

Anforderungen an einen kartographischen Viewer für Business Intelligence Systeme

**Diplomarbeit
von Georg Held
Januar 2003**



University of Applied Sciences – FH Karlsruhe

Fachbereich Geoinformationswesen

Studiengang Kartographie und Geomatik

Studienschwerpunkt Geoinformatik

Leiter: Prof. Dr. Peter Freckmann

**Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst
und keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe.
Berlin, Januar 2003**

Inhalt

1	Einleitung	9
1.1	Motivation und Zielsetzung.....	9
1.2	Aufbau der Arbeit	10
2	Anforderungen an die Thematische Kartographie für das Geomarketing	11
2.1	Geomarketing	11
2.1.1	Definition.....	11
2.1.2	Entstehungsgeschichte.....	11
2.1.3	Der Geomarketingprozess	12
2.1.4	Daten für das Geomarketing.....	13
2.1.4.1	Geodaten	13
2.1.4.1.1	Rasterdaten.....	13
2.1.4.1.2	Vektordaten.....	13
2.1.4.2	Sachdaten als thematische Information	13
2.1.4.2.1	Externe Daten	14
2.1.4.2.2	Interne Daten	14
2.1.5	Anwendungen und Methoden.....	14
2.1.5.1	Standortplanung im Geomarketing.....	14
2.1.5.1.1	Standortsuche	14
2.1.5.1.2	Standortbewertung.....	14
2.1.5.2	Database Marketing.....	15
2.1.5.3	Mediaplanung und Mediaselektion.....	15
2.1.5.4	Optimierung von Vertrieb und Service	15
2.1.5.4.1	Bildung von Marktverantwortungsregionen	16
2.1.5.4.2	Tourenplanung.....	16
2.1.6	Anforderungen an Geomarketingsysteme	16
2.2	Thematische Kartographie.....	17
2.2.1	Definition und Begriff	17
2.2.2	Gegenwärtige Kartenherstellung thematischer Karten.....	17
2.2.2.1	Webbasierte Kartenherstellung	18
2.2.2.2	moderne Karten - Webmaps	19
2.2.3	Die Grundlagen thematischer Karten	20
2.2.3.1	Aufgabe und Aussage der Darstellungen	20
2.2.3.2	Eigenschaften der Informationen	21
2.2.3.3	Die Darstellungsmöglichkeiten thematischer Inhalte.....	22
2.2.4	Kartentypen des Geomarketings.....	25
2.2.4.1	Lagekarten.....	25
2.2.4.1.1	Standortkarten	25
2.2.4.1.2	Verbreitungskarten	26
2.2.4.2	Diagrammkarten	26
2.2.4.2.1	Ortsbezogene Diagrammkarte	27
2.2.4.2.2	Streckenbezogene Diagrammkarte	27
2.2.4.2.3	Flächenbezogene Diagrammkarte	27
2.2.4.3	Choroplethenkarten	28
2.2.4.4	Darstellung von Bewegung und Genese	29
2.2.5	Die graphischen Elemente einer thematischen Karte	29

2.2.5.1	Kartenschrift	29
2.2.5.2	Signaturen einer Karte	30
2.2.5.3	Diagramme zur Darstellung von Information	30
2.2.5.3.1	Flächen-/Symbolfigurendiagramme.....	31
2.2.5.3.2	Kreissectorendiagramm.....	31
2.2.5.3.3	Säulen-/Balkendiagramm.....	32
2.2.5.3.4	Stabdiagramme.....	32
2.2.5.3.5	Kurvendiagramme	32
2.2.5.3.6	Banddiagramme.....	33
2.2.5.4	Gruppenbildung in Choroplethenkarten	33
2.2.5.5	Kartengrundlagen	34
2.2.5.5.1	Basisdaten als geographische Information.....	34
2.2.5.5.2	Projektion und Koordinatensystem einer Darstellung.....	35
2.2.5.5.3	Generalisierung von geographischen Objekten.....	35
2.2.5.6	Erläuternde Kartenzusätze	36
2.2.5.6.1	Der Titel einer Karte.....	36
2.2.5.6.2	Legende zur Erklärung des Inhaltes	36
2.2.5.6.3	Maßstabsleiste als Relationsangabe	36
2.2.5.6.4	Übersichtskarte zur räumlichen Orientierung.....	37
2.2.5.6.5	Sonstige Kartenzusätze	37
2.2.6	Zusammenführung aller Kartenelemente	37
2.3	Gestaltung von Bildschirmkarten.....	38
2.3.1	Kriterien einer Bildschirmkarte	38
2.3.1.1	Format und Größe der Darstellungen	38
2.3.1.2	Farbe und Helligkeit	39
2.3.1.3	Schrift und graphische Objekte	40
2.3.1.4	Auftretende Bildstörungen	41
2.3.2	"Neue" Aspekte zur Kartengestaltung	42
2.3.2.1	Inneres Kartendesign.....	42
2.3.2.1.1	Filter	42
2.3.2.1.2	Farbverläufe	42
2.3.2.1.3	Füllmuster.....	42
2.3.2.1.4	Transparenz.....	42
2.3.2.1.5	Animation.....	42
2.3.2.1.6	Interaktive Kartenelemente	43
2.3.2.2	Äußeres Kartendesign.....	43
2.3.2.2.1	Weiterführende Information zu einer Darstellung.....	43
2.3.2.2.2	Räumliche Navigation in thematische Karten.....	43
2.3.2.2.3	Layerkontrolle.....	44
2.3.2.2.4	Suchdialoge.....	44
2.3.2.2.5	Interaktive Kartenfenster	44
2.3.2.2.6	Weiteres Gestaltungselemente	44
2.3.3	Aufbau eines GUI für thematische Karten	45
2.3.3.1	Der Begriff graphische Benutzeroberfläche.....	45
2.3.3.2	Bestandteile eines kartographischen Viewers.....	46
2.3.3.3	Gesamtlayout der Benutzeroberfläche	46
3	Technologien.....	47
3.1	Webtechnologien	47
3.1.1	Vorteil standardisierter Technologien	47
3.1.2	XML und seine Anwendungen.....	48
3.1.2.1	Extensible Markup Language	49
3.1.2.2	Document Object Model.....	50
3.1.2.3	XHTML / HTML	51
3.1.2.4	Cascading Style Sheets.....	51

3.1.2.5	Extensible Stylesheet Language	51
3.1.2.5.1	<i>XSLT</i>	<i>51</i>
3.1.2.5.2	<i>XSL-FO</i>	<i>51</i>
3.1.2.6	Geographic Markup Language.....	52
3.2	SVG – Scalable Vector Graphics	52
3.2.1	Dateiaufbau – Dokumentenstruktur - Elemente	52
3.2.2	Graphische Möglichkeiten von SVG.....	54
3.2.2.1	Kartengeometrien.....	54
3.2.2.1.1	<i>Punkte</i>	<i>54</i>
3.2.2.1.2	<i>Linien</i>	<i>54</i>
3.2.2.1.3	<i>Flächen</i>	<i>54</i>
3.2.2.2	Text mit SVG	55
3.2.2.3	Styles zur Gestaltung	55
3.2.2.4	Koordinatensystem, Transformation und Kartenansichten.....	56
3.2.2.5	Interaktion durch Clientseitiges Scripting	56
3.2.2.6	Animationsmöglichkeiten	57
3.2.2.7	Sonstiges	57
3.2.3	SVG und Kartographie	59
3.2.3.1	Stärken	59
3.2.3.2	Schwächen.....	60
3.2.4	Implementierung.....	61
3.2.4.1	Anwendungen zur Erzeugung von SVG.....	61
3.2.4.1.1	<i>Texteditoren</i>	<i>61</i>
3.2.4.1.2	<i>WYSIWYG Autorensysteme</i>	<i>61</i>
3.2.4.2	Exportmöglichkeiten.....	61
3.2.4.3	Vektorisierungsanwendungen - Druckertreiber.....	62
3.2.4.4	Andere Erzeugungsmöglichkeiten	62
3.2.4.4.1	<i>XSL zur Transformation nach SVG.....</i>	<i>62</i>
3.2.4.4.2	<i>Programmierung.....</i>	<i>62</i>
3.2.4.4.3	<i>Toolkits</i>	<i>63</i>
3.2.4.5	SVG Viewer	63
3.2.4.5.1	<i>Plug-In Lösung</i>	<i>63</i>
3.2.4.5.2	<i>Native Browserimplementierungen.....</i>	<i>64</i>
3.2.4.5.3	<i>Standalone Viewer.....</i>	<i>64</i>
3.3	System Integration – Architektur	64
3.3.1	Überblick	64
3.3.2	Datenhaltung.....	65
3.3.2.1	Geodaten in Datenbanken	65
3.3.2.2	Sachinformationen	66
3.3.3	Serverkomponenten	66
3.3.3.1	Mapserver	66
3.3.3.2	Anwendungsserver	66
3.3.3.3	Proxyserver	66
3.3.4	Der Client zur Darstellung.....	67
3.3.5	Datenübertragung	67
4	Der Prototyp	68
4.1	Anforderungen an die Anwendung	68
4.2	Funktionalität des Prototypen	68
4.2.1	Diagrammauswahl im Prototypen	69
4.2.2	Räumliche Navigation und Orientierung in der Karte	69

4.3	Realisierung des Prototypen.....	70
4.3.1	Äußeres Designmodul des Prototypen	72
4.3.1.1	Datenversorgung - Datenbereitstellung.....	72
4.3.1.1.1	Laden der Anwendungseinstellungen.....	72
4.3.1.1.2	Laden der Basisdaten.....	73
4.3.1.1.3	Bereitstellen der Sachdaten zur Diagrammgenerierung.....	73
4.3.1.2	MapSet	75
4.3.1.2.1	Der interaktive Kartenrahmen.....	75
4.3.1.2.2	Räumliche Navigationselemente zur Kartentranslation	75
4.3.1.2.3	Maßstabsbalken	76
4.3.1.3	Übersichtskarten zur räumlichen Orientierung	76
4.3.1.4	Containerelement für Legenden	76
4.3.1.5	Drop-Down Liste zur Diagrammauswahl	77
4.3.1.6	Interaktionsmöglichkeiten der kartographischen Benutzeroberfläche.....	77
4.3.1.7	Formatvorlagen zur individuellen Stilisierung.....	78
4.3.2	Inneres Designmodul des Prototypen	78
4.3.2.1	Kartenebene.....	78
4.3.2.2	Der Diagrammgenerator	78
4.3.2.2.1	ChartSetContainer.....	79
4.3.2.2.2	ChartSet.....	79
4.3.2.2.3	Diagrammeigenschaften	80
4.3.2.3	Bereitstellung der Basisdaten	80
4.3.3	Gesamtansichten des Prototypen	81
4.4	Bewertung des Prototypen	81
4.5	Verbesserungsvorschläge für den Prototypen.....	83
5	Zusammenfassung der Arbeit.....	85
	Literaturverzeichnis.....	87
	Anhang	

Abbildungsverzeichnis

Abb. 1: Prozessstufen des Geomarketings.....	12
Abb. 2: Schematische Darstellung eines WebGIS	18
Abb. 3: Die Nutzung von Webmaps.....	19
Abb. 4: Gliederung von Webmaps nach dem Map Use Cube	20
Abb. 5: Darstellung der verschiedenen Aussagformen thematischer Karten	21
Abb. 6: Räumliche Ausprägung – Implantation.....	21
Abb. 7: Maßskalen der Statistik und ihre Charakterisierung.....	22
Abb. 8: Visuelle Variablen in Abhängigkeit der Implantation.....	23
Abb. 9: Gliederungsstufen der visuellen Variablen.....	24
Abb. 10: Geeigneter Einsatz der visuellen Variablen im Bezug auf Skala und Implantation	24
Abb. 11: Übersicht wichtiger Kartengrundtypen des Geomarketings	25
Abb. 12: Ausschnitt einer Standortkarte.....	26
Abb. 13: Ausschnitt einer Verbreitungskarte	26
Abb. 14: Ausschnitt einer ortsbezogenen Diagrammkarte	27
Abb. 15: Ausschnitt einer streckenbezogenen Diagrammkarte	27
Abb. 16: Beispiel einer flächenbezogenen Diagrammkarte	28
Abb. 17: Beispiel einer Choroplethenkarte	28
Abb. 18: Ausschnitt einer statischen Darstellung von Bewegung	29
Abb. 19: Klassifizierungsmethoden bei Choroplethenkarten.....	34
Abb. 20: Graphische Probleme im Zuge einer Generalisierung	36
Abb. 21: Informationsschichten und visuelle Hierarchie thematischer Karten	38
Abb. 22: Größen elektronischer Bildanzeige an gebräuchlichen Monitoren	39
Abb. 23: Möglichkeiten der Schriftdarstellung an einem Bildschirm	40
Abb. 24: Linien ohne und mit Antialiasing	41
Abb. 25: Layoutvorschlag für einen kartographischen Viewers.....	46
Abb. 26: Recommendation Process.....	48
Abb. 27: Übersicht der XML Welt.....	48
Abb. 28: Vereinfachtes XML Konzept	49
Abb. 29: Beispiel eines XML Elementes.....	49
Abb. 30: Beispiel eines XML Dokumentes	50
Abb. 31: Funktionsweise des DOM.....	50
Abb. 32: Grundelemente der SVG Spezifikation.....	53
Abb. 33: SVG Elemente für zusätzliche graphische Effekte.....	53
Abb. 34: Beispiel eines Pfadelementes als Linie.....	54
Abb. 35: Beispiel eines Pfadelementes als Polygon.....	54
Abb. 36: Beispiel eines SVG Textelementes	55
Abb. 37: Vereinfachte Systemarchitektur	65
Abb. 38: Die implementierte kartographische Benutzeroberfläche.....	69
Abb. 39: Vereinfachter Funktionsablauf des Prototypen	70
Abb. 40: Mögliche SVG Datei mit einem thematischen Layer	71
Abb. 41: Das XMLDataProvider Objekt.....	72
Abb. 42: Auszug aus der XML Property Bag.....	73
Abb. 43: Vereinfachtes Beispiel eines spezifizierten XML Dokumentes	73
Abb. 44: Vereinfachtes Beispiel eines spezifizierten XML Dokumentes für Sachdaten	74

