten wieder zu Nichte machen.

Daher ist der Gedanke naheliegend, "Unterkarten" in das System mit einzubinden. Mittels PopUp-Menü kann die Weltkarte durch verschiedene Kontinentalkarten, Erdteil- und/oder Länderkarten ausgetauscht werden. Der Detaillierungsgrad wurde hier an der Häufigkeit der Karten in der Testdatenbank festgelegt. So wurden Länder wie die Schweiz, Deutschland und Frankreich, von denen z.T. komplette Kartenwerke vorhanden sind, als eigene Länderkarte vorgesehen, um eine genaue und differenzierte Suche zu ermöglichen. Von Staaten wie Chile, von denen insgesamt nur 20 Karten in meist sehr kleinen Maßstäben in der Kartensammlung zu finden sind, ist eine eigene Karte hinfällig. Solche Länderkarten könnten aber bei Bedarf der Suchmaschine beigelegt werden.

Nach einer ausführlichen Studie des Datenbestandes wurde schlussendlich diese hierarchische Abfolge festgelegt (Siehe Abb. 4.1.2), beispielhaft wurde die Weltkarte und die der Bundesrepublik Deutschlands bearbeitet.

1. Ordnung	2. Ordnung	3. Ordnung
Weltkarte	Afrika	
	Antarktis	
	Asien	
	Australien	
	Europa	Balkan Benelux Deutschland Frankreich Britische Inseln Italien Österreich Polen/Tschechei Schweiz Skandinavien Spanien/Portugal
	Nordamerika	Kanada USA
	Südamerika	

[TABELLE 4.1.2: Erdteilkarten]

#### 4.1.3 Erstellung der Übersichtskarten mit vektorbasierter Graphiksoftware

Die bekanntesten vektorbasierten Zeichenprogramme sind Corel Draw, Macromedia Freehand und Adobe Illustrator. Alle drei differenzieren kaum in der Funktionalität, die Unterschiede liegen eher im Detail.

Karten sind komplexe und beschriftete Grafiken. Der Vorteil gegenüber Rasterdaten ist, dass die Schrift, Objekte oder Linien nicht aus einer Abfolge von nebeneinanderstehenden Pixel besteht, sondern mathematisch definiert sind. Solche Pfade bzw. Vektoren bleiben immer randscharf, egal in welchem Vergrößerungsfaktor.

Mit einem solchen Zeichenprogramm können die Karten digitalisiert und/oder bearbeitet werden. Die oben erwähnten Programme haben aber unterschiedliche Exportfähigkeiten: Adobe Illustrator und Corel Draw haben eine SVG-Exportfunktion. Die Funktion von Adobe ist im Moment die Mächtigste. Macromedia unterstützt SVG im Moment nicht, da sie verständlicherweise nicht ihr eigenes Produkt "Flash" schwächen möchten.

#### 4.1.4 Bearbeitung der Karten im SVG-Format

SVG ist vom Aufbau vergleichbar mit HTML. Es besteht auch aus sogenannten Tags, die durch "<" und ">" immer gekennzeichnet sind.

So ist dies der typische Anfang eines SVG-Dokuments: XML-Version und Zeichenkodierung (Zeile 1), Doctype (Zeile 2) und das SVG-Anfangstag mit den Aufrufen der Funktionen im JavaScript (Zeile 35-42).

```

0001  <?xml version="1.0" encoding="utf-8"?>
0002  <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"      "http://www.w3.org/TR/2001/REC-
      SVG-20010904/DTD/svg10.dtd" [
...
0035  <svg width="720" height="360" viewBox="0 0 720 360" xml:space="preserve"
0036  onload="initMap(evt);initCoords(evt)"
0037  onzoom="AdaptivZoom();resetCoords()"
...
Auszug aus "world.svg"

```

Aus den Grafikprogrammen können die Ebenen im SVG als Gruppe übernommen werden. Kennzeichnend hierfür ist das `<g>`, endend mit `</g>`, wobei "Land" im unten aufgeführten Beispiel der Name der Gruppe ist. Was im Quelltext zu unterst steht, wird zuletzt, also zu oberst gezeichnet.

Mit "path" ist angegeben, dass ein Pfad folgt. Mit "style" wird angegeben, dass seine Elemente im Format 13 (`style="&st13;"`) definiert sind. Dieser Stil kann im Kopf des Dokuments festgelegt werden. Eine andere wäre eine extra Stil-Datei (Cascading Stylesheets, CSS). Der Stil beinhaltet beispielsweise die Farbe, Linienstärke oder bei einem Text die Angaben zu Schriftart und Größe.

```
0047 <g id="Land">
0048 <path style="&st13;" d="M245.9,161-0.5-0.21-1.5-0.21-1.5-0.21-1.7-
0.510.4,012.8,0L245.9,16z"/>
0049 <path style="&st13;" d="M232.8,16.21-2.7-0.31-0.3,010.4-
0.410.9,0.110.2,0.112.2,0.3L232.8,16.2z"/>
...
4360 <g id="Gridnumbers" style="&st22; &st8;" pointer-events="none">
4361 <text x="709.5" y="62.1"><tspan>60°</tspan></text>
...
```

Auszug aus "world.svg"

## 4.2 Adaptives Zoomen der Karten

Adaptiv bedeutet "angepasst", Zoomen das "dem Betrachter stufenlose Heranbringen oder Wegrücken eines Objektes" [DUDEN, 1997]. In der Kartographie bedeutet dies, dass sich die Menge der Informationen in der Karte automatisch der Zoomstufe anpassen. Der Karteninhalt ist sozusagen "Anpassungsfähig".

Ein elementarer Vorteil von SVG ist genau diese Skalierbarkeit. So wird die Karte in dieser Suchmaske anfangs einmal geladen und kann dann nahezu in Echtzeit stufenlos verkleinert oder vergrößert werden. Eine weitere Funktion des adaptiven Zoomens ist, dass Ebenen sichtbar oder unsichtbar gemacht werden können. Je nach Zoomlevel können nun Elemente auftauchen oder wieder verschwinden.

### 4.2.1 Prioritäts-Zuweisung der Ebenen in SVG

Ebenen können ineinander verschachtelt bzw. in übergeordneten Gruppen zusammengefasst werden. Im Endeffekt bedeutete dies, dass das Dokument in vier Teile aufgeteilt wurde. Am Beispiel der Weltkarte sind in der ersten Gruppe die Kartenelemente, die unabhängig von der Vergrößerung immer dargestellt sind. Dazu gehören die Wasser- und Landflächen und die Grenzen. Da aber der Vergrößerungsfaktor keinen Einfluss auf diese Ebenen hat, konnten sie auch ohne weitere Untergruppierung bestehen bleiben. Die zweite Gruppe, die bei der Originalgröße sichtbar ist, ist der Zoomlevel 0, hier die Kontinentbeschriftung. Wird nun vergrößert, verschwindet diese Schrift. Dafür wird die Länderbeschriftung und die Flüsse (Zoomlevel 1) sichtbar. Bei weiterem Einzoomen könnte eine weitere Ebene sichtbar gemacht werden. Sie wurde jedoch bei dieser Karte nicht verwendet, muss aber aus technischen Gründen im Dokument vorhanden sein, auch wenn sie hier funktionslos bleibt.

Die Angabe zur "visibility", also Sichtbarkeit, einer Ebene, kann entweder sichtbar ("visible") oder verborgen ("hidden") sein. Über dieses Attribut kann die Ansicht verändert werden. Vergleichbar mit einem Kippschalter, bei dem das Licht an- und ausgeschaltet wird.

```
4049 <g id="ZoomLevel1" visibility="hidden">
...
4341 <g id="ZoomLevel0" visibility="visible">
Auszüge aus "world.svg"
```

Zoomlevel	Ebenen	Zoom <3	Zoom > 3	Zoom > 5
ohne	Land	visible	visible	visible
	Lakes	visible	visible	visible
	Boundaries	visible	visible	visible
	Grids	visible	visible	visible
	FlickerGridsNumber	visible	visible	visible
	Antarctica	visible	visible	visible
	Gridnumbers	visible	visible	visible
0	ContinentNames	visible	hidden	hidden
1	OtherNames	hidden	visible	visible
	CountryNames	hidden	visible	visible
	Arrows	hidden	visible	visible
	TerritNames	hidden	visible	visible
2	-	hidden	hidden	visible

[TABELLE 4.2.1: Zoomleveleinteilung]

#### 4.2.2 Darstellungswechsel der Generalisierungsstufen mit JavaScript

Damit nun die in 4.2.1 festgelegten Generalisierungsstufen (Zoomlevel) auch automatisch sichtbar und unsichtbar werden, muss eine Funktion namens "AdaptivZoom()" im immer mitlaufenden JavaScript ("init.js") dies übernehmen. "currentScale" wird von SVG unterstützt und ist ein numerischer Wert, der in Zeile 702 eingelesen wird. In den if-Schleifen (Zeile 706 - 720) wird dieser Wert dann verglichen und je nach Größe werden die Attribute der Gruppen neu gesetzt. Im Browser ist dies dann auch beim Bildaufbau sofort bemerkbar.



[ABBILDUNG 4.2.2-1: CurrentScale <3]



[ABBILDUNG 4.2.2-2: CurrentScale >3]

```

699 // ----- ADAPTIVES ZOOMEN -----
700
701 function AdaptivZoom() {
702     currentScale=evt.target.ownerDocument.documentElement.currentScale
703     MyNodeLevel2=svgdoc.getElementById("ZoomLevel2");
704     MyNodeLevel1=svgdoc.getElementById("ZoomLevel1");
705     MyNodeLevel0=svgdoc.getElementById("ZoomLevel0");
706     if (currentScale > 3) {
707         MyNodeLevel1.setAttribute("display","inline");
708         MyNodeLevel0.setAttribute("display","none");
709     }
710     else {
711         MyNodeLevel2.setAttribute("display","none");
712         MyNodeLevel1.setAttribute("display","none");
713         MyNodeLevel0.setAttribute("display","inline");
714     }
715     if (currentScale > 5) {
716         MyNodeLevel2.setAttribute("display","inline");
717     }
718     else {
719         MyNodeLevel2.setAttribute("display","none");
720     }
721 }

```

Auszug aus "init.js"

## 4.3 Wertübergabe und Umrechnung der Cursorposition in Geographische Koordinaten

In 3.1.1 wurde erwähnt, dass die geographischen Koordinaten der Karten das Kernstück der Suche bedeuten. Daher musste eine Funktion eingebaut werden, die die Cursorposition in Bildschirmkoordinaten in Geographische Koordinaten umrechnet und in die Suche einfließen lässt. Eine Bedingung war, dass diese Funktion für alle Übersichtskarten gleich ist und auch bei jedem Zoomfaktor das korrekte Ergebnis bringt. In den folgenden Kapiteln wird gezeigt, wie diese beiden Variablen, also die unterschiedlichen Übersichtskarten und der Zoomfaktor, berücksichtigt werden. Außerdem wird erläutert, wie die Koordinaten zum Suchkriterium werden.

### 4.3.1 Voreinstellungen in JavaScript, Perl und SVG

Zu jeder Karte werden die im "init.js" getroffenen Voreinstellungen geladen. Später können diese Voreinstellungen aber auch individuell in der Suchmaske verändert werden. Im unten gezeigten Beispiel wird zuerst die zu ladende Weltkarte festgelegt (Zeile 74, init.js) und von Perl aus in den Browser geladen (Zeile 177 - 181, map\_search.pl). Dazu kommen noch die Koordinaten, die für die Suche ohne manuelle geographische Begrenzung übernommen werden (Zeile 75 - 79, init.js). Bei der Weltkarte sind dies übrigens 200° und nicht nur 180°, um sicher zu gehen, dass auch alle Karten mit inbegriffen sind. Für Deutschland sind die Koordinaten von 6° bis 15° Länge (West/Ost) und 55° bis 47.3° Breite (Nord/Süd) maßgeblich.



Die zweite Gruppe an Werten ist wichtig bei der manuellen Einstellung des Suchgebiets. Es sind die Koordinatenangaben der "unteren linken Ecke" (minimaler x- und y-Wert) und die gesamte Ausdehnung der Karte (Zeile 81 - 84, init.js). Wieder am deutschen Beispiel wären hier die Eckkoordinaten 6° (xGradAnfang) und 47.3° (yGradAnfang) und die Ausdehnung in x-Richtung von 9° und in y von 7.7°. Siehe auch Abbildung 4.3.1.

```

070 // - Austausch der Uebersichtskarten und angepassten Default-Werten -
071
072 function uebersicht(indexNr,listNr) {
073   if (indexNr == 0) { // -- WELT --
074     parent.document.Abfrageformular.svg_karte.value = '0';
075     parent.document.all.Abfrageformular.west.value = '-200';
076     parent.document.all.Abfrageformular.ost.value = '200';
077     parent.document.all.Abfrageformular.nord.value = '100';
078     parent.document.all.Abfrageformular.sued.value = '-100';
079     ...
081     parent.document.all.Abfrageformular.xGradAusdehnung.value = '360';
082     parent.document.all.Abfrageformular.xGradAnfang.value = '-170';
083     parent.document.all.Abfrageformular.yGradAusdehnung.value = '180';
084     parent.document.all.Abfrageformular.yGradAnfang.value = '-90';

```

Auszug aus "init.js"

```

175 #----- uebersichtskarten -----
176
177 if($svg_karte==0){
178   print<<EOF;
179   <embed src="http://www.karto.ethz.ch/~schlenker/svg/world.svg" width="800"
    height="500" type="image/svg+xml"></td>
180   EOF
181 }

```

Auszug aus "map\_search.pl"

Für die Korrespondenz zwischen dem SVG-Dokument und dem JavaScript sorgen die Aufrufe im Karten-Dokument (Zeile 36 - 41, world.svg). Hier werden die oben gezeigten Funktionen gestartet. Einerseits die ständige, bei einer Mausbewegung mitlaufende Funktion, die die Koordinaten umrechnet. Andererseits die Funktion, die bei einem Mausklick (runterdrücken und loslassen) die Koordinaten, bildlich gesehen, einfriert und die Variablen der Suche mit neuen Werten füllt (Siehe 4.3.2).

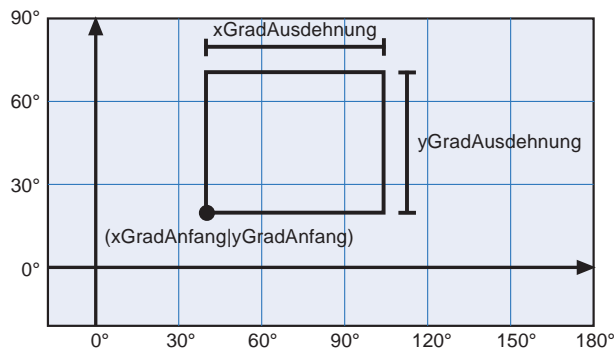
```

0035 <svg width="720" height="360" viewBox="0 0 720 360" xml:space="preserve"
0036   onload="initMap(evt);initCoords(evt)"
0037   onzoom="AdaptivZoom();resetCoords()"
0038   onscroll="resetCoords()"
0039   onmousemove="showCoords(evt)"
0040   onmousedown="anfangCoords()"
0041   onmouseup="endCoords()"
0042 >

```

Auszug aus "world.svg"





[ABBILDUNG 4.3.1: KARTENVOREINSTELLUNGEN]

### 4.3.2 Funktionen in JavaScript

Die ganze Funktion anhand des Quellcodes (von "init.js" - insg. 180 Zeilen) zu erklären, würde den Rahmen sprengen. Daher folgt nur eine textliche, in drei Schritte aufgeteilte Erläuterung.

#### 4.3.2.1 Initialisierung

Bei der Initialisierung werden zuerst die Beziehungen zwischen dem SVG-Dokument und dem JavaScript festgelegt. So können bei jeder Mausbewegung die vom SVG-Plugin gelieferten "Pixelkoordinaten" der Cursorposition (onmousemove) eingeholt werden. Ebenso wird die Gesamtgröße (width/height), die momentane Transformation (translate) und das sichtbare Gebiet (viewport) des SVG-Dokuments übergeben und in Variablen gespeichert. Für alle weiteren Berechnungen stehen sie somit zur Verfügung.

#### 4.3.2.2 Zurückstellen bei Vergrößerung / Verschiebung

Bei jeder Veränderung der Zooms oder beim Verschieben des Kartenausschnittes müssen die in 4.3.2.1 initialisierten Größen den neuen Umständen angepasst werden. Das System berechnet beispielsweise das Verhältnis von Zoomfaktor und Originalpixel, um die tatsächlich dargestellten Bildschirmpixel zu bekommen. Mit der sich veränderten Viewbox entsteht natürlich auch ein neuer sichtbarer "Nullpunkt", der zum eigentlichen voreingestellten Nullpunkt der Karte aufaddiert werden muss. Somit ist gewährleistet, dass der Cursor auch die richtigen Pixelkoordinaten übergibt.

#### 4.3.2.3 Umrechnung / Übergabe

Die eigentliche Umrechnung geschieht dann durch die eingelesenen und voreingestellten Variablen. Dieses System bietet die Möglichkeit, über diese Voreinstellungen im letzten Schritt den Übergang von den Bildschirmkoordinaten zu den Geographischen Koordinaten zu schaffen. Beim Drücken der linken Maustaste wird die Funktion gestartet, die den Anfangspunkt (nord/west) festhält, beim Loslassen dann den Endpunkt (sued/ost). Je nach Größe werden Süd und Nord bzw. West und Ost evtl. noch vertauscht, so dass es schlussendlich egal ist, wie der Benutzer seinen Suchrahmen aufzieht.

In einer Testphase wurden noch die geographischen Koordinaten ständig angezeigt, jedoch aus Übersichtlichkeitsgründen am Ende wieder weggelassen. Es ist aber denkbar, über ein Interface diese wieder zu visualisieren. Es stellte sich jedoch die Frage nach dem tatsächlichen Nutzen dieser Zusatzinformation.

## 4.4 Ausbaumöglichkeiten

Dieses System mit den verschiedenen Übersichtskarten hat ein riesiges Ausbau-Potential. Die Gliederung der Karten dieser Suchmaske wurde in 4.1.2 schon näher begründet. Sie ist aber eher das Resultat von Überlegungen, die minimale Ansprüche mit einbeziehen. Grenzen, was die Anzahl oder den Detaillierungsgrad der Karten angeht, sind nicht gesetzt. Sinnvolle Möglichkeiten, die Suche mittels zusätzlicher Karten oder Features zu verbessern, zeigen die nächsten drei Beispiele.

### 4.4.1 Indexblätter

Indexblätter sind sowohl von Atlanten als auch von Kartenwerken bekannt. Sie geben einen schnellen Überblick von den Ausmaßen des gesamten Werkes. Sie sind auch eine Art Übersichtskarten und lassen eine schnelle Suche über die Nummerierung zu.

Am Beispiel dieser Suchmaske wäre denkbar, dass mit Hilfe des adaptiven Vergrößerns die verschiedenen Indexblätter der Kartenwerke dargestellt werden. Angefangen beispielsweise mit den deutschen kleinmaßstäbigen ÜK200 (Übersichtskarte 1 : 200 000) bis hin zu den TK25 (Topographische Karte 1 : 25 000). So könnte der/die Suchende von Topographischen Karten schon über solche in Blätter gerasterte Übersichten einen schnellen Einblick bekommen.

### 4.4.2 Interaktives Anzeigen der Kartenausmaße

Schaut man sich aus der Trefferliste die Mehrinformationen zu einer Karte an, so könnte man diese zusätzlich noch als groben Rahmen in der Übersicht gleich zeichnen lassen. Da jede Karte über die Koordinaten gefunden wird, wäre der Rückwärtsschritt ebenfalls möglich, indem die Koordinaten in Cursor-Positionen zurückgerechnet würden. So würde man innerhalb des gewählten Suchgebiets eine weitere Differenzierung bekommen, was letztendlich einen noch genaueren Eindruck von der Karte hinterlassen würde.

### 4.4.3 Möglichkeit zum Druck

Da die Übersichtskarten im Vektorformat erstellt wurden, ist sogar ein qualitativ hochwertiger Druck auf Papier möglich. In der jetzigen Form können die Karten wegen der implementierten Interaktivität nicht ohne Fehlermeldungen einfach gedruckt oder exportiert werden. Mit einem zusätzlichen Button in der Suchmaske könnte ohne großen Aufwand die für den Druck optimierte Karte geladen oder auf dem Rechner des Benutzers gespeichert werden.

## 5 Suchmaske

### 5.1 Suchmaske im Allgemeinen

Wer heutzutage schon einmal im Internet etwas gesucht hat, konnte sie kaum umgehen. Die bekanntesten sind momentan bei Google, Amazon oder GelbeSeiten anzutreffen. Aus wissenschaftlicher Sicht stehen diesen kommerziellen Suchinstrumenten die Suchmasken der Bibliotheken oder Hochschulen in nichts mehr nach. Die Hochschulen stellen auf ihren Servern einer breiten Masse ihr Wissen zur Verfügung. Es gilt, diese Informationen auffindbar zu machen. Hinter jeder Suchmaske steckt natürlich auch eine Suchmaschine, die die Anforderungen der Benutzer umsetzt.

Eine Suchmaske ist die Schnittstelle zwischen den Daten und den Benutzern. Diese Suchmaske hier ist eine graphische Oberfläche, die die gezielte Suche mit einfachen Mausklicks ermöglicht. So sind die sonst recht komplizierten Datenbankabfragen für alle ganz einfach realisierbar.

Die große Kunst einer guten Suchmaske besteht darin, den Suchenden intuitiv zu seinem gewünschten Ziel zu führen. Selbst ein Erstbenutzer sollte die Online-Hilfe unberührt lassen. Der Entwickler muss darauf achten, dass alle Funktionen selbsterklärend sind.

Es ist das Ziel, dem Suchenden ein optimales Ergebnis zu liefern. Es bringt nichts, wenn die Anzahl der Treffer zu groß oder zu klein wird. Die Suche sollte also genau so differenziert sein, dass letztendlich nur eine kleine und treffende Auswahl übrig bleibt. Diesen Spagat zwischen einer effektiven Suche und dem Beibehalten der Übersichtlichkeit gilt es genau zu machen.

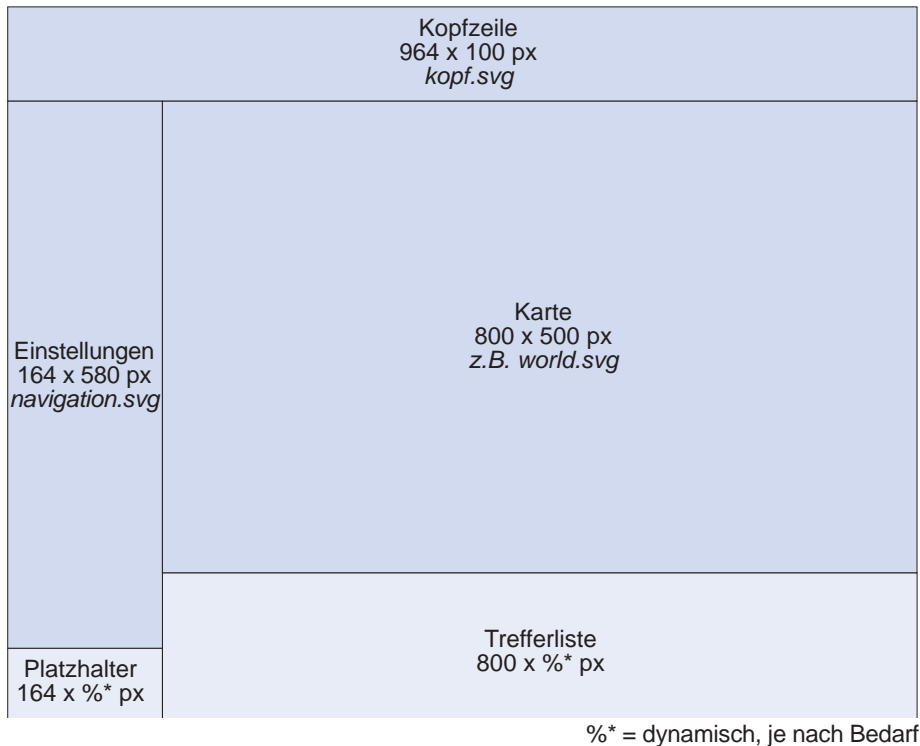
### 5.2 Sichtbare Oberfläche / Layout der Suchmaske

Um eine graphische Oberfläche möglichst einer breiten Öffentlichkeit zugänglich zu machen, bietet sich der Aufbau in einem HTML-Browser an. Damit eine übersichtliche und optisch ansprechende Seite entsteht, empfiehlt sich ein nach Funktionalität unterteiltes Layout. So wurde hier jedenfalls vom Aussehen her verschiedene Frames verwendet. Frames sind eine gängige Methode, um HTML-Seiten in mehrere Bereiche zu unterteilen, die dann auch einzeln mit Inhalt gefüllt werden können. Aus technischen und visuellen Gründen wurde hier aber mit einer Tabelle gearbeitet, womit augenscheinlich das gleiche Endergebnis erreicht wurde. Die Interaktivität zwischen den Frames funktionierte nicht zufriedenstellend. Vorteil von Frames ist, dass der Inhalt eines einzigen Teils ausgetauscht werden kann, ohne dass die anderen davon betroffen sind. Nachteil jedoch sind die unschönen Scrollbalken, wenn die darin platzierten Dokumente größer sind. Im Gegensatz dazu kann eine Tabelle immer nur als Ganzes geladen werden. Dies verlangsamt zwar den Aufbau der Seite, sieht aber in einem kleineren Browserfenster durch nur einen Scrollbalken einheitlicher aus.

Eine reine SVG-Seite klappte nicht, weil die Verlinkung auf eine SVG-Graphik nicht dynamisch unterstützt wurde.

In Punkto Aussehen und Benutzung ähneln sich die meisten Suchmasken. Es gibt jedoch gerade bei den Kartensammlungen größere Unterschiede. Das Hauptmerkmal des hier entwickelten Prototyps ist der Einbezug einer Karte. Die Karte ist zwar dominant, jedoch sind die Möglichkeiten zu einer Suche noch gut erkennbar.

Das Browserfenster wurde in fünf Teile gesplittet, oben ein Kopf mit der Überschrift und den Werkzeugen, links die Suchkriterien, rechts die Übersichtskarten und unten ein Platzhalter ohne Funktion und daneben die Trefferliste.



[ABBILDUNG 5.2: Seiten-Layout]

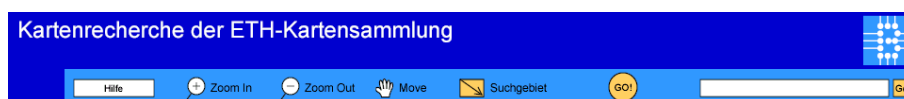
Um die Zugehörigkeit der Bibliothek der ETH Zürich wiederzugeben, wurden Stilelemente und Farbe dem Design der Webseiten der ETH-Bibliothek angeglichen. Die Technik ist jedoch unabhängig. Bei uns handelt es sich um eine HTML-Seite, in der drei SVG-Dokumente platziert sind. Deren Funktion wird in den Folgenden Zeilen noch näher erklärt.

### 5.2.1 Seitenkopf

Wie der Name schon sagt, bildet der Seitenkopf den oberen Anfang der Seite. Hier sind neben der Überschrift und dem ETH-Bibliotheks-Logo hauptsächlich die Werkzeuge zu sehen. Diese wurden zwar nicht direkt implementiert, deren Funktion ist aber durch Knöpfe (Buttons) angedeutet. Ebenfalls nur angedeutet ist die Textsuche, die in 5.5.1 noch näher erläutert wird.

Diese SVG-Datei ist im HTML platziert und statisch. Das heißt, sie kann nicht vergrößert oder verkleinert werden. Solche Einstellungen können in SVG mit "zoomAndPan="disable"" getroffen werden. Hier würden diese Einstellungen sowieso keinen Sinn machen, sie würden lediglich das Gesamtbild zerreißen.

In der Mitte befindet sich der etwas größere "GO"-Button, der die Suche startet.



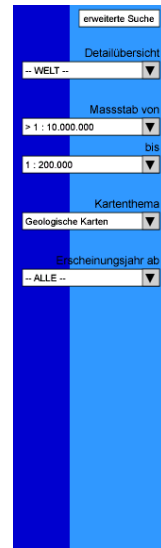
[ABBILDUNG 5.2.1: Kopf]

### 5.2.2 PopUp-Menüs

PopUp-Menüs sind eine gängige und auch graphisch gute Lösung, um mit der Maus Einstellungen zu treffen. Beim Mausklick auf eine Schaltfläche springt ein Menü auf, aus dem fest definierte Kriterien wählbar sind. Solche Menüs sind einerseits platzsparend auf einer Seite unterzubringen, andererseits auch selbsterklärend in der Bedienung.