Abb. 45: Das Objektmodell für Sachdaten.....	74
Abb. 46: Darstellung der mit dem MapSet verbundenen Javascript Objekte.....	75
Abb. 47: Die Objekte der Übersichtskarten.....	76
Abb. 48: Das legend Objekt.....	77
Abb. 49: Das DropDownList Objekt	77
Abb. 50: Übersicht der kartographischen Interaktionsmöglichkeiten	78
Abb. 51: Die Objekte des Diagrammgenerators	79
Abb. 52: Diagrammtypen und der Eigenschaften	80
Abb. 53: Format der SVG Basisdatenfragmente.....	80
Abb. 54: Drill-Down Mapping mit dem Prototypen.....	81

1 Einleitung

1.1 Motivation und Zielsetzung

Der verstärkte Konkurrenzkampf zwischen Unternehmen hat in den letzten Jahren deutlich zugenommen. Grund dafür sind vor allem übersättigte Märkte und eine stagnierende Wirtschaftslage. Deswegen suchen Firmen nach neuen Methoden, effizient zu wirtschaften. Der Bereich der **Business Intelligence** – "Synonym" für intelligente Unternehmensführung - spielt dafür eine wichtige Rolle. Dessen Vorgehensweisen unterstützen Entscheidungsträger, neue Erkenntnisse aus Unternehmensdaten zu gewinnen. Business Intelligence Systeme dienen dafür, mit den sehr großen Datenbeständen eines Unternehmens umgehen zu können. Weiter bieten sie die Möglichkeit, komplexe Analysen auf einfache Art und Weise zu erstellen. Die Resultate sind entscheidungsunterstützende Informationen, die sich in der Unternehmenspolitik bemerkbar machen. Die geographische Komponente spielte bis jetzt eine geringfügige Rolle. Unberechtigterweise, wie der Erfolg des Geomarketings aufzeigt. Geomarketing beschreibt Vorgehensweisen, die Unternehmensdaten und Marktdaten in räumlichen Bezug stellen. Die Überlagerung dieser drei Informationsschichten führt zu sogenannten räumlichen Analysen. Geoinformationssysteme (**GIS**), die eigentlich in den traditionellen Geowissenschaften eingesetzt werden, eignen sich besonders dafür. Sie stellen die erforderliche Funktionalität zur Verfügung. Eine anwenderfreundliche Bedienung zur Bewältigung der komplexen Analysen ist ein weiterer Pluspunkt. Die Resultate sind vor allem thematische Karten. Diese öffnen neue Blickwinkel, die den zugrundeliegenden Informationsgehalt entscheidungsunterstützend darstellen. Die Wissenschaft der Thematischen Kartographie bietet dafür die notwendigen Grundlagen. Deren Methoden zur geeigneten Wahl der Darstellungstypen und zur Gestaltung der Kartenelemente fördern die Aussagekraft einer Darstellung.

Den Vorteil raumbezogener Analysen sowie der daraus resultierenden Darstellungen will sich auch Business Intelligence zu Nutze machen. Dazu fehlten aber in hohem Maße kartographische Visualisierungsmöglichkeiten. Speziell hier setzt die Zielsetzung dieser Arbeit an.

Das Ziel dieser Arbeit ist, Theorie zur Erstellung einer browserbasierten Anwendung für thematische Karten des Geomarketings aufzuarbeiten. Anhand eines Prototypen sollen die theoretischen Aspekte mit **XML**-basierten Technologien in der Praxis getestet werden.

Um Grundlagen für einen prototypischen Einsatz zu selektieren, stellen sich drei wichtige Fragen:

Welche Art der Information wird dargestellt?

Wie visualisiert man diese?

Eignen sich die dafür vorgesehenen Techniken?

Die Fragestellungen sollen in dieser Arbeit umfassend beantwortet werden. Gegenwärtige Webtechnologien finden im Hinblick auf den praktischen Teil besondere Berücksichtigung. Der realisierte prototypische Kartenviewer (→ **Viewer**) soll zeigen, dass die behandelten Technologien den Einsatz in Business Intelligence Systemen gerechtfertigen. Allgemein gebräuchliche Darstellungen des Geomarketings dienen als Kartenbeispiele, wobei nicht alleine die Bandbreite der entwickelten Darstellungen eine Rolle spielt. Auch die erforderliche Funktionalität der Anwendung sowie die Integration in gegebene Systemarchitekturen sollen verdeutlicht werden.

1.2 Aufbau der Arbeit

Um der Aufgabenstellung gerecht zu werden, gliedert sich die vorliegende Arbeit in fünf Kapitel. Wobei Kapitel eins die Einleitung und Kapitel fünf die Zusammenfassung bilden.

In Kapitel zwei werden theoretische Grundlagen und allgemeine Vorgehensweisen des Geomarketings erläutert. Danach folgt eine ausführliche Diskussion der Thematischen Kartographie. Diese beinhaltet moderne Verfahren zur Kartenherstellung, Grundlagen zur Methodik und Gestaltung, Kartentypen des Geomarketings sowie benötigte Kartenelemente. Der letzte Abschnitt dieses Kapitels konkretisiert die Eigenschaften von Bildschirmkarten und evaluiert kartographische Funktionalität für eine Vieweranwendung.

Im dritten Kapitel sind die erforderlichen Technologien erläutert. Dabei spielen die vom World Wide Web Konsortium (**W3C**) vorgeschlagenen Empfehlungen eine bedeutende Rolle. Hervorgehoben werden hier Scalable Vector Graphics (**SVG**), die **XML**-basierte Spezifikation zur Beschreibung von zweidimensionalen Vektorgraphiken. Am Ende dieses Kapitels soll eine vorgestellte Client-Server Architektur die Integrierbarkeit eines Viewers in ein komplexes System aufzeigen. Alle behandelten Aspekte werden unter Berücksichtigung kartographischer Anforderungen erläutert.

Das vierte Kapitel beschreibt den realisierten Prototypen. Zuerst wird neben den Anforderungen die konkrete Funktionalität erläutert. Danach folgt die Vorstellung der entwickelten Anwendungsmodule. Die Beschreibung von Funktionalität, Realisierung und Verbesserungsvorschläge runden das Kapitel ab.

In Kapitel fünf, dem Schlusskapitel wird die Zielsetzung unter Berücksichtigung der erforderlichen Teilaufgaben zusammenfassend bewertet. Ein Ausblick gibt weiterführende Überlegungen vor allem zu einer zukünftigen Realisierung des entworfenen Viewers. Abschließend werden eigene Einschätzungen zur Zukunft von SVG gemacht.

Abschließend folgt die Auflistung der eingesetzten Quellen in Form eines Literaturverzeichnisses.

Dem Anhang liegt eine Cd-Rom bei, die zahlreiche SVG Beispiele, den Prototypen sowie ein Glossar in digitaler Form enthält. Alle im Glossar beschriebenen Begriffe sind bei erstmaliger Erwähnung im Text **fett-kursiv** formatiert. Zudem wird das Literaturverzeichnis in digitaler Form angeboten. Das Glossar ist auch in gedruckter Form im Anhang zu finden.

2 Anforderungen an die Thematische Kartographie für das Geomarketing

2.1 Geomarketing

2.1.1 Definition

SCHÜSSLER definiert Geomarketing wie folgt:

"Geomarketing bezeichnet die Planung, Koordination und Kontrolle kundenorientierter Marktaktivitäten von Unternehmen" "Es werden Methoden angewendet, die den Raumbezug der untersuchten unternehmensinternen und -externen Daten herstellen, analysieren und visualisieren sowie sie zu entscheidungsrelevanten Informationen aufbereiten" (SCHÜSSLER F. 2000, S.9).

Es sei zu bemerken, dass der Begriff "Geomarketing" erst sehr jung und keineswegs allgemein gebräuchlich ist. Fachausdrücke wie "Micromarketing" (NITSCHKE M. 1998, S.9), Business GIS (engl.) oder "Business mapping im Marketing" (LIEBERICH P. 1997, S.5) werden in diesem Zusammenhang ebenfalls verwendet.

2.1.2 Entstehungsgeschichte

Warum Geomarketing so erfolgreich ist, lässt sich aus der Entstehungsgeschichte ableiten. Dabei spielen zwei Entwicklungen eine große Rolle.

Zum einen ist der verstärkte Zwang zur wirtschaftlichen Unternehmensführung zu nennen. Dieser ist vordergründig für die Vorgehensweisen im Geomarketing verantwortlich. Dabei spielt die Entwicklung unseres Wirtschaftsraumes von Verkäufer- zu Käufermärkten, resultierend aus der Sättigung der Märkte, eine bedeutende Rolle. Verstärkt wird diese Tatsache durch eine schlechte Wirtschaftslage, die einen intensiven Konkurrenzkampf zwischen Unternehmen mit sich bringt. Der daraus entstehende verschärfte Wettbewerb zwingt Firmen, Rationalisierungen oder tiefgreifendes Controlling vorzunehmen (SCHÜSSLER F. 2000, S.10/11). Derartige Zwänge betreffen auch Marketingaktivitäten, die als eine auf den Absatzmarkt ausgerichtete Unternehmenspolitik verstanden werden. Diese setzen sich aus "strategischen und taktischen Aktivitäten" basierend auf den ihnen zugrunde liegenden Informationen eines Unternehmens zusammen (vgl. LEIBERICH P. 1997, S.4). Eine Analyse dieser Informationen ergibt, dass der Anteil raumbezogener Informationen hoch ist. Zahlen von über 50% werden in der Literatur genannt (vgl. BILL R. 1999, S.289). Das hat zur Folge, die Komponente "Raum" mit in Marketingaktivitäten eines Unternehmens einfließen zu lassen. Diese werden durch die Geomarketing Branche abgedeckt. Wie effektiv die raumbezogenen Analysen sind, zeigt der Erfolg des Geomarketings. Zentrale Themen sind die optimale Ausschöpfung erschlossener und die Erschließung neuer Märkte (vgl. LATZ A. 1997, S.174). Ein Überblick über die Methoden des Geomarketing ist weiter unten zu finden (vgl. Kapitel 2.1.5).

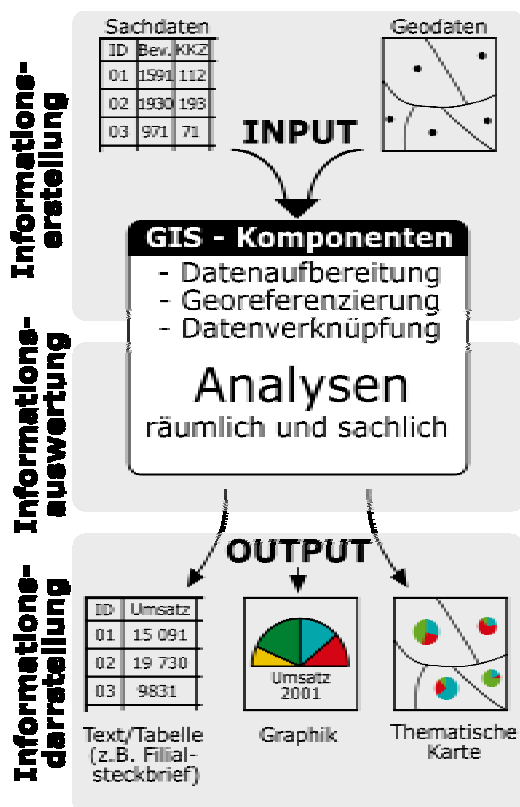
Als Zweites ist der informationstechnologische Fortschritt zu nennen, der die technische Voraussetzung für effektiven Geomarketingeneinsatz bietet. Dieser ist auf Veränderungen im Hardware- sowie im Softwarebereich zurückzuführen. Einerseits sind Computer heutzutage billig und leistungsstark. Selbst preisgünstige Systeme ermöglichen es, mit den großen Datenmengen und rechenaufwendigen Analysen des Geomarketings fertig zu werden. Andererseits muss die Entwicklung von Softwareanwendungen erwähnt werden. Besonders Geoinformationssysteme

eignen sich für Geomarketing. GIS besitzen die Fähigkeit, die großen Datenmengen zu verwalten und zu analysieren. Zudem spielen die guten kartographischen Ausgabemöglichkeiten von GIS eine wichtige Rolle. Auch die weitreichenden statistischen Möglichkeiten fördern den Einsatz von Geomarketing. Gegenwärtig stellen GIS-Hersteller umfangreiche Mittel für die Entwicklung verteilter GIS Anwendungen in einem Netzwerk bereit. Der Einsatz in webbasierten Systemen (vgl. Kapitel 2.2.2) wird sich deswegen erleichtern (vgl. SCHÜSSLER F. 2000, S.12-16). Die für räumliche Arbeiten notwendigen Datengrundlagen sind mittlerweile digital vorhanden.

2.1.3 Der Geomarketingprozess

Die allgemeine Vorgehensweise des Geomarketings basiert immer auf einem ähnlichem Schema. Wie Abbildung 1 zeigt, lassen sich drei Prozessstufen definieren. Informationserstellung, Informationsauswertung und Informationsdarstellung bilden die allgemeine Vorgehensweise des Prozesses.

Abb. 1: Prozessstufen des Geomarketings



Quelle: Eigene Darstellung basierend auf Abbildungen aus LEIBERICH P. 1997, S.80 und HELD S. 2002, S.18.