Die für eine Suche relevanten Themen oder Kriterien können somit einfach und schnell getroffen werden.

Solche Menüs sind unter HTML als einfacher Befehl integrierbar. Im SVG-Format müssen sie als extra Funktion miteingebunden werden. Da die Suchmaske einheitlich gestaltet werden sollte, wurde der steinigere Weg mit SVG gewählt. Die Funktion für solche PopUp-Menüs konnte jedoch von Andreas Neumann (ETH-Zürich) übernommen und den gewünschten Anforderungen angepasst werden.



[ABBILDUNG 5.2.2: Suchkriterien]

### 5.2.3 Übersichtskarte

Die Übersichtskarten und die Ansprüche an diese wurden im Kapitel 4 schon näher erläutert. In der Suchmaske übernehmen sie eine tragende Rolle, da der Benutzer über sie das geographische Suchgebiet genau bestimmen kann. Diese Karten können über das PopUp-Menü namens "Detailübersicht" gewechselt werden. Beispielhaft wurde nur die der Welt und Deutschlands umgesetzt. Ansonsten erscheint ein Platzhalter.

Die maximalen Ausmaße der Karte wurden im HTML (bzw. Perlscript) festgelegt. Sie sind 800 x 500 Pixel. Wird nun die Karte eingezoomt, so vergrößert sie sich nur innerhalb dieses Rahmens. Darin verschieben kann man sie über die gedrückte "Alt"-Taste und der Maus.



[ABBILDUNG 5.2.3: Karte]

#### 5.2.4 Platzhalter

Der Platzhalter ist eine Zelle der Tabelle, die lediglich mit der passenden Hintergrundfarbe hinterlegt wurde. Sie hat keine eigentliche Funktion, sondern dient lediglich der Ästhetik.



[ABBILDUNG 5.2.4: Platzhalter]

#### 5.2.5 Trefferliste

Die Trefferliste ist die textliche Angabe der gefundenen Karten. Die Suchkriterien stimmen mit den Einträgen in der Datenbank überein. Diese Ausgabe erfolgt in einer einfachen Text-Tabelle, die je nach Anzahl der Einträge ihre Länge variabel verändert.

Das Layout ist bewusst einfach und schlicht gehalten, die gute Lesbarkeit und Übersichtlichkeit stehen hier im Vordergrund. Zum Inhalt und technischen Aufbau dieser Tabelle siehe auch Kapitel 5.4.

#	Kartentitel	Herausgeber	Jahr	Maßstab 1:	Thema
1	<a href="#">CARTA GEO-ESTRUCTURAL DEL URUGUAY 1:2 000 000</a>	Ministerio de industria y energía, Instituto geológico "Ing Eduardo Terra Arocena"	1979	2000000	GEOLOGISCHE KARTEN
2	<a href="#">MAPA GEOLOGICO DE BOLIVIA 1:1 000 000</a>	Yacimientos petrolíferos fiscales Bolivianos & Servicio geológico de Bolivia	1978	1000000	GEOLOGISCHE KARTEN
3	<a href="#">Mapa geológico do estado da Bahia 1:1 000 000</a>	Estado do Bahia, Secretaria das minas e energia, coordenação da produção mineral	1978	1000000	GEOLOGISCHE KARTEN
4	<a href="#">CARTE STRUCTURALE DES ANDES SEPTENTRIONALES DE BOLIVIE 1:1 000 000</a>	O.R.S.T.O.M.	1978	1000000	GEOLOGISCHE KARTEN
5	<a href="#">MAPA GEOLOGICO DO ESTADO DO RIO DE JANEIRO 1:400 000</a>	Instituto de pesquisas espaciais	1977	400000	GEOLOGISCHE KARTEN

[ABBILDUNG 5.2.5: Trefferliste]

## 5.3 Suche / Interaktion mit der DB

Wenn man von Interaktion spricht, ist das hier ein Bindeglied, das zwei unterschiedliche Systeme oder Komponenten miteinander verbindet. Die gewählte Verbindung der graphischen Oberfläche mit der Datenbank ist nicht unbedingt die einzig mögliche, jedoch eine gut funktionierende.

### 5.3.1 Aufbau der Seite mittels Perl

Um unsere Schnittstelle zu entwickeln reichen die Möglichkeiten einer normalen, reinen HTML-Seite nicht aus. Eine Abfrage an die Datenbank kann beispielsweise nicht im HTML formuliert werden. Deshalb wurde hier das Perlsript "map\_search.pl" entwickelt und verwendet.

Das Perlscript liegt auf dem Server. Der Client (Browser) stellt eine Anfrage an der Server, nämlich den Aufruf des map\_search.pl via http. Findet der Server das Dokument, so nimmt er es und schaut in seiner Konfigurationsdatei, ob er etwas damit machen muss. Im Fall von .html sieht der Server, dass er das File lediglich als Textdatei zurückschicken muss. Im Fall von .pl hat der Server die Anweisung, zuerst die Datei von einem Programm namens "Perl" lesen zu lassen und danach das Resultat zurückzuschicken. Perl liest das .pl-File und macht eine HTML-Datei daraus und schickt diese zu dem Client.

Unten zeigt das Beispiel einerseits den Header dieser Seite im Perl (ab Zeile 20) und andererseits noch wie die Tabelle erstellt wird (ab Zeile 149). In dieser Tabelle werden dann die SVG-Teile integriert (Zeile 153).

Damit die Übersichtskarten auch austauschbar werden, musste hier mit Bedingungen gearbeitet werden. Anfangs soll immer die Weltkarte erscheinen, später aber dann auf Befehl eine Andere (siehe auch 4.3.1).

```

020     print $q->header;
021     print $q->start_html(-title=>'Suchmaske der ETH-Kartensammlung',
022         -author=>'Schlenker / IKA-ETHZ',
023         -BGCOLOR=>'white',
...
149     <table border="0" cellpadding="0" cellspacing="0" width="964" height="100%">
150
151     <!-- Kopfzeile -->
152     <tr height="100">
153         <td width="964" height="100"><embed
src="http://www.karto.ethz.ch/~schlenker/svg/kopf.svg" width="964" height="100"
type="image/svg+xml" align="left"></td>
154     </tr>
155
156     <tr valign="top" align="left" height="700">
157         <td width="964" height="100%" valign="top" align="left">
158         <table border="0" cellpadding="0" cellspacing="0" width="964" height="100%">
...

```

Auszug aus "map\_search.pl"



### 5.3.2 Deklaration und Übergabe der relevanten Variablen

Die Werte können mit Hilfe von Variablen von einem Programmiersystem zum Anderen übergeben werden. In unserem Fall wurden insgesamt vier Sprachen verwendet, die miteinander kommunizieren müssen.

#### 5.3.2.1 Deklaration und Initialisierung der Variablen

In Perl und in JavaScript müssen die Variablen vorher deklariert werden, sozusagen "namentlich erwähnt". Ein Vorteil beider Sprachen ist, dass den Variablen im Vorfeld keine Datentypen zugewiesen werden müssen, sie können also sowohl Strings als auch Zahlen beinhalten. Im Perl werden sogenannte "Hidden-Fields" erstellt. Diese Felder sind aber HTML-Objekte, deren Werte letztendlich auch von JavaScript manipuliert werden können. Das folgende Beispiel zeigt, wie ein solches "verstecktes Feld" mit dem Namen 'massstab1' und dem Anfangswert von '99999999' erstellt wird.

```
049    print $q->hidden(-name=>'massstab1',-value=>[ '99999999' ] );
Auszug aus "map_search.pl"
```

Angezeigt wird jedoch in der fertigen Suchmaske bei "von Maßstab", was dem 'massstab1' entspricht, der Wert '> 1 : 10.000.000'. Diese Voreinstellungen stammen aus dem "init.js" und sind mit der Funktion verbunden, die die PopUp-Menüs erstellt.

Nach diesem Prinzip konnten sämtliche Suchkriterien deklariert, initialisiert und entsprechend auch visualisiert werden.

#### 5.3.2.2 Verbindung von JavaScript- und Perl-Dokument

JavaScript ist maßgeblich für die Interaktivität der Seite verantwortlich. Alle Funktionen, angefangen mit der Möglichkeit des adaptiven Zoomens bis hin zu den PopUp-Menüs, sind im "init.js" eingebunden. Das elementare daran ist jedoch, dass es sich mit dem Perl-Dokument die Variablen teilen kann bzw. an dieses die neuen Werte übergibt. In dem oben gezeigten Beispiel wurde zuerst dem 'massstab1' der Wert '99999999' zugewiesen. Stellt man nun über das Menü den nächst kleineren Maßstab ein, so ist die 'indexNr' gleich '1' und der Variablen 'massstab1' wird im Perl (siehe unten) der neue Wert '10000000' zugewiesen.

```
305    function massstab1(indexNr,listNr) {
306        if (indexNr == 0) { parent.document.all.Abfrageformular.massstab1.value =
        '99999999'; }
307        if (indexNr == 1) { parent.document.all.Abfrageformular.massstab1.value =
        '10000000'; }
...
Auszug aus "init.js"
```

```
096    my $massstab1 = $q->param('massstab1');
Auszug aus "map_search.pl"
```

Damit die schon gemachten Einstellungen auch für weitere Suchen beibehalten bleiben, wurden sogenannte "Merker" eingesetzt. Diese speichern den momentanen Zustand jedes Kriteriums und geben ihn bei einem neuen Seitenaufbau wieder ab. Erst nach dem Verlassen der Seite wird der Originalzustand wieder aufgebaut. Es wäre denkbar einen Reset-Knopf einzubauen, um den Neuaufbau mit den Voreinstellungen manuell tätigen zu können.



### 5.3.2.3 Verbindung von Perl mit der MySQL-Datenbank

In Perl kann ein SQL-Statement formuliert werden. Mit Hilfe des Perl-Dokuments kann also die Verbindung zu der MySQL-Datenbank hergestellt werden. Dies ist auch eine Begründung für Wahl dieser beiden Technologien. Dieses Verbinden erfolgt mit folgendem Befehl, die Variablen wurden schon im Vorfeld initialisiert.

```
016 my $dbh = DBI->connect( $data_source, $username, $password);
    Auszug aus "map_search.pl"
```

### 5.3.2.4 Datenbankabfrage

Das SQL-Statement beinhaltet alle Suchkriterien, bei denen auch Einstellungen getroffen werden können. Es wurde darauf geachtet, dass die Namensgebung der Variablen und der Spalten der Datenbank identisch bzw. ähnlich sind. Damit lassen sie sich immer eindeutig zuordnen und wiedererkennen.

Die Abfragen können nach den Regeln der Boole'schen Algebra verknüpft oder in sich verschachtelt werden. Sämtliche numerische Werte werden direkt mit den "Gesetzen der Anordnung" verglichen, also beispielsweise "ist größer als" [SIEBER 1991]. Wobei noch anzumerken ist, dass die hier verwendete Datenbankabfrage noch recht einfach und übersichtlich gehalten wurde.

```
116 $sth = $dbh->prepare( "SELECT karten_id,karten_titel,herausgeber,jahr,massstab,thema
    FROM karten2
117     WHERE west >= '$west'
118     AND ost <= '$ost'
119     AND nord <= '$nord'
120     AND sued >= '$sued'
121     AND massstab <= '$massstab1'
122     AND massstab >= '$massstab2'
123     AND thema LIKE '$thema'
124     AND jahr >= '$jahr'
125     AND jahr <= '$jahrMax'
126     AND herausgeber LIKE '%$herausgeber%'
127     AND sprache LIKE '$sprache'
128     AND bibliothek LIKE '%$bibliothek'
129     ORDER BY jahr DESC
130 ");
    Auszug aus "map_search.pl"
```

## 5.4 Ausgabe der Kartendaten

Die Ausgabe der der Abfrage entsprechenden Karteneinträge ist die Trefferliste. Dieser Name ist von "zutreffend" ableitbar und ist die eigentliche Information, die der Benutzer haben möchte.

### 5.4.1 Anzeigen der Trefferliste

Die normale Ausgabe erfolgt in Tabellenform und wird je nach Anzahl ganz individuell in der Länge variieren. Die Tabelle wird auch von Perl generiert und zeigt die durch das SQL-Statement selektierten Einträge (ID, Titel, Herausgeber, Jahr, Maßstab und Thema). Vergleiche auch Abbildung 5.2.5.

In Zeile 198 bis 213 vom Perlscript wird die Tabelle mit ihren Spalten erstellt und beschriftet. Die Höhe von 250 Pixel im Browserfenster ist eher als Mindestgröße anzusehen, die die Tabelle später einnehmen wird.

```

198     print<<EOF;
199         <tr height="120">
200
201             <!--Trefferliste -->
202             <td width="800" height="250" valign="top" >
203                 <table border="1" cellpadding="2" cellspacing="0" width="800">
204                     <thead <tr> <td><b>\#</b></td>
205                         <td><b>Kartentitel</b></td>
206                         <td><b>Herausgeber</b></td>
207                         <td><b>Jahr</b></td>
208                         <td><b>Massstab 1 :</b></td>
209                         <td><b>Thema</b></td>
210                     </tr>
211                 </thead>
212
213     EOF
214
```

Auszug aus "map\_search.pl"

Zeile 215 bis 235 vom Perlsript überführen dann die Daten in die Tabelle. Dies geschieht Eintrag für Eintrag und Zeile für Zeile, einfach der Reihenfolge nach. Durch das Datenbanksystem werden die Einträge schon nach dem Erscheinungsjahr der Karte absteigend sortiert übergeben.

```

215     $index=0;
216     #----- Herausschreiben der Daten -----
217     for $i ( 0 .. ${table} ) {
218         $index++;
219         print "<td>$index</td>";
220         for $j ( 0 .. ${table->[$i]} ) {
221             if($j!=1){
222                 if($j!=0){
223                     if(($table->[$i][$j]) eq ""){print "<td> - </td>";}
224                     else {print "<td> $table->[$i][$j] </td>";}
225                 }
226             }
227             elsif($j==1){
228                 if(($table->[$i][$j]) eq ""){print "<td> - </td>";}
229                 else {
230                     $id=$table->[$i][$0];
231                     $url="/schlenk-cgi/detail.pl/?$id";
232                     print "<td> <a href=$url target=\"_blank\">$table->[$i][1]</a> </td>";}
233             }
234         }
235     }

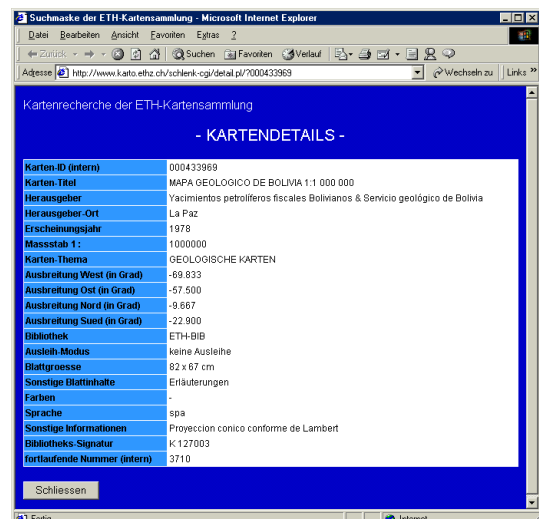
```

Auszug aus "map\_search.pl"

### 5.4.2 Detailausgabe

Bei Interesse kann der Benutzer sich noch den gesamten Datenbankeintrag einer Karte ansehen. Wählt man per Mausklick den Kartentitel in der Trefferliste aus, so springt ein neues Fenster mit einer umfassenden Tabelle auf.