Ziel dieser Prozesskette ist es, "unstrukturiert oder latent vorhandene räumliche Informationen zu strukturieren, zu ergänzen und zu analysieren" (vgl. SCHÜSSLER F. 2000, S. 34). MUFF betont, dass dieser komplexe Vorgang nicht einfach zu erreichen ist (vgl. MUFF S. 2001, S.6).

Ausgangspunkt des Prozesses sind umfangreiche Sach- und Geodatenbestände, die über einen gemeinsamen Identifikationsschlüssel verknüpft werden. Im Geomarketing dienen Geodaten als Bezugssystem für Sachdaten. Sind die Informationen für das der Auswertung dienende System aufbereitet, können räumliche (z.B. **buffering**) sowie sachliche Analysen (z.B. Selektion nach Sachdaten durch **querying**) ausgeführt werden. Die Ergebnisse werden im letzten Schritt in geeigneter Form visualisiert. Kartenausgaben sind hier der Standard, da sie entscheidungsrelevante räumliche Information am besten darstellen können (vgl. MUFF S. 2001, S.6).

2.1.4 Daten für das Geomarketing

Wie im vorherigen Kapitel erwähnt, sind Daten die Basis und deswegen auch das "Heiligtum" im Geomarketingprozess. Dabei stehen besonders firmeninterne Kundendaten, externe Marktdaten und Geodaten im Vordergrund. Faktoren für die Beurteilung der Daten sind Genauigkeit, Aktualität, Vollständigkeit und Kleinräumigkeit beziehungsweise Datentiefe (vgl. NITSCHKE M. 1998, S.49). Im Folgenden werden Geodaten, die mindestens aus einem Koordinatenpaar bestehen, von Sachdaten, die das Thema in Form von Attributen beinhalten, unterschieden.

Hinweis: Größte Schwierigkeit ist es, einen Gesamtüberblick über das vorhandene Datenangebot zu bekommen, da es keine Kataloge mit allen Produkten oder eine einheitliche Lizenz- sowie Preisgestaltung gibt. Hier besteht großer Informationsbedarf, der durch im Geoconsulting beschäftigte Personen abgedeckt wird.

2.1.4.1 Geodaten

Unter Geodaten werden alle Daten verstanden, die mindestens aus einem Koordinatenpaar bestehen. Zur Handhabung der Geodaten dienen auf unterster Ebene Datenbanken oder Dateien. Darauf aufsetzend werden als **Frontend** häufig GIS eingesetzt. Diese bieten alle Möglichkeiten zur Bearbeitung der Informationen. Dazu zählen Eingabe, Verwaltung und Zuordnung der Daten. Wichtig im Umgang mit Geodaten ist die optimierte Datenmenge und der Preis. Geodaten sind kostspielige Informationen.

In der Praxis tauchen zudem Probleme beim "Datenhandling" auf. Selbst beste technische Voraussetzungen können dieses Hindernis nicht überwinden. Außerdem besitzen Bearbeiter häufig zu wenig Daten-Know-how. Datenformat, Bearbeitungssoftware und Kartenprojektion sind weitere problematische Themen.

Unterschieden wird in den nächsten Abschnitten bezüglich zweier konzeptioneller Basismodelle zur Beschreibung von Geodatengeometrien, Raster- und Vektordaten.

2.1.4.1.1 Rasterdaten

Grundelemente von Rasterdaten sind **Pixel**, die in Form einer Tabelle die Graphik beschreiben. Ausschlaggebend für das Aussehen sind deren Werte. Um mit Rasterdaten arbeiten zu können, müssen diese georeferenziert sein (→ **Georeferenzierung**). Sie dienen meist nicht der Analyse, sondern nur zur Datenerstellung beziehungsweise räumlichen Orientierung (z.B. als Hintergrundkarten). Typische Vertreter der Rasterdaten sind topographische Karten, Luftbilder oder Straßenkarten die beispielsweise "gekachelt" (→ **Kachel**) vorliegen.

2.1.4.1.2 Vektordaten

Geometrisches Grundelement ist der Punkt, definiert durch Koordinaten eines räumlichen Koordinatensystems. Punktmengen, auch Koordinatenfolgen genannt, können Linien oder Flächen (Polygone) bilden. Durch formatinterne Attributtabelle erhalten Vektordaten zusätzliche Informationen, wie beispielsweise Nachbarschaftsbeziehungen. Zudem können die räumlichen Objekte von Vektordaten mit Sachdaten verknüpft werden. Im Geomarketing dienen Vektordaten deswegen als Basis für die Analysen. Vektordaten sind beispielsweise administrative Grenzen oder Vertriebsgebiete.

2.1.4.2 Sachdaten als thematische Information

Sachdaten sind alle thematischen Informationen, die keine geometrische Ausprägung haben. Im Geomarketingprozess können Sachdaten nach ihrem Ursprung zwischen firmenexterner und -interner Herkunft unterschieden werden.

2.1.4.2.1 Externe Daten

Externe Daten stammen von Anbietern außerhalb eines Unternehmens. Dazu zählen demographische, soziale und wirtschaftliche Daten. Das Angebot externer Daten ist groß und wird von öffentlichen und privatwirtschaftlichen Datenanbietern abgerundet. Dabei ist außerdem wichtig, dass kommerzielle Datenanbieter ein breites Spektrum auch speziell aufbereiteter Datensätze handeln (vgl. SCHÜSSLER F. 1999). Hausnummerngenaue Adressdaten, Bevölkerungszahlen und Kaufkraftkennziffern sind Repräsentanten externer Sachdaten.

2.1.4.2.2 Interne Daten

Der Ursprung interner Daten dagegen liegt in firmeninternen Aktivitäten. Man unterscheidet zwischen reinen Geschäfts- sowie Kundendaten. Dazu ist es notwendig, Geschäftsdaten praktikabel zu speichern (Unternehmensinformationssystem), und Kundendaten in ausreichendem Maße zu beziehen (z.B. durch Kundenbefragung). Besonders die Verknüpfung mit externen Marktdaten ist für die Erstellung potentieller Kennzahlen sinnvoll. Typische Vertreter interner Sachdaten sind z.B. Kundenprofile und Umsatzzahlen für einen Standort.

2.1.5 Anwendungen und Methoden

Anwendungen und Methoden bilden die im Geomarketingprozess vorgestellte Stufe der Informationsauswertung. Dafür werden eine Vielzahl von verschiedenen Analysen und Vorgehensweisen verwendet. Meist gibt es keine allgemeinen Vorgehensweisen. Vielmehr zählt der Einsatz problemspezifischer Verfahren. Diese schließen vor allem die Zielvorgabe ("Was soll erreicht werden?") mit ihren passenden Ausgangsinformationen ein. Der Einsatz kann in verschiedenen Anwendungen an völlig unterschiedlichen betriebswirtschaftlichen Methoden ansetzen. Diese Komplexität hat zur Folge, dass ein Überblick aller möglichen Analysen extrem schwierig ist. Erschwerend kommt hinzu, dass in der Literatur keine einheitlichen Begriffsdefinitionen vorhanden sind. Nachfolgend werden die wichtigsten Anwendungsgebiete unterteilt nach allgemein üblichen und bewährten Methoden erläutert. Die Unternehmensform spielt in der folgenden Gliederung keine Rolle.

2.1.5.1 Standortplanung im Geomarketing

Standortplanung behandelt alle wichtigen Indikatoren, einen potentiellen Standort zu ermitteln oder einen vorhanden Standort dem Markt anzupassen. SCHÜSSLER unterscheidet dafür zwei Aufgabenkomplexe, die sinnvollerweise meist kombiniert eingesetzt werden (SCHÜSSLER F. 2000, S.102-127):

2.1.5.1.1 Standortsuche

Die Standortsuche hat die Aufgabe, einen geeigneten/potentiellen Standort zu ermitteln. Je nach Zielvorgabe, z.B. die Erhöhung des Marktanteils in einer bestimmten Region, kommen bestimmte Methoden/Analysen zum Einsatz (z.B. Marktselektion, Gebietsanalyse, Standortselektion). Aus den Ergebnissen wird entschieden, welche Gebiete/Standorte für die Ansiedlung in Frage kommen. Um aber letztendlich eine sinnvolle Entscheidung zu treffen, ist die Standortbewertung, in diesem Fall eine potentielle, von großer Bedeutung.

2.1.5.1.2 Standortbewertung

Bei einer Standortbewertung werden vorhandene oder denkbare Standorte analysiert und beurteilt. Daher ist Standortbewertung besonders für Controllingmaßnahmen eines Unternehmens wichtig. Aus der Standortbewertung folgen beispielsweise die Optimierung eines Filialnetzes, dass kann zur

Schließung bestehender Standorte oder zur Erweiterung des Filialnetzes führen. Im Einzelhandel sind weiter die Anpassung des Warensortimentes und der Preise wichtige Maßnahmen. Folgende Vorgehensweisen, die jeweils in "gegenwärtig" und "potentiell" unterteilt werden können (je nach dem, ob es den Standort schon gibt, oder dieser erst gefunden werden muss), stehen im Vordergrund:

- **Bestimmung des Einzugsgebietes eines Standorts:**
Das Einzugsgebiet wird auf verschiedene Kriterien untersucht, wie beispielsweise Luftliniendistanz und Fahrtzeit der Kunden zu einem Standort.
- **Zielgruppenanalyse für einen Standort:**
Dabei werden die Zielgruppen nach verschiedenen Faktoren analysiert, die Größe dieser festgelegt und somit das absolute Marktpotential für den Standort bestimmt.
- **Ermittlung der Rentabilität eines Standorts:**
Diese Analyse bewertet einen Standort objektbezogen und schließt zum Beispiel Ladenfläche oder Mietkosten mit ein.
- **Zusätzlich Analyse des Standortumfeldes:**
Sehr wichtig ist zudem die Umgebung eines Standortes zu untersuchen. Faktoren zur Struktur des Umfeldes wie die Wettbewerbssituation sind dabei zu nennen.

Die Ergebnisse der Standortbewertung schlagen sich beispielsweise auf die Preispolitik eines Unternehmens für verschiedene Standorte nieder (vgl. SCHÜSSLER F. 2000, S.129-150).

2.1.5.2 Database Marketing

Methoden des Database Marketing erstellen Scoring Modelle, die Informationen über Kunden speichern. Das ist beispielsweise das Kaufverhalten von Kunden. Allgemein finden Scoring Modelle beim zielgruppenspezifischen oder **event-based Marketing** Anwendung. Geomarketing optimiert die Scoring Modelle, indem diese mit Adressdaten und firmenexternen Daten verknüpft werden. Es entstehen sogenannte Marktzellen, die deren Bevölkerung charakterisieren. Gezielte Marketingmaßnahmen in ähnlichen Gebieten resultieren daraus (vgl. SCHÜSSLER F. 2000, S.163-171).

2.1.5.3 Mediaplanung und Mediaselektion

Mediaplanung beschreibt zielgerichtete Aktivitäten mit dafür geeigneten Informationsträgern. Dabei fließen Kundendaten zur generellen Auswahl dieser mit ein. Anhand einer durchgeführten Beurteilung vorhandener Medientypen (Themen, Zielgruppen, Verteilung) kann somit der geeignetste Informationsträger selektiert werden, um beispielsweise Werbung darin zu platzieren. Vorteile entstehen dadurch nicht nur für den Werber, sondern auch für den Absatz eines Mediums, denn durch eine eindeutig definierte Ausrichtung können Inhalte (quantitativ und qualitativ) angepasst werden. Zum Beispiel gewinnt eine Zeitung durch zielgruppenspezifische Ausrichtung an Attraktivität. Eingesetzte Medientypen sind Zeitungen, Magazine, Webseiten, Werbetafeln/-plakate.

2.1.5.4 Optimierung von Vertrieb und Service

In diesem Zusammenhang kann Geomarketing Lieferpolitik und Kundendienst eines Unternehmens entscheidend verbessern. Zwei häufig eingesetzte Methoden werden in den nächsten Kapiteln behandelt.

2.1.5.4.1 Bildung von Marktverantwortungsregionen

Die Erstellung und Optimierung sogenannter Marktverantwortungsregionen werden einerseits durch vorhandene Kunden, andererseits durch potentielle Kunden bestimmt. Zudem wird auf die gegenwärtigen Marktverantwortungsregionen eines Standortes zurückgegriffen und diese berücksichtigt. Unter-, beziehungsweise übertersorgte Gebiete gilt es zu vermeiden. Danach folgt die Feinabstimmung der Gebiete nach verschiedenen Indikatoren und auf räumlicher Bezugsbasis, wie Postleitzahlengebieten (vgl. SCHÜSSLER F. 2000. S.175-188).

2.1.5.4.2 Tourenplanung

Weiter spielt effektive Tourenplanung eine bedeutende Rolle im Bereich Vertrieb und Service. Es wird zwischen strategischer und operativer Tourenplanung unterschieden:

- **strategisch:**
Die Gesamtheit der Kundentermine werden so auf die Mitarbeiter verteilt, dass effiziente Termingestaltung gegeben ist.
- **operativ:**
Berechnung der kürzesten sowie schnellsten Route für ein bestimmtes Fahrzeug. Faktoren wie die Länge der Strecke, die durchschnittliche Geschwindigkeit, der Straßentyp oder das Gelände (Höhenmodell) spielen eine wichtige Rollen.

Tourenplanung findet vor allem in der Verkehrslogistik Anwendung (vgl. SCHÜSSLER F. 2000, S.188-193).