Dies funktioniert ähnlich wie das Erstellen der Trefferliste. Im Quelltext aus 5.4.1 kommt in der Zeile 231 der Aufruf für ein neues Perldokument ("detail.pl"). Zusammen mit der ID der angewählten Karte wird eine Anfrage an den Server geschickt. Der Server sendet ein HTML-File an den Client zurück. In einem neuen Browserfenster wird eine neue Tabelle erstellt und mit den abgefragten Informationen gefüllt. Auf Knopfdruck kann dann dieses Fenster wieder geschlossen werden.



[ABBILDUNG 5.4.2: Detailausgabe]

## 5.5 Ausbaumöglichkeiten

Auch wenn Suchmasken im Allgemeinen schon einen hohen Standart erreicht haben, so lassen sich die Programmierer immer neue Möglichkeiten einfallen, wie man es dem Suchenden noch einfacher machen kann und dabei immer genauere Ergebnisse zu bekommen.

### *5.5.1 Textsuche*

Die Textsuche ist bei den meisten Suchmasken der zentrale Punkt. Sie wurde aber bei diesem Prototypen nicht implementiert. Es wurde in erster Linie darauf geachtet, die graphische Suche mit Maus und Übersichtskarte zu demonstrieren. Das Einbinden eines Namensregister würde aber die Suche komplettieren. Dabei könnte man auf das verwendete Prinzip über die Koordinaten-Suche aufbauen. Eine weitere Datenbank, bei der Städte, Gewässer, Verkehrsverbindungen oder gar Regionen mit geographischen Bezugskordinaten gespeichert sind, könnte die Basis für eine Textsuche bedeuten. Vergleichbar mit heutigen Instrumenten der Verbindungssuche der Bahngesellschaften würde ein Ort ausgewählt. Bei mehreren gleichen Namen müsste noch genauer differenziert werden. Dieser Punkt oder bei Flüssen auch eine Linie würde die Koordinaten an die herkömmliche Suche weitergeben, alle anderen Einstellungen zum Thema oder Maßstab der Karte sind natürlich immer noch relevant.

Der große Vorteil eines Städtereisters mit Bezugskordinaten liegt darin, dass nicht nur über den Titel der Karte gesucht werden kann, sondern die Geographische Lage des Objekts. In solch einem Register selbst Vororte verzeichnet, die sonst in keinem Kartentitel auftauchen.

### *5.5.2 Mehrere Sprachen wählbar*

In der Schweiz werden drei Amtssprachen (deutsch, französisch, italienisch) gesprochen und dem Englisch als internationale Sprache einen großen Stellenwert zugeschrieben. Es ist gut vorstellbar, dass diese Suchmaske für die ETH-Kartensammlung auch Multilingual angeboten werden soll. Die Umstellung der Suchmaske würde nur einen geringen Aufwand bedeuten, da sowieso nicht sehr viel Text vorhanden ist. So könnte der Benutzer theoretisch vor dem Aufstarten des Systems die Wahl seiner bevorzugten Sprache treffen oder auch nach dem automatisch erfolgten "deutschen" Start die Sprache ändern.

Dies hat aber wohlgermerkt nichts mit dem Kriterium der Karten-Sprache im PopUp-Menü "Sprachen" zu tun, es geht hier lediglich um die Oberfläche.

### *5.5.3 Reservation der Karten*

Bei den meisten Suchmaschinen der Bibliotheken ist man es gewohnt, bei Interesse sich per Mausklick ein Buch vormerken zu lassen. Theoretisch wäre dies natürlich auch bei der Kartensammlung möglich. Es erfordert aber die Anbindung an das übergeordnete Bibliothekssystem, bei dem alle Ausleihen vermerkt werden. Dies würde eine weitere Abfrage an den Bibliothek-Server bedeuten.

## 6 Schlusswort

### 6.1 Erfolg / Misserfolg

Der hier entwickelte Prototyp einer graphischen Suchmaske der Kartensammlung der ETH-Zürich ist funktionsfähig im Internet zu besichtigen (URL im Anschluss). Die Datenbank ist ein Probeauszug von der Kartensammlung, also nicht der volle Umfang. Mit über 10 000 Einträgen ist dieser Datensatz aber ausreichend.

URL: <http://www.karto.ethz.ch/~schlenker>

#### *6.1.1 Einschränkungen*

- Die Suchmaske läuft nur fehlerfrei auf einem PC, mit dem Microsoft InternetExplorer 5.0 oder 5.5 und dem Adobe SVG-PlugIn 3.
- Außerdem wird der zum Abgrenzen des Suchgebiets aufgezoogene Rahmen virtuell zwar gemacht, nicht aber visuell mitgezeichnet.
- Da es sich um eine Datenbankabfrage handelt, ist auch ein Server notwendig, auf dem die Datenbank hinterlegt ist. Ebenfalls wird auf dem Server das Programm Perl benötigt, das die Internetseiten generiert. Aus diesen Gründen ist eine Stand-Alone-Version auf einer CD nicht realisierbar. Zu Demonstrationszwecken wurden jedoch einige Beispielseiten erstellt, die die Funktionalität verdeutlichen sollen.

#### *6.1.2 Fazit*

Dieser Prototyp ist noch weit ausbaufähig und bricht damit ein neues Zeitalter der Kartensuche im Internet an. So fern SVG in den neuen Browsern zum Standard wird, ist dies eine Möglichkeit, den Kartenbestand der ETH einer breiteren Öffentlichkeit näher zu bringen. Für wissenschaftliche Auswertungen kann mit diesem Werkzeug schon im Vorfeld nach möglichen Quellen gesucht werden. Die Erleichterung wird den Studenten und Institutsmitarbeitern zu gute kommen.

Bleibt noch zu hoffen, dass dieser Prototyp in naher Zukunft durch eine ausgebaute Komplettversion ersetzt wird.

### 6.2 Unterstützung

Unterstützt wurde diese Diplomarbeit von:

- Prof. Dr. Christian Herrmann, FH-Karlsruhe, Fachbereich Geoinformationswesen, Studiengang Kartographie und Geomatik.
- Tobias Dahinden, ETH-Zürich/Schweiz, Institut für Kartographie.
- Dr. Jürg Bühler, ETH-Zürich/Schweiz, Bibliothek, Kartensammlung.
- Andreas Neumann, ETH-Zürich/Schweiz, Institut für Kartographie.

## 6.3 Literaturverzeichnis

- ASCHE, Hartmut & Christian HERRMANN 2002: Thematic Cartography; aus: Basic Cartography, Vol. 2 - Chapter 4. Reed Educational and Professional Publishing Ltd, Oxford.
- BAUER, Bernhard 1960: Ullstein-Synonymen-Lexikon. Ullstein GmbH, Frankfurt/M, Berlin.
- BÜCHNER, Heino & Dennis TRAUB & Rik ZAHARADKA & Oliver ZSCHAU 2000: Web Content Management. Galileo Press GmbH, Bonn.
- BÜHLER, Jürg 2001: Von der Kartensammlung zum Zentrum für Rauminformation; aus: Kartensammlung und Kartendokumentation 13 - Die digitale Kartenbibliothek (Momentaufnahme).
- DREIER, Thomas 1997: Urheberrecht und digitale Werkverwertung. Friedrich-Ebert-Stiftung, satz+druck GmbH, Düsseldorf.
- DUDEN 1997: Das Fremdwörterbuch. Bibliographisches Institut & F.A. Brockhaus AG, Mannheim.
- FIBINGER, Iris 2002: SVG - Scalable Vector Graphics. Markt+Technik Verlag, München.
- GRELL, Jens-Peter 2001: Das Projekt "Elektronisches Kartenarchiv"; aus: Kartensammlung und Kartendokumentation 13 - Die digitale Kartenbibliothek (Momentaufnahme).
- HEGERING, Heinz-Gerd & Sebastian ABECK & Bernhard NEUMAIR 1999: Intigriertes Management vernetzter Systeme. dpunkt - Verlag für digitale Technologie, Heidelberg.
- LUDWIG, Björn 2001: Management komplexer Systeme. VDI-Hauptgruppe, Düsseldorf.
- MAYERS Lexikon, 1997. Bibliographisches Institut & F.A. Brockhaus AG, Mannheim.
- MOOS, Alfred & Gerhard DAUES 1991: SQL-Datenbanken - Der Weg vom Konzept zur Realisierung in dBASE. Friedrich Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig.
- ROTHFUSS, Gunther & Christian RIED 2001: Content Management mit XML. Springer-Verlag (Xpert.press), Berlin, Heidelberg, New York.
- SAUER, Herrmann 1991: Relationale Datenbanken - Theorie und Praxis. Addison-Wesley (Deutschland) GmbH, München, Kevelaer, Troisdorf.
- SIEBER 1991: Mathematische Formeln. Ernst Klett Verlage GmbH, Stuttgart.

## 7 Anhang

### 7.1 kartensammlung.pl

```
#!/usr/bin/perl

use CGI;
use DBI;
$q = new CGI;
my $script_name = $q->script_name;
my $path_info = $q->path_info;
my $server = "karpc13";
my $database = "kartensammlung";
my $data_source = "DBI:mysql:$database:$server";
my $username = "karten";
my $password = "sammlung";
my $reply = $ARGV[0];

my $datenfile = "karten_eth_b.sed";

#----- anmelden bei der datenbank -----

my $dbh = DBI->connect( $data_source, $username, $password) or die "Can't connect to $data_source: $dbh-
>errstr\n";

#----- variablen -----

my $empty = "";
my $karten_id = $empty;
my $signatur = $empty;
my $karten_titel = $empty;
my $herausgeber = $empty;
my $herausgeber_ort = $empty;
my $jahr = $empty;
my $massstab = $empty;
my $thema = $empty;
my $koordinaten = $empty;
my $west = $empty;
my $ost = $empty;
my $nord = $empty;
my $sued = $empty;
my $bibliothek = $empty;
my $ausleihe = $empty;
my $blattgroesse = $empty;
my $sonst_blattinhalte = $empty;
my $farben = $empty;
my $sprache = $empty;
my $sonst_info = $empty;

#----- lese aus ascii dokument -----

open(DAT, $datenfile) or die("Could not open file!");
@rohdaten=<DAT>;

foreach $linie (@rohdaten) {
    #----- aufteilen der linie
    chop($linie);
    @nachdollar = split(/\$\$/, $linie);
    @preamble = split(/ /, $nachdollar[0]);
    @nachbalken = split(/\|\|\|\|\|/, $linie);

    $karten_id = $preamble[0];

    #----- schauen, ob ein eintrag existiert
    $sth = $dbh->prepare("SELECT ('karten_id') FROM karten2 WHERE (karten_id = '$karten_id') ");
    $resp=$sth->execute;
    $table = $sth->fetchall_arrayref;

    $counter=0;
    for $i ( 0 .. ${$table} ) { $counter++;}
    if($counter==0){
        $rc = $dbh->do("INSERT INTO karten2 (karten_id) VALUES ('$karten_id') " ) or die "Problems with karten_id
        $karten_id entry...";
    }
}
```

```

#----- falls nein, eintrag erstellen

#----- untersuchen der leitzahlen

#----- signatur -----

if($preamble[1] eq 'Z30-1') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq '3') {
      $laenge = length($wort);
      $signatur = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET signatur = '$signatur' WHERE karten_id = '$karten_id' " ) or die
        "Problems with signatur $signatur entry";
    }
  }
}

#----- karten_titel -----

if($preamble[1] eq '245') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'a') {
      $laenge = length($wort);
      $karten_titel = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET karten_titel = '$karten_titel' WHERE karten_id = '$karten_id' " )
        or die "Problems with karten_titel $karten_titel entry";
    }
  }
}

#----- herausgeber -----

if($preamble[1] eq '260') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'b') {
      $laenge = length($wort);
      $herausgeber = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET herausgeber = '$herausgeber' WHERE karten_id = '$karten_id' " ) or
        die "Problems with herausgeber $herausgeber entry";
    }
  }
}

#----- herausgeber_ort -----

if($preamble[1] eq '260') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'a') {
      $laenge = length($wort);
      $herausgeber_ort = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET herausgeber_ort = '$herausgeber_ort' WHERE karten_id = '$karten_id'
        " ) or die "Problems with herausgeber_ort $herausgeber_ort entry";
    }
  }
}

#----- jahr -----

if($preamble[1] eq '260') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'c') {
      $laenge = length($wort);
      $jahr = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET jahr = '$jahr' WHERE karten_id = '$karten_id' " ) or die "Problems
        with jahr $jahr entry";
    }
  }
}

```



```

}

#----- massstab -----

if($preamble[1] eq '255') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'a') {
      $laenge = length($wort);
      $massstab = substr($wort, 3, $laenge-1);
      @mass = split(/ /,$massstab);
      $ma = $mass[0].$mass[1].$mass[2];
      $src=$dbh->do("UPDATE karten2 SET massstab = '$ma' WHERE karten_id = '$karten_id' " ) or die "Problems
        with dbh...";
    }
  }
}

#----- thema -----

#----- schauen, ob ein thema-eintrag existiert
$ssth = $dbh->prepare("SELECT karten_id, thema FROM karten2 WHERE (karten_id = '$karten_id') ");
$respp=$sth->execute;
$thema_table = $sth->fetchall_arrayref;

if($thema_table->[$n][1] eq ""){
  if($preamble[1] eq '691E1') {
    foreach $wort (@nachdollar) {
      $ifwert = substr($wort, 0, 1);
      if($ifwert eq 'a') {
        $laenge = length($wort);
        $thema = substr($wort, 1, $laenge);
        #----- einfüegen in datenbank -----
        $src=$dbh->do("UPDATE karten2 SET thema = '$thema' WHERE karten_id = '$karten_id' " ) or die "Problems
          with thema $thema";
      }
    }
  }
}

#----- koordinaten -----

if($preamble[1] eq '255') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'c') {
      $laenge = length($wort);
      $koordinaten = substr($wort, 1, $laenge);
      @richtung = split(/\/,$koordinaten);
      # trennt 'koordinaten' in länge und
      breite
      @laengengrade = split(/\/,$richtung[0]);
      # trennt 'länge' in von und
      bis
      @breitengrade = split(/\/,$richtung[1]);
      # trennt 'breite' in von und
      bis
    }
  }
}

#----- west -----
@teil_west = split(/ /,$laengengrade[0]);
# trennt 'von länge'
in vorzeichen und werte
@koord_west = split(/\/,$teil_west[1]);
# trennt
'von länge werte' in grad und minuten
$west = $koord_west[0] + ($koord_west[1]/60);
# verrechnet 'von länge grad' und 'von länge minuten'
if($teil_west[0] eq 'W') {
  # setzt vorzeichen
  $west = $west*(-1);
}
$src=$dbh->do("UPDATE karten2 SET west = '$west' WHERE karten_id = '$karten_id' " ) or die "Problems
with west $west entry";

#----- ost -----
@teil_ost = split(/ /,$laengengrade[1]);
# trennt 'bis länge'
in vorzeichen und werte
@koord_ost = split(/\/,$teil_ost[1]);
#
trennt 'bis länge werte' in grad und minuten
$ost = $koord_ost[0] + ($koord_ost[1]/60);
#
verrechnet 'bis länge grad' und 'bis länge minuten'
if ($teil_ost[0] eq 'W') {
  # setzt vorzeichen
  $ost = $ost*(-1);
}

```

```

    }
    $src=$dbh->do("UPDATE karten2 SET ost = '$ost' WHERE karten_id = '$karten_id' " ) or die "Problems
with ost $ost entry";

#----- nord -----
@teil_nord = split(/ /,$breitengrade[0]); # trennt 'von breite'
in vorzeichen und werte
@koord_nord = split(/\^/, $teil_nord[1]); # trennt
'von breite werte' in grad und minuten
$nord = $koord_nord[0] + ($koord_nord[1]/60);
# verrechnet 'von breite grad' und 'von breite minuten'
if ($teil_nord[0] eq 'S') {
  # setzt vorzeichen
  $nord = $nord*(-1);
}
$src=$dbh->do("UPDATE karten2 SET nord = '$nord' WHERE karten_id = '$karten_id' " ) or die "Problems
with nord $nord entry";

#----- sued -----
@teil_sued = split(/ /,$breitengrade[1]); # trennt 'bis breite'
in vorzeichen und werte
@koord_sued = split(/\^/, $teil_sued[1]); # trennt
'bis breite werte' in grad und minuten
$sued = $koord_sued[0] + ($koord_sued[1]/60);
# verrechnet 'bis breite grad' und 'bis breite minuten'
if ($teil_sued[0] eq 'S') {
  # setzt vorzeichen
  $sued = $sued*(-1);
}
$src=$dbh->do("UPDATE karten2 SET sued = '$sued' WHERE karten_id = '$karten_id' " ) or die "Problems
with sued $sued entry";
}
}
}

#----- bibliothek -----

if($preamble[1] eq '040') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'a') {
      $laenge = length($wort);
      $bibliothek = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET bibliothek = '$bibliothek' WHERE karten_id = '$karten_id' " ) or
      die "Problems with bibliothek $bibliothek entry";
    }
  }
}

#----- ausleihe -----

if($preamble[1] eq 'Z30-1') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'F') {
      $laenge = length($wort);
      $ausleihe = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET ausleihe = '$ausleihe' WHERE karten_id = '$karten_id' " ) or die
      "Problems with ausleihe $ausleihe entry";
    }
  }
}

#----- blattgroesse -----

if($preamble[1] eq '300') {
  foreach $wort (@nachdollar) {
    $ifwert = substr($wort, 0, 1);
    if($ifwert eq 'c') {
      $laenge = length($wort);
      $blattgroesse = substr($wort, 1, $laenge);
      #----- einfüegen in datenbank -----
      $src=$dbh->do("UPDATE karten2 SET blattgroesse = '$blattgroesse' WHERE karten_id = '$karten_id' " )
      or die "Problems with blattgroesse $blattgroesse entry";
    }
  }
}

```

```

    }
}

#----- sonst_blattinhalt -----

if($preamble[1] eq '300') {
    foreach $wort (@nachdollar) {
        $ifwert = substr($wort, 0, 1);
        if($ifwert eq 'e') {
            $laenge = length($wort);
            $sonst_blattinhalt = substr($wort, 1, $laenge);
            #----- einfüegen in datenbank -----
            $src=$dbh->do("UPDATE karten2 SET sonst_blattinhalt = '$sonst_blattinhalt' WHERE karten_id =
                '$karten_id' " ) or die "Problems with sonst_blattinhalt $sonst_blattinhalt entry";
        }
    }
}

#----- farben -----

if($preamble[1] eq '300') {
    foreach $wort (@nachdollar) {
        $ifwert = substr($wort, 0, 1);
        if($ifwert eq 'b') {
            $laenge = length($wort);
            $farben = substr($wort, 1, $laenge);
            #----- einfüegen in datenbank -----
            $src=$dbh->do("UPDATE karten2 SET farben = '$farben' WHERE karten_id = '$karten_id' " ) or die "Problems
                with farben $farben entry";
        }
    }
}

#----- sprache -----

if($preamble[1] eq '008') {
    foreach $wort (@nachbalken) {
        $laenge = length($wort);
        $sprache = substr($wort, 0, $laenge-2);
        #----- einfüegen in datenbank -----
        $src=$dbh->do("UPDATE karten2 SET sprache = '$sprache' WHERE karten_id = '$karten_id' " ) or die
            "Problems with sprache $sprache entry";
    }
}

#----- sonst_info -----

if($preamble[1] eq '500') {
    foreach $wort (@nachdollar) {
        $ifwert = substr($wort, 0, 1);
        if($ifwert eq 'a') {
            $laenge = length($wort);
            $sonst_info = substr($wort, 1, $laenge);
            #----- einfüegen in datenbank -----
            $src=$dbh->do("UPDATE karten2 SET sonst_info = '$sonst_info' WHERE karten_id = '$karten_id' " ) or
                die "Problems with sonst_info $sonst_info entry";
        }
    }
}

#----- ende untersuchen der leitzahlen

}

print "arbeit erledigt (hoffentlich) \n";

#----- abmelden -----

$dbh->disconnect;
close(DAT);

exit;

```

## 7.2 map\_search.pl

```
#!/usr/bin/perl

use CGI;
use DBI;
$q = new CGI;                                     # zugriff auf sich selbst, q ist fenster
my $script_name = $q->script_name;
my $path_info = $q->path_info;
my $server = "karpcl3";
my $database = "kartensammlung";
my $data_source = "DBI:mysql:$database:$server";
my $username = "karten";
my $password = "sammlung";
my $reply = $ARGV[0];

my $dbh = DBI->connect( $data_source, $username, $password) or die "Can't connect to $data_source: $dbh-
>errstr\n";

#----- Prelude -----

print $q->header;                                # erstellung des heads
print $q->start_html(-title=>'Suchmaske der ETH-Kartensammlung',      # head infos und anfang des bodys
                    -author=>'Schlenker / IKA-ETHZ',
                    -BGCOLOR=>'white',
                    -font=>'Arial',
                    -script=>$jscript);

#----- Transfer-Formulare (Hidden-Fields) -----

print $q->startform(-name=>"Abfrageformular",
                  -action=>"$script_name"
                  );

print $q->hidden(-name=>'karten_id');

print $q->hidden(-name=>'svg_karte',-value=>['0']);

print $q->hidden(-name=>'xGradAusdehnung',-value=>['360']);
print $q->hidden(-name=>'xGradAnfang',-value=>['-170']);
print $q->hidden(-name=>'yGradAusdehnung',-value=>['180']);
print $q->hidden(-name=>'yGradAnfang',-value=>['-90']);

print $q->hidden(-name=>'west',-value=>['-200']);
print $q->hidden(-name=>'ost',-value=>['200']);
print $q->hidden(-name=>'nord',-value=>['100']);
print $q->hidden(-name=>'sued',-value=>['-100']);

print $q->hidden(-name=>'massstab1',-value=>['99999999']);
print $q->hidden(-name=>'merkerMassstab1',-value=>['0']);
print $q->hidden(-name=>'massstab2',-value=>['5000000']);
print $q->hidden(-name=>'merkerMassstab2',-value=>['2']);

print $q->hidden(-name=>'thema',-value=>['%']);
print $q->hidden(-name=>'merkerThema',-value=>['0']);

print $q->hidden(-name=>'jahr',-value=>['%']);
print $q->hidden(-name=>'merkerJahr',-value=>['0']);
print $q->hidden(-name=>'jahrMax',-value=>['9999']);

# print $q->hidden(-name=>'merkerSuche',-value=>['0']);

print $q->hidden(-name=>'herausgeber',-value=>['%']);
print $q->hidden(-name=>'merkerHerausgeber',-value=>['0']);

print $q->hidden(-name=>'sprache',-value=>['%']);
print $q->hidden(-name=>'merkerSprache',-value=>['0']);

print $q->hidden(-name=>'bibliothek',-value=>['%']);
print $q->hidden(-name=>'merkerBibliothek',-value=>['0']);

print $q->endform;
#
```

```

#----- Datentransfer -----

my $karten_titel = $q->param('karten_id');
my $svg_karte = $q->param('svg_karte');

my $west = $q->param('west');
my $ost = $q->param('ost');
my $nord = $q->param('nord');
my $sued = $q->param('sued');
if ($nord < $sued) {
    $breiteChange = $nord;
    $nord = $sued;
    $sued = $breiteChange;
}
if ($ost < $west) {
    $laengeChange = $ost;
    $ost = $west;
    $west = $laengeChange;
}

my $massstab1 = $q->param('massstab1');
my $massstab2 = $q->param('massstab2');
if ($massstab1 < $massstab2) {
    $massstabChange = $massstab1;
    $massstab1 = $massstab2;
    $massstab2 = $massstabChange;
}

my $thema = $q->param('thema');

my $jahr = $q->param('jahr');
my $jahrMax = $q->param('jahrMax');

my $herausgeber = $q->param('herausgeber');

my $sprache = $q->param('sprache');

my $bibliothek = $q->param('bibliothek');

$sth = $dbh->prepare("SELECT karten_id,karten_titel,herausgeber,jahr,massstab,thema FROM karten2
                        WHERE west >= '$west'
                        AND ost <= '$ost'
                        AND nord <= '$nord'
                        AND sued >= '$sued'
                        AND massstab <= '$massstab1'
                        AND massstab >= '$massstab2'
                        AND thema LIKE '$thema'
                        AND jahr >= '$jahr'
                        AND jahr <= '$jahrMax'
                        AND herausgeber LIKE '%$herausgeber%'
                        AND sprache LIKE '$sprache'
                        AND bibliothek LIKE '%$bibliothek'
                        ORDER BY jahr DESC
                        ");

$resp = $sth->execute;
$table = $sth->fetchall_arrayref;

#----- Titel -----

print<<EOF;

<!-- Form für den Submit-Button
<form name="Abfrageformular">
<input type="text" name="formInput">
<a href="javascript:document.theForm.submit();">Submit</a>
</form> -->

<table border="0" cellpadding="0" cellspacing="0" width="964" height="100%">

<!-- Kopfzeile -->
    <tr height="100">

```

```

    <td width="964" height="100"><embed src="http://www.karto.ethz.ch/~schlenker/svg/kopf.svg" width="964"
height="100" type="image/svg+xml" align="left"></td>
</tr>

<tr valign="top" align="left" height="700">
  <td width="964" height="100%" valign="top" align="left">
    <table border="0" cellpadding="0" cellspacing="0" width="964" height="100%">
      <tr valign="top" align="left" height="100%">

        <!--Navigationsspalte -->
        <td width="164" height="100%" bgcolor="#0000CC">
          <embed
src="http://www.karto.ethz.ch/~schlenker/svg/navigation.svg" width="164" height="580" type="image/svg+xml">
          </td>

          <td height="100%">
            <table border="0" cellpadding="0" cellspacing="0"
width="800" height="100%">

              <tr height="500">

                <!--Karte -->
                <td height="500" width="800">

EOF

#----- uebersichtskarten -----

if($svg_karte==0){
print<<EOF;
  <embed src="http://www.karto.ethz.ch/~schlenker/svg/world.svg" width="800" height="500"
type="image/svg+xml"></td>
EOF
}

if($svg_karte==9){
print<<EOF;
  <embed src="http://www.karto.ethz.ch/~schlenker/svg/deutschland.svg" width="800" height="500"
type="image/svg+xml"></td>
EOF
}

if($svg_karte==1 || $svg_karte==2 || $svg_karte==3 || $svg_karte==4 || $svg_karte==5 || $svg_karte==6 ||
$svg_karte==7 || $svg_karte==8
|| $svg_karte==10 || $svg_karte==11 || $svg_karte==12 || $svg_karte==13 || $svg_karte==14 ||
$svg_karte==15 || $svg_karte==16
|| $svg_karte==17 || $svg_karte==18 || $svg_karte==19 || $svg_karte==20 || $svg_karte==21){
print<<EOF;
  <embed src="http://www.karto.ethz.ch/~schlenker/svg/placeholder.svg" width="800" height="500"
type="image/svg+xml"></td>
EOF
}

print<<EOF;

<tr height="120">

  <!--Trefferliste -->

  <td width="800"

    <table border="1"

      <thead <tr>

        <td><b>\#</b></td>

        <td><b>Kartentitel</b></td>

        <td><b>Herausgeber</b></td>

        <td><b>Jahr</b></td>

        <td><b>Massstab 1 :</b></td>

        <td><b>Thema</b></td>

      </tr>

    </thead>