2.1.6 Anforderungen an Geomarketingsysteme

Unter Geomarketingsystemen versteht man Informationssysteme, die für Geomarketingzwecke geeignet sind. Sie müssen immer den in Abbildung 1 vorgestellten Prozess erreichen. Wichtigste Anforderung ist das Zusammenwirken folgender Systemkomponenten:

- **Datenaufbereitung:**
Zur Datenaufbereitung zählt die Erfassung, Geokodierung und Weiterverarbeitung zu verwendbaren Informationen. Editiermöglichkeiten zur Datenpflege ist deswegen entscheidender Faktor. Um Daten in ein System einzuspielen, bedarf es einer Importfunktion.
- **Datenschnittstelle:**
Eine Datenschnittstelle muss vorhanden sein, um Daten in Analysen und Darstellungen integrieren zu können. Dazu zählen umfangreiche Selektiermöglichkeiten sowie die Datenanbindung an Datenbanken. Sowohl Geo- als auch Sachdaten müssen integrierbar sein.
- **Analysefunktionen:**
Die im Geomarketing notwendigen Analysen sind zum großen Teil durch GIS Funktionalitäten abzudecken. Diese werden in der Fachwelt als "*spatial analysis*" bezeichnet. Möglichkeiten zur individuellen Automatisierung von Analysen müssen geboten werden.
- **Darstellungs-/Ausgabekomponente:**
Diese muss verschiedene Möglichkeiten zur Ausgabe von Informationen vorgeben. Wichtigste sind Text, Tabellen, Graphiken und Karten. Dabei ist auch die Informationsweitergabe durch verschiedene Exportmöglichkeiten unabdingbar.
Für eine webbasierte kartographische Darstellungskomponente eines Geomarketing Systems müssen weitere Anforderungen gestellt werden. Eine entsprechende Anwendung muss eine spezifizierte Datenversorgung erhalten. Über eine anwenderfreundliche Benutzeroberfläche soll der Anwender Interaktion betreiben können. Jedoch beschränkt sich die Funktionalität der Anwendung auf die Gestaltung und Darstellung thematischer Karten. Weiter sind Möglichkeiten zur geeigneten Visualisierung von Vektor- und

Rasterdaten notwendig. Wichtige Kartentypen und Darstellungselemente müssen implementiert sein. Der Karteninhalt ist individuell selektierbar und die Darstellung passt sich automatisch an. Der Funktionsumfang eines Viewers dient nicht zur Durchführung komplexer Analysen.

- **Sonstiges:**

Neben den Basisfunktionalitäten sind an ein ausgereiftes Geomarketing System noch zusätzliche Anforderungen zu stellen. Die Entwicklungen der letzten Jahre förderten in zunehmendem Maße Webtechnologien sowie anwenderfreundliche Systeme.

In der Praxis enthalten Geomarketingsysteme Komponenten aus den Bereichen Geo- und Unternehmensinformationssystemen und statistischen Programmen (z.B. Tabellenkalkulation).

2.2 Thematische Kartographie

2.2.1 Definition und Begriff

Thematische Kartographie "beschäftigt sich in Theorie und Praxis vor allem mit Entwurf, Gestaltung und Herausgabe thematischer Karten sowie mit deren Informationsgehalt und Nutzungsmöglichkeiten" (WILHELMY H. 1996, S.195). Sie stellt raumbezogene Themen nicht - topographischer Art dar, die als thematische Karten bezeichnet werden. Die Abgrenzung gegenüber topographischen Karten ist nicht genau gesteckt. Thematische Karten beinhalten auch topographische Gegebenheiten. WITT betont, dass im Gegensatz zu topographischen Karten, bei denen die Orientierung im Vordergrund steht, in thematischen Karten das Thema die entscheidende Rolle spielt (vgl. WITT W. 1967, S.17/18).

Der Begriff "thematische Karte", erstmals 1934 von R. von Schumacher verwendet, hat sich nach dem Zweiten Weltkrieg durchgesetzt. Ältere Bezeichnungen sind angewandte Karte oder Spezialkarte (vgl. ARNBERGER E. 1997, S.13).

HINWEIS: Erschwerend für die Wissenschaft der Thematischen Kartographie ist, dass Fachbegriffe nicht standardisiert sind. Aus diesem Grund finden in dieser Arbeit nur ausgewählte Begriffe - vom Autor der Quelle abhängig - Erwähnung. Hinzu kommt, dass zwar viele Lehrbücher für die Thematische Kartographie zu finden sind, doch wenige dem zeitgemäßen Anspruch gerecht werden. So sind die folgenden Kapitel als Zusammenführung der wichtigsten Grundlagen mit vielen eigenen Thesen bezüglich dieser Arbeit zu sehen.

2.2.2 Gegenwärtige Kartenherstellung thematischer Karten

Der von KRAAK vorgestellte kartographische Kommunikationsprozess schildert die allgemeine Vorgehensweise zur Herstellung einer Karte. Zuerst wird die Information analysiert, danach die Darstellungstypen und die graphischen Elemente festgelegt. Anhand der visualisierten Sachverhalte soll der Kartennutzer die Information effektiv nutzen können (vgl. KRAAK M.-J. 2001b, S.57). Moderne Kartenherstellung bringt neue Arbeitsmethoden mit, diese Vorgehensweise zu erreichen. Damit verbunden ändern sich auch die Anwendungsgebiete. Allgemein kann behauptet werden, dass die Kartenherstellung eng mit den technischen Veränderungen in der Druck- und Medienbranche zusammenhängt (vgl. PETERSON M. 1995, S.76).

Geomarketing ist eines der Beispiele, das ohne moderne Kartenherstellungstechniken keinen Erfolg hätte. Die Darstellung von Echtzeitdaten erfordert einfache sowie automatisierbare Schritte der Kartenherstellung. Da Karten des Geomarketings nicht der uneingeschränkten Mobilität des Nutzers ausgesetzt sind, werden diese hauptsächlich für digitale Medien hergestellt. Die Eigenschaften von Karten verschiedener Medien zeigt die Übersicht in REICHENBERGER & ANGSÜSSER & MENG. Dabei wird vor allem zwischen inhaltlichen Attributen und

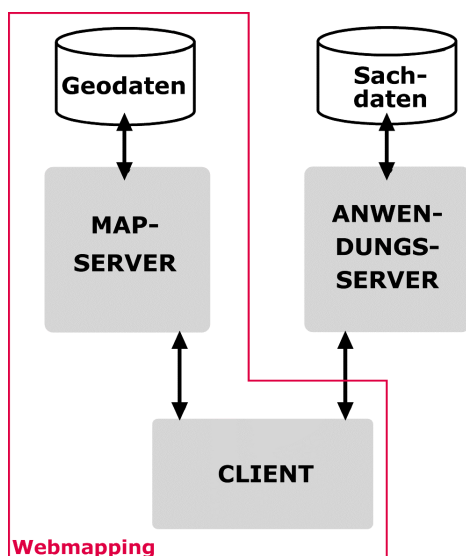
Nutzungseigenschaften unterschieden. Sieht man von den digitalen mobilen Karten ab, werden den Ansprüchen des Geomarketings nur Bildschirmkarten eines Onlinesystems gerecht (vgl. REICHENBERGER T. & ANGSSÜSSER S. & MENG L. 2002, S.164). Kartographische Onlinesysteme können in **Webmapping**-Systeme oder **WebGIS** unterteilt werden. Beide sind im Folgenden erläutert.

2.2.2.1 Webbasierte Kartenherstellung

Unter Webmapping versteht man die kartographische Visualisierung von Geodaten über das Internet. Ein Client-Server-System bietet auf einem **Mapserver** Applikationen, die aus einer **Geodatenbank** die nötigen räumlichen Informationen abrufen und weiterverarbeiten. Die entstandenen Karten heißen Webmaps. Der **Server** überträgt die Daten zum **Client**, der diese visualisiert. Man spricht in diesem Zusammenhang auch von "**rendern**". Der Client besteht aus einem Viewer, der räumliche Navigationsfunktionen sowie eine Layerkontrolle zur Verfügung stellt. In Abgrenzung zum WebGIS, hat Webmapping keine oder nur teilweise Analysefunktion. DICKMANN sieht hier einen Konflikt, da diese Aussage keine eindeutige Trennung von Webmapping und WebGIS ermöglicht. Typisches Beispiel für Webmapping ist ein interaktiver Stadtplan im Internet (DICKMANN F. 2001, S.17-21).

Ein vorhandenes Webmapping-System kann durch die Anbindung einer Sachdatenbank mit zugehörigem Sachdatenserver und der Bereitstellung eines **Anwendungsservers** mit GIS Funktionalität (auch **Geoservice** genannt) zu einem WebGIS erweitert werden. Die detaillierten technischen Variationen eines WebGIS seien hier mal dahin gestellt. Letztendlich werden diese nach Kriterien, wie viel Funktionalität von welcher Systemkomponente zur Verfügung gestellt wird, unterschieden. Eine Beschreibung verschiedener WebGIS Typen bietet MAC GILLAVRY (vgl. MAC GILLAVRY E. 2000, S.21-24). Dabei ist zu beachten, dass ein WebGIS meist aus mehreren selbständigen Anwendungsservern, auch **Komponentenserver** genannt, besteht. Wie in Abbildung 2 zu sehen ist, beinhaltet ein WebGIS die zur Visualisierung von Geodaten notwendigen Webmappingkomponenten.

Abb. 2: Schematische Darstellung eines WebGIS



Quelle: Eigene Darstellung.

Der Einsatz eines WebGIS für thematische Karten ersetzt den traditionellen Kartenherstellungsprozess komplett. Der Kartograph findet seinen Tätigkeitsbereich vor allem in der Programmierung von, für die kartographische Visualisierung wichtigen Softwarekomponenten.

Weiter muss er alle Technologien, von der Datenbeschaffenheit bis zum Client, im Überblick behalten. CARTWRIGHT & PETERSON fügen hinzu, dass der Kartennutzer derartiger Systeme nicht mehr der subjektiven Meinung des Kartographen ausgeliefert ist. Vielmehr können die Darstellungen individuell angepasst werden (vgl. CARTWRIGHT W. & PETERSON M. 1999, S.2/3).

Probleme eines WebGIS ergeben sich gegenwärtig technologischer Art. Eingesetzte System- und Programmieretechnik, Datentransformation sowie die Übertragungsbandbreite von Netzwerken bilden häufige Problemstellungen. Meist resultiert aus diesen eine niedrige Systemperformanz.

Ein wesentlicher Aspekt bezüglich der technischen Realisierung und Funktionalität ist die Wahl des Datenmodells. Unterschieden werden Raster- und Vektordatenmodelle, die beide für Webmappingzwecke geeignet sind. Eine ausführliche Diskussion beider Typen würde zu weit gehen, jedoch sind wichtige Unterschiede im Bezug auf einen kartographischen Viewer ansprechen.

Zum einen erfordern thematische Karten des Geomarketings vor allem zur Analyse, aber auch zur Darstellung, Vektordaten (vgl. Kapitel 2.1.4.1.2). Zum anderen ist die Skalierbarkeit von Vektordaten anzusprechen, die beispielsweise stufenloses Zoomen ohne zusätzliche Serverkommunikation ermöglicht. Geeignete Vektordatenformate für Webanwendungen sind noch im Frühstadium ihrer Entwicklung und werden deswegen auch nicht in gängigen Webbrowsern implementiert. Es sind eigene Viewer oder Plug-In's erforderlich.

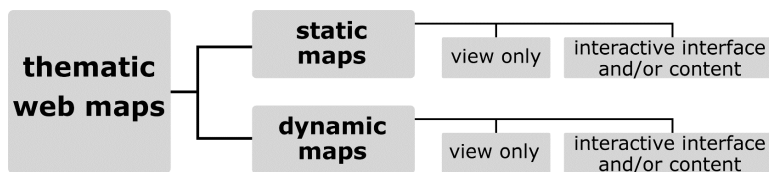
Hinweis: Einen umfangreichen Überblick gegenwärtiger Vektorformate geben NEUMANN & WINTER (vgl. NEUMANN A. & WINTER A. 2001, Kapitel 5).

2.2.2.2 moderne Karten - Webmaps

Die aus Webmapping und WebGIS entstandenen Karten werden als Webmaps bezeichnet. Um einen Überblick über diese Art der Darstellung zu erhalten, dienen verschiedene Gliederungen.

Zum einen ist die Klassifikation nach deren Nutzung zu nennen. Man unterscheidet zwischen statischen und dynamischen Darstellungen (Abbildung 3). Statisch bedeutet, es handelt sich um Karten mit bewegungslosem Inhalt. Dynamische Webmaps sind dagegen animierte Darstellungen. Beide können mit oder ohne Funktionalität ausgestattet sein.

Abb. 3: Die Nutzung von Webmaps

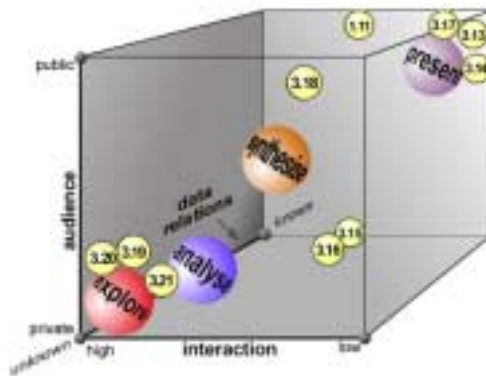


Quelle: Eigene Darstellung nach KRAAK M.-J. 2001a, S.3.

Für die vorliegende Arbeit sind hauptsächlich interaktive statische Karten von Bedeutung.

Zum anderen ist der Map Use Cube zu erwähnen (Abbildung 4). Er macht eine Webmap abhängig von den drei Faktoren Interaktion, Information und Anwendungsbereich. Diese werden als Achsen eines Würfels definiert. Werden Webmaps nach diesem Würfel klassifiziert, erhält man dadurch Aufgabe und Aussage dieser (vgl. MACEACHREN A.M. 1994, S.5-8 + S.11/12).