```

```

EOF

$index=0;
#----- Herausschreiben der Daten -----
for $i ( 0 .. ${#table} ) {
    $index++;
    print "<td>$index</td>";
    for $j ( 0 .. ${table->[$i]} ) {
        if($j!=1){
            if($j!=0){
                if(($table->[$i][$j]) eq ""){print "<td> - </td>";}
                else {print "<td> ${table->[$i][$j]} </td>";}
            }
        }
        elseif($j==1){
            if(($table->[$i][$j]) eq ""){print "<td> - </td>";}
            else {
                $id=$table->[$i][0];
                $url="/schlenk-cgi/detail.pl/?$id";
                print "<td> <a href=$url target=\"_blank\">${table->[$i][1]}</a> </td>";}
        }
    }
    print "</tr>";
}
print<<EOF;

</table>
</td>
</tr>
</table>
</td>
</tr>
</table>

EOF

$dbh->disconnect;
print $q->end_html; # abschluss des bodys

exit;

```

## 7.3 init.js (Auswahl)

```
var svgdoc;

// ----- POP UP MENUES -----

// --- Menue-Inhalte ---

var sLselections0 = new Array("-- WELT --","AFRIKA","ANTARKTIS","ARKTIS","ASIEN","AUSTRALIEN","EUROPA","Balkan","Benelux","Deutschland","Frankreich","Britische Inseln","Italien","Oesterreich","Polen/Tschechei/Slowak.,","Schweiz","Skandinavien","Spanien/Portugal","NORDAMERIKA","Kanada","USA (ohne Alaska)","SUED-/MITTELAMERIKA");
var sLselections1 = new Array("> 1 : 10.000.000","1 : 10.000.000","1 : 5.000.000","1 : 1.000.000","1 : 500.000","1 : 200.000","1 : 100.000","1 : 50.000","1 : 25.000","1 : 10.000","< 1 : 10.000");
var sLselections2 = new Array("> 1 : 10.000.000","1 : 10.000.000","1 : 5.000.000","1 : 1.000.000","1 : 500.000","1 : 200.000","1 : 100.000","1 : 50.000","1 : 25.000","1 : 10.000","< 1 : 10.000");
var sLselections3 = new Array("-- ALLE --","Allgemeine Atlanten","Alte Karten","Anthropologische Karten","Archaeologische Karten","Architekturkarten","Astronomische Karten","Bathymetrische Karten","Bergbaukarten","Bevoelkerungskarten","Bildhafte Kartendarstell.,","Bildungswesen","Bodenkarten","Eisenbahnbau-/ Strassenb.,","Eisenbahnkarten","Energiewirtschaftliche K.,","Ernaehrungskarten","Ethnograph./ Ethnolog K.,","Fischerei-/ Jagdkarten","Freizeitkarten","Geodaetische Karten","Geograph./ Landeskundl. K.,","Geologische Karten","Geomagnetische Karten","Geomorphologische Karten","Geotechn./ Baugrunderkarten","Geschichtskarten","Glaziolog./ Glazialgeolog. K.,","Gravimetrische Karten","Hydrolog./-graph./-geol. K.,","Industr./ Gewerbe / Handw.,","Kartenmaterial im Allg.,","Kartogr. Lehrmittel / Schul.,","Klimakarten","Lagerstaetten-/ Bodensch.K.,","Landnutzungskarten","Landwirtschaftskarten","Luft-/ Satellitenbilder","Luftfahrtkarten","Meeresgeologische Karten","Militaer- / Kriegskarten","Mineralogische Karten","Naturgeogr./ Physische K.,","Ozeanographische Karten","Petrographische Karten","Plaene","Politische Karten","Postwesen Karten","Raumplanungs-/ -ordn. K.,","Religions- / Konfessionsk.,","Schiffahrtkarten","Seismische Karten","Siedlungskarten","Sprachgeographische K.,","Stadt- / Ortsplaene","Strassen-/ Strassenverk. K.,","Stratigraphische Karten","Tektonische Karten","Tiergeographische Karten","Topographische Karten","Touristen- / Reisekarten","Uebersichtskarten","Umwelt-/ Naturschutzkarten","Vegetationskarten","Verkehrs- / Transportkarten","Verwaltungskarten","Vulkanologische Karten","Waldkarten","Wanderkarten","Wasserbaukarten","Wassersportkarten","Wirtschaftskarten","Zeitzonekarten");
var sLselections4 = new Array("-- ALLE --","2002","2001","2000","1998","1995","1990","1980","1970","1960","1950","1925","1900","< 1900");
var sLselections5 = new Array("-- ALLE --","Aemter","Institute");
var sLselections6 = new Array("-- ALLE --","Arabisch","Chinesisch","Daenisch","Deutsch","Englisch","Finnisch","Franzoesisch","Griechisch","Hollaendisch","Islaendisch","Italienisch","Japanisch","Lateinisch","Mehrsprachig","Norwegisch","Polnisch","Portugiesisch","Rumaenisch","Russisch","Schwedisch","Spanisch","Tschechisch","Tuerkisch","Ungarisch","Vietnamesisch","Yugoslavisch");
var sLselections7 = new Array("-- ALLE --","ETH BIB","ETH GEO","ETH FOR","ETH KARTO","ZBZ");

// --- initMap ---

function initMap(evt) { //initialize svg references
    svgdoc=evt.getTarget().getOwnerDocument();
}

// ***** Selectionlists Inhalte und Uebergabewerte zur Suche *****

function initNavSellists(evt) {
    //Merker in den Selectionboxes
    var merkerKarte = parent.document.Abfrageformular.svg_karte.value;
    var merkerMassstab1 = parent.document.Abfrageformular.merkerMassstab1.value;
    var merkerMassstab2 = parent.document.Abfrageformular.merkerMassstab2.value;
    var merkerThema = parent.document.Abfrageformular.merkerThema.value;
    var merkerJahr = parent.document.Abfrageformular.merkerJahr.value;
    var merkerHerausgeber = parent.document.Abfrageformular.merkerHerausgeber.value;
    var merkerSprache = parent.document.Abfrageformular.merkerSprache.value;
    var merkerBibliothek = parent.document.Abfrageformular.merkerBibliothek.value;

    //usage: createSelectionList(IndexToStringArray,width,xOffset,yOffset,height(nrOfElements),nrToPreSelect(zero-based!),functionName);
    //create an empty group width Id "selection+indexNr"
    sLcreateSelectionList(0,145,10,60,14,merkerKarte,"uebersicht");
}
```



```

sLcreateSelectionList(1,145,10,120,14,merkerMassstab1,"massstab1");
sLcreateSelectionList(2,145,10,160,14,merkerMassstab2,"massstab2");

sLcreateSelectionList(3,145,10,220,14,merkerThema,"thema");
sLcreateSelectionList(4,145,10,280,14,merkerJahr,"jahr");

sLcreateSelectionList(5,145,10,340,14,merkerHerausgeber,"herausgeber");
sLcreateSelectionList(6,145,10,380,9,merkerSprache,"sprache");
sLcreateSelectionList(7,145,10,420,14,merkerBibliothek,"bibliothek");
}

// --- Menue-Funktionen für die Uebergabewerte ---

// - Alarm-Funktion bei noch nicht bearbeiteten Themen -
function dummy(indexNr,listNr) {
  alert("Sie waelten die Listen-Nr. "+(indexNr)+", "+eval("sLselections"+listNr)[indexNr]+", aus. Diese
    einschraenkende Funktion ist jedoch im Prototyp noch nicht implementiert.");
}

xGradAusdehnung=parent.document.all.Abfrageformular.xGradAusdehnung.value

// - Austausch der Uebersichtskarten und angepassten Default-Werten -

function uebersicht(indexNr,listNr) {
  if (indexNr == 0) { // -- WELT --
    parent.document.Abfrageformular.svg_karte.value = '0';
    parent.document.all.Abfrageformular.west.value = '-200';
    parent.document.all.Abfrageformular.ost.value = '200';
    parent.document.all.Abfrageformular.nord.value = '100';
    parent.document.all.Abfrageformular.sued.value = '-100';
    parent.document.all.Abfrageformular.massstab1.value = '99999999';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '0';
    parent.document.all.Abfrageformular.massstab2.value = '5000000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '2';
    parent.document.all.Abfrageformular.xGradAusdehnung.value = '360';
    parent.document.all.Abfrageformular.xGradAnfang.value = '-170';
    parent.document.all.Abfrageformular.yGradAusdehnung.value = '180';
    parent.document.all.Abfrageformular.yGradAnfang.value = '-90';
    parent.document.all.Abfrageformular.submit();
  }
  if (indexNr == 1) { // AFRIKA
    parent.document.Abfrageformular.svg_karte.value = '1';
    parent.document.all.Abfrageformular.west.value = '-20';
    parent.document.all.Abfrageformular.ost.value = '50';
    parent.document.all.Abfrageformular.nord.value = '40';
    parent.document.all.Abfrageformular.sued.value = '-35';
    parent.document.all.Abfrageformular.massstab1.value = '10000000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '1';
    parent.document.all.Abfrageformular.massstab2.value = '100000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '6';
    parent.document.all.Abfrageformular.submit();
  }
  ...
  ...
  ...

  if (indexNr == 9) { // Deutschland
    parent.document.Abfrageformular.svg_karte.value = '9';
    parent.document.all.Abfrageformular.west.value = '6';
    parent.document.all.Abfrageformular.ost.value = '15';
    parent.document.all.Abfrageformular.nord.value = '55';
    parent.document.all.Abfrageformular.sued.value = '47.3';
    parent.document.all.Abfrageformular.massstab1.value = '5000000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '2';
    parent.document.all.Abfrageformular.massstab2.value = '25000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '8';
    parent.document.all.Abfrageformular.xGradAusdehnung.value = '12';
    parent.document.all.Abfrageformular.xGradAnfang.value = '5.5';
    parent.document.all.Abfrageformular.yGradAusdehnung.value = '9.7';
    parent.document.all.Abfrageformular.yGradAnfang.value = '-55';
    parent.document.all.Abfrageformular.submit();
  }
  ...
  ...
  ...

```

```

// - Massstab1 (von) -
function massstab1(indexNr,listNr) {
  if (indexNr == 0) { parent.document.all.Abfrageformular.massstab1.value = '99999999';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '0';}
  if (indexNr == 1) { parent.document.all.Abfrageformular.massstab1.value = '10000000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '1';}
  if (indexNr == 2) { parent.document.all.Abfrageformular.massstab1.value = '5000000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '2';}
  if (indexNr == 3) { parent.document.all.Abfrageformular.massstab1.value = '1000000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '3';}
  if (indexNr == 4) { parent.document.all.Abfrageformular.massstab1.value = '500000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '4';}
  if (indexNr == 5) { parent.document.all.Abfrageformular.massstab1.value = '200000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '5';}
  if (indexNr == 6) { parent.document.all.Abfrageformular.massstab1.value = '100000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '6';}
  if (indexNr == 7) { parent.document.all.Abfrageformular.massstab1.value = '50000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '7';}
  if (indexNr == 8) { parent.document.all.Abfrageformular.massstab1.value = '25000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '8';}
  if (indexNr == 9) { parent.document.all.Abfrageformular.massstab1.value = '10000';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '9';}
  if (indexNr == 10) { parent.document.all.Abfrageformular.massstab1.value = '0';
    parent.document.all.Abfrageformular.merkerMassstab1.value = '10';}
}

// - Massstab2 (bis) -
function massstab2(indexNr,listNr) {
  if (indexNr == 0) { parent.document.all.Abfrageformular.massstab2.value = '99999999';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '0';}
  if (indexNr == 1) { parent.document.all.Abfrageformular.massstab2.value = '10000000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '1';}
  if (indexNr == 2) { parent.document.all.Abfrageformular.massstab2.value = '5000000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '2';}
  if (indexNr == 3) { parent.document.all.Abfrageformular.massstab2.value = '1000000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '3';}
  if (indexNr == 4) { parent.document.all.Abfrageformular.massstab2.value = '500000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '4';}
  if (indexNr == 5) { parent.document.all.Abfrageformular.massstab2.value = '200000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '5';}
  if (indexNr == 6) { parent.document.all.Abfrageformular.massstab2.value = '100000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '6';}
  if (indexNr == 7) { parent.document.all.Abfrageformular.massstab2.value = '50000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '7';}
  if (indexNr == 8) { parent.document.all.Abfrageformular.massstab2.value = '25000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '8';}
  if (indexNr == 9) { parent.document.all.Abfrageformular.massstab2.value = '10000';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '9';}
  if (indexNr == 10) { parent.document.all.Abfrageformular.massstab2.value = '0';
    parent.document.all.Abfrageformular.merkerMassstab2.value = '10';}
}

// - Kartenthema -
function thema(indexNr,listNr) {
  if (indexNr == 0) { parent.document.all.Abfrageformular.thema.value = '%';
    parent.document.all.Abfrageformular.merkerThema.value = '0';}
  if (indexNr == 1) { parent.document.all.Abfrageformular.thema.value = 'ALLGEMEINE ATLANTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '1';}
  if (indexNr == 2) { parent.document.all.Abfrageformular.thema.value = 'ALTE KARTEN (VOR 1850)';
    parent.document.all.Abfrageformular.merkerThema.value = '2';}
  if (indexNr == 3) { parent.document.all.Abfrageformular.thema.value = 'ANTHROPOLOGISCHE KARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '3';}
  if (indexNr == 4) { parent.document.all.Abfrageformular.thema.value = 'ARCHAEOLOGISCHE KARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '4';}
  if (indexNr == 5) { parent.document.all.Abfrageformular.thema.value = 'ARCHITEKTURKARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '5';}
  if (indexNr == 6) { parent.document.all.Abfrageformular.thema.value = 'ASTRONOMISCHE KARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '6';}
  if (indexNr == 7) { parent.document.all.Abfrageformular.thema.value = 'BATHYMETRISCHE KARTEN (KARTENMATERIAL)';
    parent.document.all.Abfrageformular.merkerThema.value = '7';}
  if (indexNr == 8) { parent.document.all.Abfrageformular.thema.value = 'BERGBAUKARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '8';}
  if (indexNr == 9) { parent.document.all.Abfrageformular.thema.value = 'BEVOELKERUNGSKARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '9';}
  if (indexNr == 10) { parent.document.all.Abfrageformular.thema.value = 'BILDHAFTE KARTENDARSTELLUNGEN
    (PANORAMEN, BILDKARTEN)'; parent.document.all.Abfrageformular.merkerThema.value = '10';}
  ...
  ...