Abb. 4: Gliederung von Webmaps nach dem Map Use Cube



Quelle: Darstellung aus ELZAKKER C.P.J.M. 2001, S.25.

Der Würfel zeigt, dass thematische Karten, die vorwiegend zur Analyse dienen, hohe Interaktion erfordern. Der Benutzer muss deswegen die Karte genau studieren. Die entscheidenden Daten sind hauptsächlich fachlicher Art. Der Nutzerkreis besteht zum überwiegenden Teil aus Fachleuten.

Erkenntnisse aus der Einordnung von Karten in das Würfelmodell erlauben deswegen eine Aussage über die Darstellungsmöglichkeiten des Clients eines Webmapping Systems. Ein Viewer kann aufgrund seiner Datengrundlagen, Funktionalität und des Benutzerkreises beurteilt werden.

2.2.3 Die Grundlagen thematischer Karten

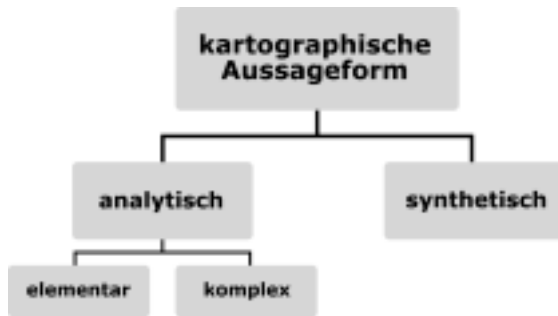
WITT betont, dass die "Überzeugungskraft der Karte, damit aber auch verbunden die Verantwortlichkeit" entscheidend von der Wahl der kartographischen Ausdrucksmittel abhängt. Diese werden durch die Grundlagen thematischer Karten erläutert. Dazu zählen vor allem die Aufgabe, die Methodik und die Darstellungstypen einer thematischen Karte, die im Folgenden umfassend beschrieben werden (vgl. WITT W. 1967, S.47).

2.2.3.1 Aufgabe und Aussage der Darstellungen

Aufgabe einer thematischen Karte ist es, ein oder mehrere Themen anhand seiner oder ihrer Erscheinung im Raum kartographisch sinnvoll darzustellen. Themen sind geordnete, quantitative und qualitative Gegebenheiten. Diese können aus Messungen, Statistiken und Analysen resultieren. Themen sind außerdem sinnvoll zu wählen (vgl. WILHELMY H. 1996, S.196).

Die Aussage thematischer Karten beinhaltet in erster Linie Angaben zu räumlichen Gegebenheiten. Dabei spielen die Lage, Verbreitung, Bewegung von Gegebenheiten eine wichtige Rolle. Qualitäten, Quantitäten und Ordnungen sind die wichtigsten Eigenschaften dieser (vgl. Kapitel 2.2.3.2). Zudem geben thematische Karten die Möglichkeit, zukünftige Thematiken wie Tendenzen und Fiktionen zu behandeln (vgl. WILHELMY H. 1996, S.196/197). Die verschiedenen Aussageformen thematischer Karten zeigt Abbildung 5.

Abb. 5: Darstellung der verschiedenen Aussageformen thematischer Karten



Quelle: Eigene Darstellung basierend auf der Ausführung in ARNBERGER E. 1997, S.17-20.

Eine analytische Aussage ist gegeben, wenn ein Sachverhalt ungegliedert oder gegliedert dargestellt wird (WILHELMY H. 1996, S.202). Wie Abbildung 5 zeigt, unterscheidet ARNBERGER diese in elementaranalytisch und komplexanalytisch. Elementaranalytische Aussageform stellt einen Sachverhalt dar. Komplexanalytische behandelt mehrere Inhalte. Anwendung finden diese Karten in allen Bereichen, besonders in den Wissenschaften und deren statistischen Methoden. Thematische Karten mit synthetischer Aussage enthalten mehrere Gegebenheiten, wobei diese zusammen eine völlig neue Aussage ergeben. Die einzelnen Erscheinungen sind nicht mehr direkt interpretierbar. Methodisch ist diese Art der Aussage am schwierigsten zu erreichen. Es bedarf an komplexen Auswertungen aus multidimensionalen Datenquellen. (vgl. ARNBERGER E. 1997, S.17-20).

2.2.3.2 Eigenschaften der Informationen

Die Informationen sind wichtigster Bestandteil thematischer Karten. Sie stellen die Hauptaussage einer Karte dar und bilden die Basis für die Darstellungsmöglichkeiten. Deswegen ist es besonders wichtig, diese auf ihre Eigenschaften zu untersuchen. Im Folgenden werden die räumliche Ausprägung und die statistischen Eigenschaften der Sachdaten untersucht.

Die geographische Ausprägung ist Voraussetzung für eine kartographische Umsetzung. Sie bildet die Basis jeder Darstellung und beeinflusst das Aussehen einer Karte erheblich. Die geographische Ausprägung wird durch die Implantation einer Gegebenheit beschrieben. Abbildung 6 zeigt die Implantation und deren Unterscheidung in punkt-, linien- und flächenhaft.

Abb. 6: Räumliche Ausprägung – Implantation

Ausprägung	Beschreibung	Beispiele
punkthaft (positionstreu)	Ein Sachverhalt ist einer Ortslage im Raum zuzuordnen. Dieser kann dabei auch situationstreu dargestellt werden. Das bedeutet, die Situation (topographischen Objekte) einer Gegebenheit fließt in die Ausprägung des kartographischen Objektes ein.	Standortzeichen
linienhaft (streckentreu)	Ein Sachverhalt ist einer Linie zuzuordnen. Die topographische Ausprägung der Strecke muss dabei generalisiert werden.	Banddiagramme
flächenhaft (raumtreu)	Ein Sachverhalt ist einer Fläche zuzuordnen. Die Fläche wird in ihrer Ausprägung nicht verändert.	Choroplethenkarte

Quelle: Eigene Darstellung nach BERTIN J. 1972, S.52/53 und WILHELMY H. 1996, S.213.

Neben der geographischen Ausprägung werden verschiedene Arten und Eigenschaften der thematischen Information betrachtet. Die vorliegenden statistischen Elemente, die in einer Karte dargestellt werden sollen, heißen Merkmale oder Merkmalsträger. Ein Merkmal kann in Ausprägungen (Werte) und Variablen (Merkmalsdimensionen) unterteilt werden. Unter einer Variablen versteht man eine Funktion, die jedem Merkmal genau einen Wert zuordnet. Die Wertemenge wird als Daten bezeichnet, die in Skalen ausgedrückt werden. Im Folgenden ein

Überblick über die möglichen Skalen, die ausschlaggebend für die Darstellung sind (vgl. BAHRENBURG G. & GIESE E. & NIPPER J. 1999, S.13-15).

Abb. 7: Maßskalen der Statistik und ihre Charakterisierung

Skala	Relationen und Operationen	Zweck	Variable	Beispiel
Nominal	$x_i = x_j; x_i \neq x_j$	Identifikation von Untersuchungselementen	qualitativ	Geschlecht
Ordinal	$x_i = x_j$ $x_i < x_j; x_i > x_j$	Identifikation von Untersuchungselementen + Ordnung	qualitativ geordnet	Zentralität eines Ortes
Intervall	$x_i = x_j$ $x_i < x_j; x_i > x_j$ $x_i = x_j + a; x_i = x_j - b; x_i + x_j = c$	Identifikation und Ordnung von Untersuchungselementen + Bewertung	quantitativ	Temperatur
Ratio	$x_i = x_j$ $x_i < x_j; x_i > x_j$ $x_i = x_j + a; x_i = x_j - b; x_i + x_j = c$ $x_i = x_j \cdot a; x_i = x_j / b; x_i \cdot x_j = c$	Identifikation, Ordnung und Bewertung von Untersuchungselementen + Ratio	quantitativ	Einkommen, Entfernungen

Quelle: Eigene Darstellung basierend auf der Ausführung in DENT B. 1996, S.79-81 und BAHRENBURG G. & GIESE E. & NIPPER J. 1999, S.15-17.

Eine nominalskalierte Merkmalsausprägung drückt lediglich Gleichheit oder Verschiedenartigkeit aus. Eine Ordinalskala bringt dagegen eine natürliche Rangordnung mit sich. In beiden Fällen spricht man von qualitativen Variablen. Intervallskalierte Daten gehen einen Schritt weiter und lassen einen Vergleich der Abstände zwischen den Ausprägungen zu. Die ratioskalierte Wertemengen unterscheidet die Werte durch einen einheitlichen Faktor. Dieser ermöglicht die Bildung eines "sinnvollen" Quotienten. Weiter ist der Wert 0 der absolute Nullpunkt. Intervall- und ratioskalierte Daten gehören zur Gruppe der metrischen Skalen, die ausschließlich Quantitäten darstellen. Diese können für den kartographischen Darstellungstypen folgende wichtige Eigenschaften besitzen:

- **absolut:** Zahlen ohne jeglichen Bezug (Achtung: auch Prozentwerte können absolut sein!)
- **relativ:** Bezugswerte wie Dichtezahlen und Quoten
- **diskret:** besitzen endlich viele Merkmalsausprägungen (z.B. Einwohnerdichte)
- **stetig:** besitzen unendlich viele Merkmalsausprägungen, auch wenn nur in einem bestimmten Intervall (z.B. das Relief der Erde). Diese werden auch Kontinua genannt und können durch Isolinien dargestellt werden.

Allgemein sei noch erwähnt, dass Daten eines bestimmten Skalenniveaus auch in ein niedrigeres Niveau transformiert werden können - umgekehrt ist das nicht möglich. Eine derartige Transformation bringt eine Abnahme des Informationsgehaltes mit sich.

Die Werteeigenschaften und die Lage im Raum grenzt die Darstellungsmöglichkeiten einer Erscheinung stark ein (vgl. Abbildung 10).

Hinweis: BERTIN erwähnt eine weitere Variableneigenschaft, die Länge. Diese drückt sich in der Anzahl der Werte aus. Wichtig ist außerdem, dass der hier verwendete Begriff "Variable" bei BERTIN als Komponente bezeichnet wird (vgl. BERTIN J. 1972, S.41).

2.2.3.3 Die Darstellungsmöglichkeiten thematischer Inhalte

Die Darstellungsmöglichkeiten thematischer Inhalte diskutieren deren graphische Ausprägung. Basierend auf den Werteeigenschaften und der Implantation werden die kartographischen Möglichkeiten einer Darstellung festgelegt.

In diesem Zusammenhang spielt das graphische System BERTIN's die entscheidende Rolle. Er untersucht die Veränderbarkeit eines Flecks. Er nennt die verschiedenen Ausprägungen visuelle Variablen. Folgende Variablen werde unterschieden:

Die beiden Dimensionen einer Fläche, Größe, Helligkeitswert, Muster, Farbe, Richtung und Form (BERTIN J. 1972, S.50). Dabei hat jede Variable besondere Eigenschaften und Aussagekraft. Einen Überblick der Variablen in Abhängigkeit der Implantation gibt Abbildung 8.

Abb. 8: Visuelle Variablen in Abhängigkeit der Implantation

Variable	Flächenhaft	Linienhaft	Punkthaft
Größe			
Helligkeitswert			
Form			
Farbe			
Richtung			
Muster/Textur			

Quelle: Eigene Darstellung basierend auf der Abbildung in BERTIN J. 1974, S.74/76.

Eine Variable ist dabei nicht für jede Art der Implantation geeignet. Besonders ungeeignete sind die flächenhafte Größenvariation und die linienhafte Variation der Form. Andere Variationen, wie eine linienhafte Richtungsvariation sind zwar möglich, werden aber nur selten eingesetzt.

Bei der Auswahl der richtigen visuellen Variable für die Darstellung der Merkmalsausprägungen ist es von großer Bedeutung, wie visuelle Variablen wahrgenommen werden. BERTIN unterscheidet zwischen assoziativer, selektiver, geordneter und quantitativer Wahrnehmung. Abbildung 9 gibt eine Übersicht darüber:

Abb. 9: Gliederungsstufen der visuellen Variablen

Variable	Gliederungsstufen der visuellen Variable			
Dimensionen der Ebene	assoziativ	selektiv	geordnet	quantitativ
Größe	dissoziativ	selektiv	geordnet	quantitativ
Helligkeitswert	dissoziativ	selektiv	geordnet	nur elementar erfassbar
Muster	assoziativ	selektiv	geordnet	
Farbe	assoziativ	selektiv		
Richtung	assoziativ	(selektiv)		
Form	assoziativ			

Erläuterung der Begriffe (vgl. BERTIN J. 1974, S.104) :

- assoziativ: alle Zeichen werden als gleichartig betrachtet
- selektiv: alle Zeichen werden als verschieden betrachtet; sie bilden Familien
- geordnet: Alle Zeichen werden als geordnet betrachtet
- quantitativ: Alle Zeichen werden als untereinander proportional betrachtet
- elementare Stufe des Erfassens: es wird nur eine einzige Beziehung erfasst (BERTIN J. 1974, S.18/149).

Quelle: Eigene Darstellung basierend auf der Ausführung in BERTIN J. 1974, S.77/104.

Die Wahrnehmung einer Variable wird zur endgültigen Festlegung in enge Verbindung mit einem Skalenniveau und der Implantation gebracht (Abbildung 10). Zum Beispiel werden ordinalskalierte Informationen in linienhafter Ausprägung geordnet wahrgenommen. In der Praxis stellen sich bestimmte Variablen als besonders geeignet heraus. Der Kreis der Darstellungsmöglichkeiten wird dadurch noch mal verkleinert.

Abb. 10: Geeigneter Einsatz der visuellen Variablen im Bezug auf Skala und Implantation

Implantation	Skalenniveau		
	nominal	ordinal	metrisch
punkthaft	Form, Farbe, Muster	Größe, Helligkeitswert	Größe, Helligkeitswert
linienhaft	Form, Farbe, Muster	Größe, Helligkeitswert, Muster	Größe, Helligkeitswert
flächenhaft	Farbe, Muster	Helligkeitswert, Muster	Helligkeitswert

Quelle: Eigene Darstellung nach BERTIN 1974 und OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.75.

Zusätzlich kann die Aussage der Darstellung durch Variablenkombinationen verstärkt werden. Meist dient sie auch zur differenzierten Darstellung von Sachverhalten.

Beispiele für Variablenkombinationen sind:

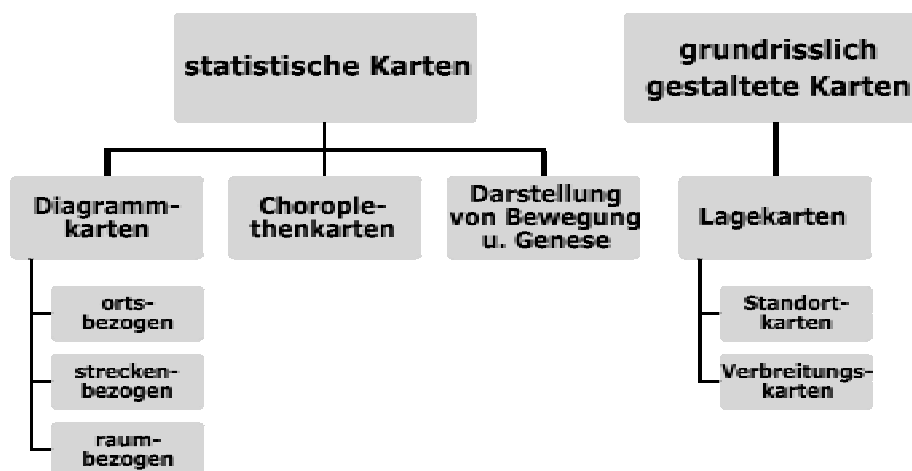
- **Helligkeit – Farbe:** Diese Kombination verlängert eine einpolige Farbskala, indem eine zweite Farbe mit einbezogen wird. Dabei ist wichtig, dass die vorhandene Helligkeitsabstufung trotzdem eingehalten wird. Dadurch erhöht sich die Länge der Farbskala. Es können bei der Klassifizierung der Werte zusätzliche Gruppen gebildet werden.

- **Helligkeit – Muster:** Diese Kombination stellt Mustervarianten in unterschiedlichen Helligkeitswerten dar. Sie wird häufig bei Schraffuren eingesetzt.
- **Größe – Farbe:** Diese Kombination ermöglicht es, durch Farbvariation Qualitäten und durch Größenvariation Quantitäten darzustellen. Sie ist beispielsweise bei Balken- oder Kreissektorendiagrammen zu sehen.

2.2.4 Kartentypen des Geomarketings

Kartengrundtypen des Geomarketings bilden die Basis für Geomarketinguntersuchungen. Aus ihnen können wichtige Erkenntnisse gewonnen werden. Die zugrundeliegenden Daten lassen weitere räumliche Analysen zu. Abbildung 11 zeigt die wichtigsten Grundtypen thematischer Karten für Geomarketing. Diese werden allgemein in statistische und grundrisslich gestaltete Karten unterteilt.

Abb. 11: Übersicht wichtiger Kartengrundtypen des Geomarketings



Quelle: Eigene Darstellung basierend auf der Abbildung in IMHOF E. 1970, S.102/103 und der Ausführung von FRÜHLING J. & STEINGRUBE W. 1997, S.123-131.

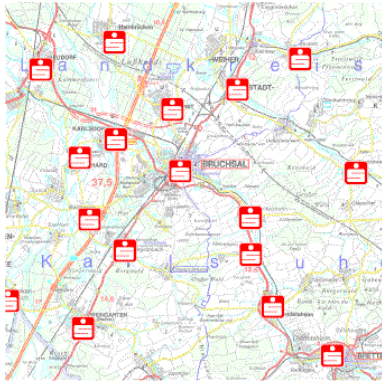
Statistische Karten drücken mindestens eine Erscheinung primär durch Mengenwerte (Quantität) aus. Grundrisslich gestaltete Karten vermitteln Vorkommnisse in ihrer Lage. Die Praxis zeigt, dass statistische Karten die Mehrheit bilden.

Hinweis: Die aus komplizierten Analysen resultierenden kartographischen Darstellungen, wurden nicht berücksichtigt. Diese zählen nicht zu Kartengrundtypen sondern sind spezielle "Abkömmlinge".

2.2.4.1 Lagekarten

2.2.4.1.1 Standortkarten

In Standortkarten werden Standorte jeglicher Art (typisch für Geomarketing sind beispielsweise Filialen) anhand von Positionssignaturen im Mittelpunkt der Objektverbreitung dargestellt. Lehrbücher der Thematischen Kartographie bezeichnen diesen Darstellungstypen auch als Positions- bzw. Ortslagekarte (WILHELMY H. 1996, S.217, und IMHOF E. 1972, S.107). Punktartige Kleinsignaturen vermitteln die verschiedenen Standorte am Besten. Für das schnelle Verständnis ist es sinnvoll, auf Firmenlogos zurückzugreifen. Jedoch müssen diese geeignet sein. Piktogrammähnliche und oder auch allgemein bekannte Logos sind besonders dienlich. Abbildung 12 zeigt ein Beispiel mit Firmensymbolen als Hinweis zur Lage des Standortes.

Abb. 12: Ausschnitt einer Standortkarte

Quelle: Eigene Darstellung.

Hinweis: Die Lage der Symbole stimmt nicht mit der realen Lage der entsprechenden Filialen überein!

2.2.4.1.2 Verbreitungskarten

Die Verbreitung einer Erscheinung wird in einem geeignetem Maßstab mittels flächenhafter Signaturen dargestellt. In Verbreitungskarten werden nur Qualitäten dargestellt. Moderne Kartographie bedient sich am besten der visuellen Variablen Farbe oder Muster. Bei Gebietsüberschneidungen können unterschiedliche Grenzliniendarstellungen, Muster als flächenhafte Ausprägung oder Transparenz für eine gelungene Darstellung sorgen. Oft sind Verbreitungskarten schon Ergebnis von Auswertungen, die die Verteilung von Objekten zu einem "Pseudoareal" zusammenfassen (vgl. WITT W. 1996, S.220/221 oder ARNBERGER E. 1997, S.132). Die kartographische Fachliteratur bezeichnet diesen Kartentyp auch als Gebietslagekarte oder Gattungsmosaik (vgl. IMHOF E. 1972, S.115). Abbildung 13 zeigt verschiedene Verbreitungen in einer Farbvariation. Da die Flächen sich überschneiden und damit die Grundlage sichtbar bleibt, sind die Flächen transparent.

Abb. 13: Ausschnitt einer Verbreitungskarte

Quelle: Eigene Darstellung.

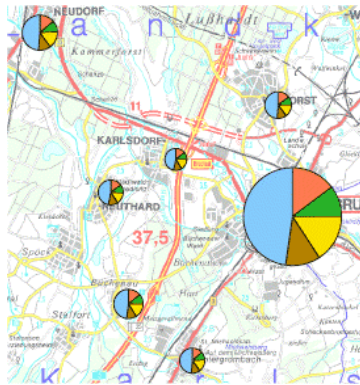
2.2.4.2 Diagrammkarten

Alle thematischen Karten mit Diagramminhalt heißen Diagrammkarten. Unter einem Diagramm versteht man die "zeichnerische Darstellung von Größen- und Mengenverhältnissen" (vgl. LANGENSCHIEDTS FREMDWÖRTERLEXIKON ONLINE). Es werden Quantitäten für bestimmte Geoobjekte dargestellt. Diagramme können ortsbezogen, streckenbezogen oder flächenbezogen sein. Dabei soll klar ersichtlich sein, auf welche räumliche Komponente sich die Diagramme einer Karte beziehen. Die folgenden Kapitel unterscheiden Diagrammkarten nach der Implantation der Gegebenheiten.

2.2.4.2.1 Ortsbezogene Diagrammkarte

Mutiert eine Positionssignatur einer Verbreitungskarte zu einem Diagramm spricht man von einer ortsbezogenen/-gebundenen Diagrammkarte (vgl. ARNBERGER E. 1997, S.34). Dabei ist zu beachten, dass der Diagrammschwerpunkt Referenzpunkt zum Geoobjekt ist. Abbildung 14 zeigt eine Diagrammkarte mit Kreissektorendiagrammen für verschiedene Standorte.

Abb. 14: Ausschnitt einer ortsbezogenen Diagrammkarte



Quelle: Eigene Darstellung.

2.2.4.2.2 Streckenbezogene Diagrammkarte

Streckenbezogene Diagramme werden stark vereinfacht auf eine "Linie" abgebildet. Wichtigste Diagrammtypen sind Linien- oder Banddiagramme ohne Richtungsaussage (Darstellungen mit Aussage über die Richtung ergeben eine Kombination aus streckenbezogenen Diagrammkarten und Vektorenbildern). Banddiagramme können topographisch richtig, vereinfacht, gestreckt oder direkt verlaufen. Dies hängt jedoch vom Maßstab ab (vgl. IMHOF E. 1972, S.192/193).

Im Geomarketing werden Banddiagramme häufig im Bezug auf Straßensegmente (z.B. Konsumverhalten pro Straßensegment) oder für den Warenfluss ohne Richtungsangabe verwendet. Abbildung 15 zeigt einen Ausschnitt einer streckenbezogenen Diagrammkarte basierend auf den Straßenabschnitten.

Abb. 15: Ausschnitt einer streckenbezogenen Diagrammkarte



Quelle: Ausschnitt aus WITT W. 1967, Abb.76.

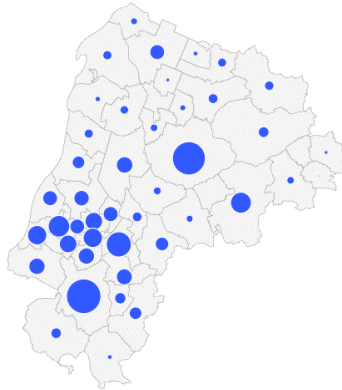
Der Kartenausschnitt zeigt die Verkehrsströme aus den Jahren 1951 (blau), 1955 (violett) und 1959 (rot) anhand von überlagerten Banddiagrammen. Diese verlaufen aufgrund des großen Maßstabes topographisch richtig.

2.2.4.2.3 Flächenbezogene Diagrammkarte

Einem flächenbezogenen Diagramm liegt zusammengefasste Information einer Fläche zugrunde. Das Diagramm liegt dabei im Schwerpunkt des Gebietes. Administrative Grenzen, Postleitzahlen

oder andere Gebietsaufteilungen bilden die Informationsgrundlage. Abbildung 16 zeigt ein Beispiel einer flächenbezogenen Diagrammkarte. Die Kreisdiagramme liegen im Mittelpunkt der Flächen.

Abb. 16: Beispiel einer flächenbezogenen Diagrammkarte



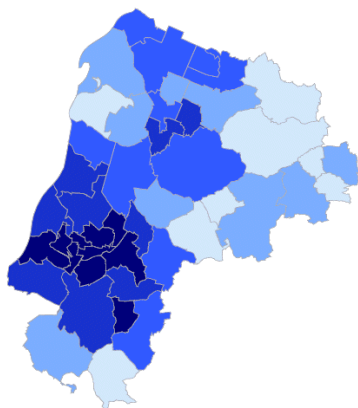
Quelle: Eigene Darstellung.

2.2.4.3 Choroplethenkarten

Choroplethenkarten stellen relative Werte (Verhältniszahlen) flächenbezogen dar. Dabei handelt es sich meist um unechte Flächen, da die dargestellte Zahl für die gesamte Bezugseinheit gilt. Unterschiede innerhalb der Bezugsgebiete werden unterschlagen (vgl. OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.38). Daraus wird ersichtlich, dass die Wahl der Bezugsflächen entscheidend zur Aussagekraft beiträgt. Die thematische Verbreitung eines Sachverhaltes sollte gleichmäßig über die Zuordnungsfläche verteilt sein. Dabei spielt besonders die Geographie eines Gebietes eine entscheidende Rolle. Ein hoher Anteil an Gebirge oder Gewässer lassen keine wirklich treffende Aussage zu.

Choroplethenkarten werden anhand von gestuften Verlaufszahlen dargestellt. Die Wertemenge muss deswegen klassifiziert werden (vgl. 2.2.5.4). Eine stufenlose Darstellung wäre technisch unproblematisch, hat jedoch weniger Aussagekraft. Die schematische Zusammenfassung der Werte zu sinnvollen Gruppen eignet sich besser, um Gegebenheiten zu erläutern (vgl. WITT W. 1967, S.295). Abbildung 17 zeigt ein Beispiel einer Choroplethenkarte. Dargestellt wird die Bevölkerungsdichte anhand von fünf Klassen.

Abb. 17: Beispiel einer Choroplethenkarte



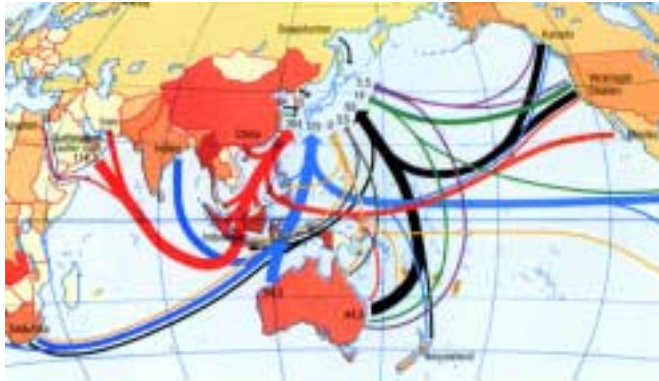
Quelle: Eigene Darstellung.