```

```

...
if (indexNr == 70) { parent.document.all.Abfrageformular.thema.value = 'WASSERSPORTKARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '70';}
if (indexNr == 71) { parent.document.all.Abfrageformular.thema.value = 'WIRTSCHAFTSKARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '71';}
if (indexNr == 72) { parent.document.all.Abfrageformular.thema.value = 'ZEITZONENKARTEN';
    parent.document.all.Abfrageformular.merkerThema.value = '72';}
}

// - Erscheinungsjahr -
function jahr(indexNr,listNr) {
    if (indexNr == 0) { parent.document.all.Abfrageformular.jahr.value = '%';
        parent.document.all.Abfrageformular.merkerJahr.value = '0';}
    if (indexNr == 1) { parent.document.all.Abfrageformular.jahr.value = '2002';
        parent.document.all.Abfrageformular.merkerJahr.value = '1';}
    if (indexNr == 2) { parent.document.all.Abfrageformular.jahr.value = '2001';
        parent.document.all.Abfrageformular.merkerJahr.value = '2';}
    if (indexNr == 3) { parent.document.all.Abfrageformular.jahr.value = '2000';
        parent.document.all.Abfrageformular.merkerJahr.value = '3';}
    if (indexNr == 4) { parent.document.all.Abfrageformular.jahr.value = '1998';
        parent.document.all.Abfrageformular.merkerJahr.value = '4';}
    if (indexNr == 5) { parent.document.all.Abfrageformular.jahr.value = '1995';
        parent.document.all.Abfrageformular.merkerJahr.value = '5';}
    if (indexNr == 6) { parent.document.all.Abfrageformular.jahr.value = '1990';
        parent.document.all.Abfrageformular.merkerJahr.value = '6';}
    if (indexNr == 7) { parent.document.all.Abfrageformular.jahr.value = '1980';
        parent.document.all.Abfrageformular.merkerJahr.value = '7';}
    if (indexNr == 8) { parent.document.all.Abfrageformular.jahr.value = '1970';
        parent.document.all.Abfrageformular.merkerJahr.value = '8';}
    if (indexNr == 9) { parent.document.all.Abfrageformular.jahr.value = '1960';
        parent.document.all.Abfrageformular.merkerJahr.value = '9';}
    if (indexNr == 10) { parent.document.all.Abfrageformular.jahr.value = '1950';
        parent.document.all.Abfrageformular.merkerJahr.value = '10';}
    if (indexNr == 11) { parent.document.all.Abfrageformular.jahr.value = '1925';
        parent.document.all.Abfrageformular.merkerJahr.value = '11';}
    if (indexNr == 12) { parent.document.all.Abfrageformular.jahr.value = '1900';
        parent.document.all.Abfrageformular.merkerJahr.value = '12';}
    if (indexNr == 13) { parent.document.all.Abfrageformular.jahr.value = '0';
        parent.document.all.Abfrageformular.merkerJahr.value = '13';}

    if (indexNr == 13) { parent.document.all.Abfrageformular.jahrMax.value = '1899';}
    else { parent.document.all.Abfrageformular.jahrMax.value = '9999';}
}

// ----- ERWEITERTE SUCHE -----

function SuchArt() {
    MyNode=svgdoc.getElementById("erweiterteSuche");
    if (MyNode.getAttribute("visibility") == "hidden") {
        svgdoc.getElementById("SuchText").getFirstChild().setData("einfache Suche");
        MyNode.setAttribute("visibility","visible");
    }
    else {
        svgdoc.getElementById("SuchText").getFirstChild().setData("erweiterte Suche");
        MyNode.setAttribute("visibility","hidden");
    }
}

// - Herausgeber -
function herausgeber(indexNr,listNr) {
    if (indexNr == 0) { parent.document.all.Abfrageformular.herausgeber.value = '%';
        parent.document.all.Abfrageformular.merkerHerausgeber.value = '0';}
    if (indexNr == 1) { parent.document.all.Abfrageformular.herausgeber.value = 'amt';
        parent.document.all.Abfrageformular.merkerHerausgeber.value = '1';}
    if (indexNr == 2) { parent.document.all.Abfrageformular.herausgeber.value = 'inst';
        parent.document.all.Abfrageformular.merkerHerausgeber.value = '2';}
}

// - Sprache -
function sprache(indexNr,listNr) {
    if (indexNr == 0) { parent.document.all.Abfrageformular.sprache.value = '%';
        parent.document.all.Abfrageformular.merkerSprache.value = '0';}
    if (indexNr == 1) { parent.document.all.Abfrageformular.sprache.value = 'ara';

```

```

parent.document.all.Abfrageformular.merkerSprache.value = '1';
if (indexNr == 2) { parent.document.all.Abfrageformular.sprache.value = 'chi';
  parent.document.all.Abfrageformular.merkerSprache.value = '2';}
if (indexNr == 3) { parent.document.all.Abfrageformular.sprache.value = 'dan';
  parent.document.all.Abfrageformular.merkerSprache.value = '3';}
if (indexNr == 4) { parent.document.all.Abfrageformular.sprache.value = 'ger';
  parent.document.all.Abfrageformular.merkerSprache.value = '4';}
if (indexNr == 5) { parent.document.all.Abfrageformular.sprache.value = 'eng';
  parent.document.all.Abfrageformular.merkerSprache.value = '5';}
if (indexNr == 6) { parent.document.all.Abfrageformular.sprache.value = 'fin';
  parent.document.all.Abfrageformular.merkerSprache.value = '6';}
if (indexNr == 7) { parent.document.all.Abfrageformular.sprache.value = 'fre';
  parent.document.all.Abfrageformular.merkerSprache.value = '7';}
if (indexNr == 8) { parent.document.all.Abfrageformular.sprache.value = 'grc';
  parent.document.all.Abfrageformular.merkerSprache.value = '8';}
if (indexNr == 9) { parent.document.all.Abfrageformular.sprache.value = 'dut';
  parent.document.all.Abfrageformular.merkerSprache.value = '9';}
if (indexNr == 10) { parent.document.all.Abfrageformular.sprache.value = 'ice';
  parent.document.all.Abfrageformular.merkerSprache.value = '10';}
...
...
...
if (indexNr == 25) { parent.document.all.Abfrageformular.sprache.value = 'vie';
  parent.document.all.Abfrageformular.merkerSprache.value = '25';}
if (indexNr == 26) { parent.document.all.Abfrageformular.sprache.value = 'scc';
  parent.document.all.Abfrageformular.merkerSprache.value = '26';}
}

// - Bibliothek -
function bibliothek(indexNr,listNr) {
  if (indexNr == 0) { parent.document.all.Abfrageformular.bibliothek.value = '%';
    parent.document.all.Abfrageformular.merkerBibliothek.value = '0';}
  if (indexNr == 1) { parent.document.all.Abfrageformular.bibliothek.value = 'BIB';
    parent.document.all.Abfrageformular.merkerBibliothek.value = '1';}
  if (indexNr == 2) { parent.document.all.Abfrageformular.bibliothek.value = 'GEO';
    parent.document.all.Abfrageformular.merkerBibliothek.value = '2';}
  if (indexNr == 3) { parent.document.all.Abfrageformular.bibliothek.value = 'FOR';
    parent.document.all.Abfrageformular.merkerBibliothek.value = '3';}
  if (indexNr == 4) { parent.document.all.Abfrageformular.bibliothek.value = 'KARTO';
    parent.document.all.Abfrageformular.merkerBibliothek.value = '4';}
  if (indexNr == 5) { parent.document.all.Abfrageformular.bibliothek.value = 'ZBZ';
    parent.document.all.Abfrageformular.merkerBibliothek.value = '5';}
}

// ***** Koordinaten aus den Karten auslesen und umrechnen *****

//global variables
var svgObjX;
var svgObjY;
var svgSVGObj;
var svgDoc;
var ulXCorner;
var ulYCorner;
var origPixSize;
var pixSize;
var offsetX;
var offsetY;
var yMinusVal;
var refCoordRect;
var ratioY;
var pixHeight;
var pixWidth;
var scale;
var trans;
var transx;
var transy;
var width;
var height;

var xGradAusdehnung;
var xGradAnfang;
var yGradAusdehnung;
var yGradAnfang;

var xCoordRelativ;
var yCoordRelativ;
var xCoordAbsolut;

```

```

var yCoordAbsolut;

// ----- Funktion initCoords(evt) -----

function initCoords(evt) {
  // Retrieve the SVG document object:
  var directTarget = evt.getTarget();
  if( directTarget.getNodeType() != 9 ) // if not DOCUMENT_NODE
    svgDoc = directTarget.getOwnerDocument();
  else
    svgDoc = directTarget;

  //get reference to text-Element
  svgObjX = svgDoc.getElementById("coordx");
  svgObjY = svgDoc.getElementById("coordy");
  //get reference to text within text-Element
  svgObjX = svgObjX.getFirstChild();
  svgObjY = svgObjY.getFirstChild();
  svgSVGObj = svgDoc.getDocumentElement();
  //reference to coordinate box (for later scaling and translating)
  refCoordRect = svgDoc.getElementById("groupShowCoords");
  //initialize coordinates, pixsize etc.
  var viewBox = new String(svgSVGObj.getAttribute("viewBox"));
  var viewboxes = viewBox.split(' ');
  ulXCorner = viewboxes[0];
  ulYCorner = viewboxes[1];
  width = viewboxes[2];
  height = viewboxes[3];

  pixWidth = svgSVGObj.getAttribute("width");
  pixHeight = svgSVGObj.getAttribute("height");
  origPixSize = width / pixWidth;

  //a value to subtract Y-Values, because of inverted y-axis
  yMinusVal = 0;

  //determine ratio for coordinate box placement along y-axis
  translateY = getTranslate("groupShowCoords","y");
  ratioY = translateY / height;

  //call resetCoords();
  resetCoords();
}

// ----- Funktion resetCoords() -----

function resetCoords() {
  //get current zoom and pan values
  scale = svgSVGObj.getCurrentScale();
  trans = svgSVGObj.getCurrentTranslate();
  transx = trans.getX();
  transy = trans.getY();

  //reset offset-values and pixSize according to current scale and translate
  pixSize = origPixSize / scale;
  offsetX = parseFloat(ulXCorner) - transx * pixSize;
  offsetY = parseFloat(ulYCorner) - transy * pixSize;
}

// ----- Funktion showCoords(evt) -----

function showCoords(evt) {
  var htmlHeight = 500;
  var htmlWidth = 800;

  if ( (htmlHeight / htmlWidth) > (height / width) ) {
    var picHeight = height * (htmlWidth / width);
    var picWidth = htmlWidth;
  }
  else {
    var picWidth = width * (htmlHeight / height);
    var picHeight = htmlHeight;
  }

  var xVersatz = (htmlWidth - picWidth) / 2 ;
  var yVersatz = (htmlHeight - picHeight) / 2 ;

```

```

//to get coordinates
xCoordRelativ = (((offsetX + evt.getClientX() * pixSize) - xVersatz) / picWidth);
yCoordRelativ = (((offsetY + evt.getClientY() * pixSize) - yVersatz) / picHeight);

xGradAusdehnung = parent.document.all.Abfrageformular.xGradAusdehnung.value;
xGradAnfang = parent.document.all.Abfrageformular.xGradAnfang.value;
yGradAusdehnung = parent.document.all.Abfrageformular.yGradAusdehnung.value;
yGradAnfang = parent.document.all.Abfrageformular.yGradAnfang.value;

xGradAusdehnungFloat = parseFloat(xGradAusdehnung);
xGradAnfangFloat = parseFloat(xGradAnfang);
yGradAusdehnungFloat = parseFloat(yGradAusdehnung);
yGradAnfangFloat = parseFloat(yGradAnfang);

xCoordAbsolut = xCoordRelativ * xGradAusdehnungFloat + xGradAnfangFloat;
yCoordAbsolut = (yCoordRelativ * yGradAusdehnungFloat + yGradAnfangFloat) * (-1);

}

// ----- Funktion anfangCoords(evt) -----

function anfangCoords() {
  //alert("laenge: "+(xCoordAbsolut)+" breite: "+(yCoordAbsolut));
  parent.document.all.Abfrageformular.nord.value = yCoordAbsolut;
  parent.document.all.Abfrageformular.west.value = xCoordAbsolut;
}

// ----- Funktion showCoords(evt) -----

function endCoords() {
  //alert("laenge: "+(xCoordAbsolut)+" breite: "+(yCoordAbsolut));
  parent.document.all.Abfrageformular.sued.value = yCoordAbsolut;
  parent.document.all.Abfrageformular.ost.value = xCoordAbsolut;
}

// ----- Funktion getTranslate(evt) -----

function getTranslate(myElement,xOrY) {
  //get reference to element
  element = svgDoc.getElementById(myElement);

  //first get transform value of coordinate box
  var curTransform = element.getAttribute("transform");
  curTransform = new String(curTransform); //Wert in ein String umwandeln
  //no fear from Regular expressions ... just copy it, I copied it either ...
  var translateRegExp=/translate\(([+-]?[d+](\s*[s,]\s*)([+-]?[d+])\)\s*/;
  //This part extracts the translation-part from the overall transform-string
  if (curTransform.length != 0)
  {
    var result = curTransform.match(translateRegExp);
    if (result == null || result.index == -1)
    {
      var oldTranslateX = 0;
      var oldTranslateY = 0;
    }
    else
    {
      var oldTranslateX = result[1];
      var oldTranslateY = result[3];
    }
  }
  if (xOrY == "x") {
    return oldTranslateX;
  }
  if (xOrY == "y") {
    return oldTranslateY;
  }
}

// *****Sonstige Funktionen (Werkzeuge, Zoomlevels, etc.) *****

// ----- ADAPTIVES ZOOMEN -----

function AdaptivZoom() {
  currentScale=evt.target.ownerDocument.rootElement.currentScale
  MyNodeLevel2=svgdoc.getElementById("ZoomLevel2");