Hinweis: Hinzufügen ist, dass in der deutschsprachigen Literatur auch von Dichtemosaik (vgl. IMHOF E. 1972, S.163-176), Mosaikkartogrammen (vgl. WILHELMY H. 1996, S.221/222) oder Flächenmosaik (vgl. ARNBERGER E. 1997, S.129) gesprochen wird.

2.2.4.4 Darstellung von Bewegung und Genese

Bei der Darstellung von Bewegung und Genese unterscheidet man zwischen zwei Kartentypen. Zum einen sind statische Darstellungen zu nennen, die dynamische Vorgänge vermitteln. Meist wird dabei auf Darstellung von Bewegung mit quantitativer Aussage und Richtungsangaben zurückgegriffen (vgl. Abbildung 18).

Abb. 18: Ausschnitt einer statischen Darstellung von Bewegung



Quelle: transformierte Darstellung aus dem DIERCKE WELTATLAS 1988, S.231, Japan – Rohstoffabhängigkeit (1985).

Weiter dienen Kartenserien zur Darstellung von Genese in statischen Karten mit dynamischem Inhalt. Beide Darstellungstypen wurden jedoch als problematisch eingestuft (vgl. ARNBERGER 1997, S.162 und WITT W. 1967, S.20).

Zum anderen sind **Computeranimationen** zu nennen. Diese dienen der Kartographie zur Darstellung des temporalen Faktors von Daten (= temporale Animation) und zur Wiedergabe gleicher Zustände in verschiedener Art und Weise (= nontemporale Animation). Computeranimation bietet dadurch zusätzliche Visualisierungsmöglichkeiten (vgl. DRANSCH D. 1997, S.15/16). Diese werden in Webmaps verbessert Bewegung und Genese darstellen können.

Auch im Geomarketing spielen Bewegungen und Genese eine wichtige Rolle. Sie dienen beispielsweise zur Veranschaulichung des Warenflusses oder der Umsatzentwicklung

Hinweis: Es wird darauf hingewiesen, dass eine ausführliche Behandlung dieses Themas den Rahmen der Arbeit gesprengt hätte. Trotzdem ist bei der Wahl der einzusetzenden Technologien Animation berücksichtigt worden, um eine spätere Implementierung nicht unmöglich zu machen (vgl. Kapitel 3.2.2.6).

2.2.5 Die graphischen Elemente einer thematischen Karte

Die graphischen Elemente einer thematischen Karte tragen einen wesentlichen Teil dazu bei, die Aussage einer Darstellung dem Leser zu vermitteln. Karteninhalt sowie die erklärenden Zusatzangaben sind die Bestandteile einer Darstellung. Basierend auf den in Abschnitt 2.2.3 vorgestellten Methoden zur Visualisierung von Daten werden im Folgenden die graphischen Elemente erläutert. Dabei geben die Kartentypen des Geomarketings den Rahmen.

2.2.5.1 Kartenschrift

Kartenschrift, nicht zu verwechseln mit typographischen Signaturen, wird nur selten im direkten Bezug auf das Thema eingesetzt. Meist dient Schrift zur Identifizierung von Geoobjekten in der Basiskarte und zur Erläuterung in Kartenzusätzen. Auch aufgrund dessen, dass Kartenschrift auf den Leser unruhig wirkt, sollte sie nicht zu dominant sein. Das Schriftbild hängt sehr stark von der Art des Ausgabemediums ab, deswegen sind weitere Aspekte bezüglich der Schriftformatierung in Kapitel 2.3.1.3 zu finden.

Hinweis zur Schrift in Karten: Bei thematischen Darstellungen kann auf diese weitgehend verzichtet werden, da Namen in diesen Karten weniger von Bedeutung sind. Schriftzusätze können beispielsweise durch **Tooltip**'s wiedergegeben werden (vgl. Kapitel 2.3.2.2.1).

2.2.5.2 Signaturen einer Karte

IMHOF bezeichnet Signaturen als "genormte, uniformierte Kleinfiguren", die für bestimmte Objektgruppen stehen (vgl. IMHOF E. 1972, S.58). Die Objektgruppen sind dabei meist nominal skaliert. Signaturen besitzen punkt-, linien- oder flächenhafte Ausprägung. Zur Gestaltung von Signaturen werden die visuellen Variablen BERTIN's herangezogen (vgl. Kapitel 2.2.3.3). Für die jeweilige geographische Ausprägung gelten die in Abbildung 10 vorgeschlagenen Variablen. Im Folgenden wird besonders auf punktförmige Signaturen eingegangen, die in thematischen Karten dominieren. Punkthafte Signaturen unterscheidet man in sprechende (auch Figurenbilder genannt) und geometrische. Erstgenannte besitzen die Fähigkeit, selbsterklärend zu sein. Wie schon angesprochen, sind im Geomarketing piktogrammähnliche Firmensymbole beliebt. Sind diese zu "bildlich", sollte auf geometrische Signaturen zurückgegriffen werden. Geometrische Signaturen dagegen setzen sich aus geometrischen Figuren zusammen, die durch eine ausführliche Legende erklärt werden müssen. Geometrische Signaturen passen sich durch geeignete Darstellung dem Kartenbild besser an. Weitere Merkmale geometrischer Signaturen sind Gruppen- und Kombinationsfähigkeit. Beispielsweise erhalten alle Signaturen eines Themas die gleiche Farbe oder Form. Grundsätzlich ist hinzuzufügen, dass das Überlagern von Signaturen vermieden werden sollte. Weiter sind Signaturen freizustellen (→ **Freistellung**), um sich vom Kartenhintergrund abzuheben. Eingeschränkt kann durch punkthafte Signaturen auch eine Aussage über Größen oder Bedeutung verwirklicht werden. Ordinalskalierte Aussagen, wie klein/groß/mittel bestimmen dann das Signaturendesign. Die Berücksichtigung von Quantitäten führt zu Flächen-/Symbolfigurendiagrammen, die in Kapitel 2.2.5.3.1 beschrieben sind (vgl. OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.36/37).

Hinweis: Signaturen tragen wesentlich zum graphischen Bild einer Karte bei. Wären sie global normiert, müsste aufgrund dessen, dass Signaturen mit anderen Kartenelementen abgestimmt werden, jede Karte "gleich" aussehen. Daher sind Signaturen nur innerhalb von Kartenwerken respektive Kartenviewern normiert. Eine Normierung ist noch aus weiteren Gründen, wie der unendlichen Möglichkeit, Zustände mit Signaturen darzustellen, nicht praktikabel (vgl. IMHOF E. 1972, S.61).

2.2.5.3 Diagramme zur Darstellung von Information

Um statistische Informationen zu visualisieren, bedient man sich den Diagrammen. Platziert in einer Karte, geben diese sowohl einen akzeptablen Gesamtüberblick über eine Wertemenge, als auch detaillierte Werteangaben. Die bessere Überschaubarkeit spiegelt sich am Besten im Vergleich zu Wertetabellen wieder. Tabellen, als alternative Form der Präsentation müssen "gelesen" (IMHOF E. 1972, S.72) werden. Jedoch muss im Gegensatz dazu gesagt werden, dass Diagramme kein Allheilmittel für die Darstellung komplizierter Statistiken durch noch komplexere Diagrammfiguren sind. Eine Karte kann schnell missinterpretiert oder überladen werden (vgl. IMHOF E. 1972, S.72/73). Um gute Diagramme zu gestalten, spielen wie auch bei der Signaturgestaltung die visuellen Variablen eine entscheidende Rolle (vgl. Kapitel 2.2.3.3). Außerdem müssen die folgenden allgemeinen Grundsätze zum sinnvollen Einsatz von Diagrammen in thematischen Karten befolgt werden:

- **Wahl des Diagrammmaßstabes:** Die richtige Wahl des Diagrammmaßstabes ist sehr wichtig. Sie bestimmt die Darstellung (größtmögliches Diagramm), wobei Unterschiede dennoch sichtbar bleiben sollten. Obwohl Überschneidungen möglichst zu vermeiden sind, kommen sie oft in Thematischen Karten vor. Dann gilt, kleine Diagramme stehen im

Vordergrund und sollten sich beispielsweise durch einen Rand von dem darunter liegenden Diagramm abgrenzen (→ **Freistellung**).

- **Gestaltung von Diagrammen:** Die Diagramme in der Karte sollten möglichst einfach gehalten sein, in der Legende folgt eine genauere Erläuterung. Das heißt auch, dass möglichst wenig Diagrammbeschriftung in der Karte erfolgt. In Bildschirmkarten können Ereignisse (*engl. Events*) sehr nützlich sein, um beispielsweise beim Überfahren eines Diagramms mit der Maus, weitere Informationen anzuzeigen.
- **Kopplung/Kombination von Diagrammen:** Diagramme können beinahe beliebig gekoppelt beziehungsweise kombiniert werden, jedoch sollte die Anordnung für jede Diagrammkombination/-kopplung in einer Karte konsequent durchgeführt werden. Auch Geomarketing bedient sich Diagrammkombinationen, wie gegenüberliegenden Halbkreisdiagrammen.
- **Akzentuierung von Diagrammen:** Diagramme müssen sich vom Rest der Karte gut, aber nicht übertrieben abheben.
- **Räumliche/perspektivische Diagramme:** Meist werden 3D-Diagramme nur dekorativ eingesetzt, was in ganz wenigen Fällen sogar sinnvoll sein kann (je nach Kartenbild). Vorwiegend verfälscht dieser Diagrammtyp aber die Aussage, weicht vom Grundsatz einer simplen Darstellung ab und erfordert hohe Ansprüche an die menschliche Vorstellungskraft. Volumina sind schwer in ihrer Größe abschätzbar. 3D-Diagramme sind bei besonders großer Wertespanne bedingt sinnvoll einsetzbar. In der Kartographie gilt es sie zu vermeiden.
- **Platzierung von Diagrammen:** Die automatische Platzierung und Skalierung von Diagrammen/Signaturen ist noch nicht vollständig erforscht (Forschungsgebiet der Informatik) und bedarf noch der Optimierung.

Diagrammtypen gelten auch ohne Karte, beispielsweise in Graphiken, als eine geeignete Darstellungsform. Im Folgenden die wichtigsten, längst aber nicht alle Diagrammtypen.

Hinweis: Die folgende Gliederung richtet sich nach dem Aussehen der Diagramme und ist im Hinblick auf die Prototyprealisierung aber zweckmäßig. Abbildungen der Diagrammtypen sind im Anhang (Cd-Rom) zu finden.

2.2.5.3.1 Flächen-/Symbolfigurediagramme

Flächendiagramme zeigen Werte als proportionale Flächen von geometrischen Figuren, wie Kreis oder Quadrat (vgl. BOUNFORD T. & CAMPBELL A. 2001. S.72). Weitere Figuren, wie sprechende Signaturenformen, die einen Hinweis auf das Thema geben, können ebenso verwendet werden (vgl. ARNBERGER E. 1997, S.49). Die Quantitätsaussage drückt sich auch bei Symbolen in der Fläche aus (vgl. OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.50).

2.2.5.3.2 Kreissektorendiagramm

Kreissektoren- oder Tortendiagramme gehören auch zur Gruppe der Flächendiagramme und geben neben der quantitativen Gesamtaussage auch noch die Möglichkeit, diese in verschiedene Qualitäten zu unterteilen. Mehrere Merkmalsausprägungen können in den verschiedenen Sektoren dargestellt werden. Der Sektorenwinkel sollte nicht zu klein werden, da der Wert ansonsten nicht mehr erkennbar ist (BERTIN drückt diesen Effekt mit einer zu langen Komponente aus; vgl. BERTIN J. 1974, S.211). Ist das trotzdem der Fall, müssen kleine Werte zu größeren Gruppen zusammengefasst werden. Der erste Sektor ist der wichtigste und beginnt immer bei "12 Uhr". Akzentuierung wird vor allem durch die Flächenfüllung (z.B. Farbe/Muster) der Sektoren gesetzt. Dabei gilt, je kleiner der Sektor, desto kräftiger die Füllung (z.B. Farbe); für große Sektoren umgekehrt. Kreissektorendiagramme (*engl. pie charts* oder kurz "pies") sind zeitlose Darstellungen,

also ungeeignet für temporale Veränderungen. Selten findet man größengleiche Tortendiagramme, in denen nur die Teilrelationen dargestellt werden und die Größe aller Diagramme konstant bleibt.

2.2.5.3.3 Säulen-/Balkendiagramm

Ein Säulen-/Balkendiagramm setzt sich aus einem kartesischen Koordinatensystem, in dessen Raster Säulen/Balken der Werte entsprechend, die Länge abgetragen wird, zusammen. Die Grundlinie kann horizontal oder vertikal angelegt werden. Balken und Säulen sind immer gleich breit zu wählen und entsprechend einzufärben, ähnlich der Tortendiagramme. Zusätzlich können dezente Hilfsraster im Diagrammhintergrund zur verbesserten Interpretation der Werte miteinfließen (vgl. BOUNFORD T. & CAMPBELL A. 2001, S.54). Unterschieden werden:

- **Balken:**
Jede Qualität steht, ausgedrückt durch einen Balken, für sich. Abstände zwischen den Balken müssen nicht bestehen.
- **Säulen:**
Alle Qualitäten werden zusammenfassend in einer Säule dargestellt. Die Höhe dieser wird also aus der Summe der einzelnen Werte/Höhen gebildet. Diese alternative Darstellung zum Tortendiagramm hat Vorteile (nicht breit, da nur eine Säule!) bei nicht so großen Summenintervallen. Man spricht auch von gestapelten oder unterteilten Balkendiagrammen.