```

```

MyNodeLevel1=svgdoc.getElementById("ZoomLevel1");
MyNodeLevel0=svgdoc.getElementById("ZoomLevel0");
if (currentScale > 3) {
  MyNodeLevel1.setAttribute("display","inline");
  MyNodeLevel0.setAttribute("display","none");
}
else {
  MyNodeLevel2.setAttribute("display","none");
  MyNodeLevel1.setAttribute("display","none");
  MyNodeLevel0.setAttribute("display","inline");
}
if (currentScale > 5) {
  MyNodeLevel2.setAttribute("display","inline");
}
else {
  MyNodeLevel2.setAttribute("display","none");
}
}

// ----- WERKZEUGE -----

// --- Hilfe ---
function Hilfe() {
  alert("(1) geeignete Uebersicht waehlen (2) Suchkriterien einstellen (3) Suchgebiet festlegen (4) Los!");
}

// --- Zoom In ---
function FuncZoomIn() {
  alert("Einzoomen bisher nur ueber 'ctrl'-Taste oder rechte Maustaste moeglich.");
  MyNode=svgdoc.getElementById("zoomInOn");
  if (MyNode.getAttribute("visibility") == "hidden") {
    MyNode.setAttribute("visibility","visible");
    MyNode2=svgdoc.getElementById("zoomOutOn");
    MyNode2.setAttribute("visibility","hidden");
    MyNode3=svgdoc.getElementById("moveOn");
    MyNode3.setAttribute("visibility","hidden");
    MyNode4=svgdoc.getElementById("auswahlOn");
    MyNode4.setAttribute("visibility","hidden");
  }
}

// --- Zoom Out ---
function FuncZoomOut() {
  alert("Auszoomen bisher nur ueber 'ctrl'+ 'shift'-Taste oder rechte Maustaste moeglich.");
  MyNode=svgdoc.getElementById("zoomOutOn");
  if (MyNode.getAttribute("visibility") == "hidden") {
    MyNode.setAttribute("visibility","visible");
    MyNode2=svgdoc.getElementById("zoomInOn");
    MyNode2.setAttribute("visibility","hidden");
    MyNode3=svgdoc.getElementById("moveOn");
    MyNode3.setAttribute("visibility","hidden");
    MyNode4=svgdoc.getElementById("auswahlOn");
    MyNode4.setAttribute("visibility","hidden");
  }
}

// --- Move ---
function FuncMove() {
  alert("Karte verschieben bisher nur ueber 'alt'-Taste moeglich.");
  MyNode=svgdoc.getElementById("moveOn");
  if (MyNode.getAttribute("visibility") == "hidden") {
    MyNode.setAttribute("visibility","visible");
    MyNode2=svgdoc.getElementById("zoomInOn");
    MyNode2.setAttribute("visibility","hidden");
    MyNode3=svgdoc.getElementById("zoomOutOn");
    MyNode3.setAttribute("visibility","hidden");
    MyNode4=svgdoc.getElementById("auswahlOn");
    MyNode4.setAttribute("visibility","hidden");
  }
}

// --- Auswahlrechteck zeichnen ---
function FuncAuswahl() {
  alert("Suchgebiet wird von der mit der Maus auf der Karte aufgezogenen Flaechе uebernommen. (MouseDown
    bis MouseUp)");
  MyNode=svgdoc.getElementById("auswahlOn");

```

```
if (MyNode.getAttribute("visibility") == "hidden") {
    MyNode.setAttribute("visibility", "visible");
    MyNode2=svgdoc.getElementById("zoomInOn");
    MyNode2.setAttribute("visibility", "hidden");
    MyNode3=svgdoc.getElementById("zoomOutOn");
    MyNode3.setAttribute("visibility", "hidden");
    MyNode4=svgdoc.getElementById("moveOn");
    MyNode4.setAttribute("visibility", "hidden");
}
}

// --- Submit-Funktion ---
function execute(){
    parent.document.all.Abfrageformular.submit();
    // alert("uebergabewert von jahr: "+(parent.document.Abfrageformular.jahr.value));
}

// --- Detailsuche ---
function FuncDetailsuche() {
    alert("Detailsuche nicht implementiert");
}
```



## 7.4 world.svg (Auswahl)

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd"
[
  <!ENTITY st0 "font-family:'Helvetica-BoldOblique';">
  <!ENTITY st1 "font-size:2.94;">
  <!ENTITY st2 "font-size:3.36;">
  <!ENTITY st3 "font-size:3.64;">
  <!ENTITY st4 "font-size:3.0799;">
  <!ENTITY st5 "fill:none;stroke:#FFFFFF;stroke-width:0.3;stroke-linejoin:round;stroke-dasharray:0.9 0.6;">
  <!ENTITY st6 "font-family:'Helvetica-Bold';">
  <!ENTITY st7 "font-size:3;">
  <!ENTITY st8 "font-size:6;">
  <!ENTITY st9 "fill:none;stroke:#000000;stroke-width:0.35;">
  <!ENTITY st10 "font-size:2.5;">
  <!ENTITY st11 "font-size:2.8;">
  <!ENTITY st12 "font-size:7.5;">
  <!ENTITY st13 "fill:#F5CB99;stroke:#FFFFFF;stroke-width:0.3;stroke-linecap:round;stroke-linejoin:round;">
  <!ENTITY st14 "fill:none;stroke:#FFFFFF;stroke-width:0.3;stroke-linecap:round;stroke-linejoin:round;stroke-
    dasharray:0 1;">
  <!ENTITY st15 "fill:#9BCAE1;">
  <!ENTITY st16 "fill:#9BCAE1;stroke:#FFFFFF;stroke-width:0.5;stroke-linecap:round;stroke-linejoin:round;">
  <!ENTITY st17 "fill:#323366;stroke:none;">
  <!ENTITY st18 "letter-spacing:0;">
  <!ENTITY st19 "letter-spacing:0.7;">
  <!ENTITY st20 "letter-spacing:1.5;">
  <!ENTITY st21 "letter-spacing:3.6;">
  <!ENTITY st22 "font-family:'ArialMT';">
  <!ENTITY st23 "letter-spacing:-0.1;">
  <!ENTITY st24 "letter-spacing:-0.2;">
  <!ENTITY st25 "letter-spacing:-0.3;">
  <!ENTITY st26 "letter-spacing:33.8;">
  <!ENTITY st27 "opacity:0.3;fill:#1E5FA9;stroke:#1E5FA9;stroke-width:0.5;">
  <!ENTITY st28 "fill:none;stroke:#FFFFFF;stroke-width:0.3;stroke-linecap:round;stroke-linejoin:round;">
]
>

<svg width="720" height="360" viewBox="0 0 720 360" xml:space="preserve"
  onload="initMap(evt);initCoords(evt)"
  onzoom="AdaptivZoom();resetCoords()"
  onscroll="resetCoords()"
  onmousemove="showCoords(evt)"
  onmousedown="anfangCoords()"
  onmouseup="endCoords()"
  >

<script xlink:href="init.js" type="text/ecmascript" />

<g id="Water">
  <path style="&st15;" d="M720.2,360.2H0.2V0.2h720.1v360.2z"/>
</g>
<g id="Land">
  <path style="&st13;" d="M245.9,161-0.5-0.21-1.5-0.21-1.5-0.21-1.7-0.510.4,012.8,0L245.9,16z"/>
  <path style="&st13;" d="M232.8,16.21-2.7-0.31-0.3,010.4-0.410.9,0.110.2,0.112.2,0.3L232.8,16.2z"/>
  <path style="&st13;" d="M452,16.710.5-0.112.6-0.11-0.2,0.1L452,16.7z"/>
  <path style="&st13;" d="M460.6,16.811.1-0.112,0.11-0.6,0.2L460.6,16.8z"/>
  <path style="&st13;" d="M459.7,17z"/>
  <path style="&st13;" d="M459.7,1710.8,0L459.7,17z"/>
  <path style="&st13;" d="M449.9,17.211.7-0.2L449.9,17.2z"/>
  <path style="&st13;" d="M447.3,17.71-0.1-0.212.2,0.112.6-0.11-2.8,0.2L447.3,17.7z"/>
  <path style="&st13;" d="M516.3,17.8z"/>
  ...
  ...
  ...
  <path style="&st5;" d="M486.9,115.711.5-0.510.3-0.811-0.511.2,0.5h0.8h0.510.6,0.5"/>
  <path style="&st5;" d="M484.8,114.51-1.3-1.21-0.5-1.8v-1.410.8-0.41-0.5-0.91-0.6-0.8L482,108v-0.511.3-
    0.9"/>
</g>
<g id="Grids" style="&st27;">
  <path d="M36.3,360.2c0-120,0-240,0-360"/>
  <path d="M96.4,360.2c0-120,0-240,0-360"/>
  ...
  ...
  
```

```

...

<path d="M0.2,60.2c239.3,0,478.6,0,717.8,0c0.7,0,1.3,0,2,0"/>
</g>
...
...
...

<g id="ZoomLevel2" display="none" pointer-events="none">
</g>
<g id="ZoomLevel1" display="none" pointer-events="none">

<g id="OtherNames">
  <text transform="matrix(0.75 0 0 1 26.6816 217.6973)"><tspan x="0" y="0" style="&st17; &st6; &st7;
    &st18;">Society</tspan><tspan x="0" y="3.4" style="&st17; &st6; &st7; &st18;">Arch.</tspan><tspan
    x="0" y="6.7" style="&st17; &st6; &st7; &st18;">(Fr.)</tspan></text>
  <text transform="matrix(0.75 0 0 1 50.0181 235.7178)"><tspan x="0" y="0" style="&st17; &st6; &st7;
    &st18;">Austral Is.</tspan><tspan x="0" y="3.4" style="&st17; &st6; &st7; &st18;">(Fr.)</tspan></text>
  <text transform="matrix(0.75 0 0 1 64.4863 219.2012)"><tspan x="0" y="0" style="&st17; &st6; &st7;
    &st18;">Gambier Is. (Fr.) </tspan></text>
  <text transform="matrix(0.75 0 0 1 59.6616 197.8672)"><tspan x="0" y="0" style="&st17; &st6; &st7;
    &st18;">Marquesas (Fr.)</tspan></text>
  ...
  ...
  ...
  <text x="308.707" y="215.3193"><tspan>St. Helena (U.K.)</tspan></text>
  <text x="703.5068" y="217.3877"><tspan>Niue (N.Z.)</tspan></text>
</g>
</g>

<g id="ZoomLevel0" display="inline" pointer-events="none">
<g id="ContinentNames" style="&st17; &st0; &st12; &st20;">
  <text transform="matrix(1 0 0 1 107.2656 86.2837)"><tspan x="0" y="0">NORTH-</tspan><tspan x="-3.7"
    y="9">AMERICA</tspan></text>
  <text transform="matrix(1 0 0 1 207.1611 204.9043)"><tspan x="0" y="0">SOUTH-</tspan><tspan x="-3.9"
    y="9">AMERICA</tspan></text>
  <text x="359.0508" y="160.9502"><tspan>AFRICA</tspan></text>
  <text x="494.1572" y="87.0195"><tspan>ASIA</tspan></text>
  <text x="573.9775" y="235.0879"><tspan>AUSTRALIA</tspan></text>
  <text x="344.1113" y="85.2715"><tspan>EUROPE</tspan></text>
</g>
</g>

<g id="Antarctica">
  <text transform="matrix(1 0 0 1 194.2378 352.3613)" pointer-events="none"><tspan x="0" y="0" style="&st17;
    &st0; &st12; &st26;">ANTARCTICA</tspan></text>
</g>

<g id="Gridnumbers" style="&st22; &st8;" pointer-events="none">
  <text x="709.541" y="62.1416"><tspan>60°</tspan></text>
  <text x="709.6387" y="122.2085"><tspan>30°</tspan></text>
  ...
  ...
  ...
  <text x="632.085" y="359.2158"><tspan>150°</tspan></text>
  <text x="692.4287" y="359.2158"><tspan>180°</tspan></text>
</g>
<g id="Frame">
  <path style="&st9;" d="M720.1,360.2H0.2V0.2h719.9V360.2z"/>
</g>

<g id="groupShowCoords" transform="translate(0 0)" visibility="hidden">
  <g id="coords" style="font-size:12;">
    <text x="5" y="20">X:</text>
    <text id="coordx" x="25" y="20"> </text>
    <text x="100" y="20">, Y:</text>
    <text id="coordy" x="120" y="20"> </text>
  </g>
</g>
</svg>

```

## 7.5 navigation.svg

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd"
[
  <!ENTITY st0 "fill:#000000;stroke:none;">
  <!ENTITY st1 "font-family:'arial';">
  <!ENTITY st2 "font-size:12;">
  <!ENTITY st3 "fill-rule:nonzero;clip-rule:nonzero;fill:#FFFFFF;stroke-width:1.2;stroke:#000000;stroke-
    miterlimit:4;">
  <!ENTITY st4 "font-size:10;">
]>
<svg width="164pt" height="580pt" viewBox="0 0 164 580" zoomAndPan="disable" xml:space="preserve"
  onload="initMap(evt);initNavSellists(evt)">

<script xlink:href="selectionList.js" type="text/ecmascript" />
<script xlink:href="init.js" type="text/ecmascript" />

// Hintergrund
<rect width="60" height="580" fill="#0000CC" />
<rect x="60" width="104" height="580" fill="#3399FF" />

// Suche Button
<g id="suche">
  <rect x="70" y="5" width="85" height="18" style="&st3;" onclick="SuchArt()"/>
  <text pointer-events="none"><tspan x="112" y="18" text-anchor="middle" style="&st0; &st1; &st4;"
    id="SuchText">erweiterte Suche</tspan></text>
</g>

// Menue-Beschriftung
<g id="beschriftung" pointer-events="none" text-anchor="end" style="&st0; &st1; &st2;">
  <text><tspan x="155" y="55">Detailübersicht</tspan></text>
  <text><tspan x="155" y="115">Massstab von</tspan></text>
  <text><tspan x="155" y="155">bis</tspan></text>
  <text><tspan x="155" y="215">Kartenthema</tspan></text>
  <text><tspan x="155" y="275">Erscheinungsjahr ab</tspan></text>
</g>

// Pop-Up-Menues
<g id="erweiterteSuche" visibility="hidden" pointer-events="none" style="&st0; &st1; &st2;">
  <text><tspan x="155" y="335" text-anchor="end">Herausgeber</tspan></text>
  <text><tspan x="155" y="375" text-anchor="end">Sprache</tspan></text>
  <text><tspan x="155" y="415" text-anchor="end">Bibliothek</tspan></text>
  <g id="sLselectionBox7" />
  <g id="sLselectionBox6" />
  <g id="sLselectionBox5" />
</g>

<g id="sLselectionBox4" />
<g id="sLselectionBox3" />
<g id="sLselectionBox2" />
<g id="sLselectionBox1" />
<g id="sLselectionBox0" />
</svg>
  
```