2.2.5.3.4 Stabdiagramme

Stabdiagramme setzen sich aus einem kartesischen Koordinatensystem und in dessen Raster eingetragenen Stäben zusammen. Deren Grundlinie wird horizontal oder vertikal angelegt. Stab- oder Stäbchendiagramme sind das Resultat vieler aneinandergereihter Werte. Die Stäbe sind möglichst schmal zu halten (je nach Anzahl der Werte) und sollten durch einen kleinen Zwischenraum voneinander abgegrenzt werden (BAHRENBERG G. & GIESE E. & NIPPER J. 1999, S.33). Stäbe besitzen im Gegensatz zu Balken oder Säulen keinen Rand und sind meist einfarbig. Anwendung findet diese Art von Diagrammen bei zeitlichen Veränderungen. Sie sind dabei besonders zur Darstellung von Werten geringer Dichte (= Zeitintervall zwischen den Messwerten ist relativ groß) geeignet. Eine besondere Art sind die geklappten Stabdiagramme, die sowohl positive als auch negative Stäbchen abbilden.

2.2.5.3.5 Kurvendiagramme

Kurven-/Liniendiagramme stellen die Korrelation zweier Werte in einem arithmetischen Koordinatensystem dar. Logarithmische oder andere Koordinatensysteme sollten nur falls zwingend erforderlich, verwendet werden. Nicht jeder ist in der Lage, die damit verbundenen Verzerrungen richtig zu interpretieren. Die Ursache (dazu zählt auch die Zeit) wird immer auf der Abszisse, der Effekt (Merkmalsausprägung) stets auf der Ordinate dargestellt. Falls die Daten kein anderes Format erfordern, sollte ein Kurvendiagramm im Querformat angelegt werden. Die Kurve ist deswegen einprägsamer und Werte sind besser unterscheidbar. Teile der Achsen können gestaucht werden, damit Details besser sichtbar werden. Hilfsrasterlinien sind, solange diese dezent im Hintergrund bleiben, eine Methode, die Diagramme noch verständlicher zu gestalten. Bei Diagrammen mit nur einer Kurve ist es ratsam, die Kurve "einzufangen". Die Fläche zwischen der Abszisse und der Kurve wird hervorzuheben, indem diese beispielsweise leicht eingefärbt wird (vgl. BOUNFORD T. & CAMPBELL A. 2001, S.44). Im Gegensatz zu Stabdiagrammen darf die Wertedichte, das heißt der zeitliche Intervall der Messwerte, nicht zu klein sein. Kurvendiagramme eignen sich besonders zur Darstellung von Entwicklungen, Tendenzen sowie zu deren Vergleichen.

2.2.5.3.6 Banddiagramme

Banddiagramme sind der am meisten verbreitete Diagrammtyp der streckenbezogene Darstellungen zulässt. Grundsätzlich müssen Banddiagramme keine Richtung anzeigen, jedoch wird oftmals in Form einer Pfeilspitze darauf zurückgegriffen. Wie in Kapitel 2.2.4.4 erwähnt, ist es je nach Maßstab notwendig, die topographische Ausprägung zu generalisieren. Banddiagramme lassen sich mit den visuellen Variablen Größe (in Form einer gestuften oder gleitenden Breitenvariation des Bandes), Farbe oder Muster realisieren (vgl. ANBERGER E. 1997, S.105-108). Kritischste Punkte für eine automatische Generierung sind der Anfangs- sowie Endpunkt eines Bandes, da an diesen oftmals mehrere verschiedene Bänder zusammentreffen. Es kann zu "hässlichen" Überlagerungen kommen. Wie allgemein bei Diagrammen üblich, liegen dünne Bänder im Vordergrund (vgl. DENT B. 1996, S.226).

2.2.5.4 Gruppenbildung in Choroplethenkarten

Klassifizierung oder Gruppenbildung wird hauptsächlich bei Choroplethenkarten eingesetzt. Grundsätzlich werden Klassen anhand verschiedener Methoden gebildet. (vgl. Abbildung 19) und diesen die entsprechenden Werte zugeteilt. Die verschiedenen Klassen werden primär in einer Variation des Helligkeitwertes gestaltet. Dabei entsprechen die niedrigen Werteklassen hellen Tönen, die hohen Werteklassen dunklen Tönen. Eine unmittelbare Wiedergabe des Einzelwertes ist zwar genauer, doch kann das Wesentliche nur schlecht vermittelt werden (vgl. Kapitel 2.2.4.3). Gruppenbildung prägt das Gesamtbild einer Darstellung entscheidend. Daher sollten die folgenden allgemeinen Anforderungen an die Klassenbildung unbedingt beachtet werden. Sie stellen eine Kombination der Ausführungen von WITT, BAHRENBURG et al und OLBRICH et al dar:

- Klassen dürfen sich nicht überdecken (eindeutige Klassengrenzen bilden).
- Das gesamte Werteintervall muss abgedeckt werden. Klassenmitten und Klassengrenzen sollten "einfache" Zahlen sein und die Klassen müssen geschlossen sein.
- Die Klassenanzahl steht in einem angepassten Verhältnis zu den vorliegenden Daten, und darf nicht die Anzahl der maximalen graphischen Unterscheidbarkeit, beispielsweise die Segmente einer Farbskala, überschreiten.
- Wahl der für die Daten geeigneten Klassenbildungsmethode.

(vgl. WITT W. 1967, S.296/297; BAHRENBURG G. & GIESE E. & NIPPER J. 1999, S.32 und OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.40).

In Abbildung 19 wird besonders auf die Klassifizierung bei Choroplethenkarten eingegangen. Bei diesen sind mehrere Klassifizierungsmethoden möglich:

Abb. 19: Klassifizierungsmethoden bei Choroplethenkarten

Art der Methode	Beschreibung / Beurteilung		Vorgehen / Beispiel
Willkürlich	Gruppen werden nach einer geeigneten, aber willkürlich gewählten Skala gebildet. Eine nicht sehr sinnvolle Methode, ermöglicht aber den quantitativen Vergleich zu anderen Gebieten, da sich Skala weder nach Werteintervall oder Anzahl der Werte noch nach anderen Gesichtspunkten richtet.		0-<5,5-<10,20-<25,... oder 0-<10,10-<25,25-<50,...
Mathe- matisch	Es werden geringster und größter Wert ermittelt und nach mathematischen Prinzipien unterteilt. Werte-stufen haben überhaupt keinen räumlichen Bezug, trotzdem weit verbreitet. Nebenstehend vier übliche Möglichkeiten	Stufen nach Quantilen, gleich viele Bezugsflächen pro Klasse	Berechnen der Klassengröße (Verhältnis Anzahl der Klassen zu Gesamtweranzahl) sortierte Wertemenge klassifizieren
		Stufen nach gleichen Intervall (=arithmetische Reihe 1. Ordnung)	0-<5,5-<10,10-<15, ...
		Stufen nach arithmetischer Progression (=arithm. Reihe 2. Ordnung)	der Abstand zweier aufeinander folgender Glieder wächst um einen konstanten Betrag
		Stufen nach geometrischer Progression	konstantes Zahlenverhältnis zwischen den Mittel-/Grenzwerten benachbarter Gruppen
Natürlich	Bildung von Häufigkeitsdiagrammen, gestuft wird dann nach charakteristischen Gruppen, gute Vergleichbarkeit, geographisch unbrauchbar.		Schwellwerte (meist Minima) eines Histogramms werden Klassengrenzen
Räumlich	Bildung der räumlichen Verteilung der charakteristischen Gruppen, und Zusammenschließung dieser. Schlechte Vergleichbarkeit, sehr bearbeiterabhängig und zeitaufwendig		Quantilen unter Berücksichtigung des Histogramms

Quelle: Eigene Darstellung basierend auf der Ausführung von HAKE G. & GRÜNREICH D. 1994, S.437/438; ARNBERGER E. 1997, S.93/94 und WILHELMY H. 1996, S.311-313.

Über die beste Klassenbildungsmethode wird heftig gestritten, Statistiker/Informatiker ziehen die mathematische, natürliche oder auch die statistische vor. Dagegen neigen Kartographen/Geographen zu einer räumlichen Gruppenbildung; unschwer daran zu erkennen, dass WITT diese Forderung als erste in seiner Aufzählung erwähnt (WITT W. 1967, S.296). Auswirkungen der Ergebnisse bei verschiedenen Klassifizierungsmethoden behandeln OLBRICH et al, DENT und WITT (vgl. OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.39-46; DENT B. 1996 S.124-138 und WITT W. 1967 S.307-317). Eine besondere Hilfestellung kann schon die geeignete Wahl der Bezugseinheiten nach geographischen Gesichtspunkten geben. So gelten beispielsweise unbewohnbare Gebiete, wie das Hochgebirge oder große Wasserflächen, als nicht analyserelevant und werden auch in der Darstellung nicht berücksichtigt.

Die geographische Gruppenbildung ist für den praktischen Einsatz im Geomarketing nicht geeignet, da sie sehr stark von der subjektiven Betrachtungsweise des Bearbeiters abhängt. Außerdem werden fast alle Karten des Geomarketings aufgrund ihrer Komplexität automatisch von Computerprogrammen erstellt. Diesen ist es nicht möglich, die komplizierten Faktoren, die für eine geographische Analyse notwendig sind, zu berücksichtigen.

Hinweis: Ein kartographischer Viewer kann individuelle Klassifizierung anbieten, die der Anwender durch Interaktion, beispielsweise in Form eines Histogramms mit Schieberegler, vornimmt (vgl. EMC3 2002 und ANDRIENKO G. & ANDRIENKO N. 1999, Kapitel 4).

2.2.5.5 Kartengrundlagen

Die Kartengrundlage einer thematischen Karte gibt dem Sachverhalt den geographischen Bezug und dient zur räumlichen Orientierung (vgl. ARNBERGER 1997, S.37). Für die Platzierung der thematischen Inhalte sind Koordinatenangaben ausschlaggebend. Diese und die Kartengrundlage bilden die Basisdaten einer Darstellung. Die Güte dieser ist häufig von den verfügbaren finanziellen Mitteln und den Werkzeugen für das "Datenhandling" abhängig. Im Folgenden die wichtigsten Gedanken zu Basisdaten.

2.2.5.5.1 Basisdaten als geographische Information

Bezüglich der Wahl der Basisdaten ist es wichtig, welche Daten als Grundlage dienen sollen und wie detailliert sie vorliegen müssen. Zum ersten Punkt ist zu sagen, dass nur das Nötigste in der

Karte enthalten sein sollte, damit die Kartengraphik nicht überladen wirkt. Dabei ist das Thema und der Inhalt entscheidend (vgl. IMHOF E. 1972, S.214). Wie detailliert die Daten sein müssen, entscheiden mehrere Faktoren. Zum einen sind die üblichen Anforderungen an die Generalisierung zu nennen, zum anderen muss besonders bei Bildschirmkarten auf die Datenmenge geachtet werden. Das System sollte in einer akzeptablen Geschwindigkeit arbeiten und trotzdem eine gute Darstellung ermöglichen. Diese Faktoren bedingen sich gegenseitig (vgl. Kapitel 2.2.5.5.3).

Hinweis: Bei der Gestaltung der Basiskarte legt man auch schon den Grundstein für ein Gesamtlayout. Von diesem hängt ab, ob eine Rahmenkarte, die die benachbarte Regionen mit einbezieht, oder eine Inselkarte, die nur das darzustellende Gebiet zeigt, entsteht.

2.2.5.5.2 Projektion und Koordinatensystem einer Darstellung

Aufgrund dessen, dass Karten ein zweidimensionales Abbild eines dreidimensionalen Objektes sind, müssen sie projiziert werden. Allgemein gelten die kartographischen Grundsätze zur Wahl der besten Projektion. Je nach Lage auf der Erdkugel und Maßstab der Karte wird eine geeignete Projektion ausgewählt. Jedoch ist zu beachten, dass für Webmappingzwecke Projektionen mit kartesischen Koordinatensystemen allgemein üblich sind. Eine ausführliche Diskussion der verschiedenen Projektionen würde hier zu weit führen, können aber der Fachliteratur entnommen werden (vgl. dazu OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.29-32 und HAKE G. & GRÜNREICH D. 1994, S.42-80). Karten des Geomarketings, die sich hauptsächlich Mikroregionen widmen, stellen nicht zu hohe Ansprüche an die Projektion. Probleme ergeben sich bei Webmapping-Systemen, die Kontinente und gleichzeitig Mikroregionen in unterschiedlichen Zoomstufen darstellen sollen. Hier müssen teilweise abenteuerliche Wege beschritten werden, um die geeignete Projektion zu finden.

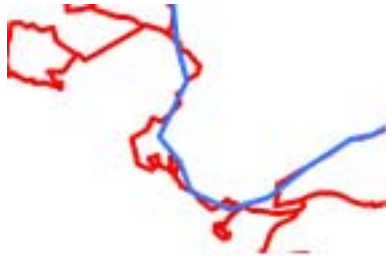
Hinweis: Die Wahl der Projektion und des Koordinatensystems hängt in der Praxis stark von der eingesetzten Technologie ab. Dabei ist es wichtig zu wissen, welche Koordinatensysteme vom eingesetzten Viewer am besten verarbeitet werden können. Im Allgemeinen sind das rechtwinklige Koordinatensysteme. *On-the-fly* Umrechnungen sollten vermieden werden, da diese erheblich zur hohen Systemauslastung beitragen. Für Deutschland eignet sich besonders "UTM Zone 32, nördliche Hemisphäre" auf beispielsweise einem **WGS 84** Ellipsoiden.

Längen- und Breitenkreise des Koordinatensystems können bei kleinmaßstäblichen Darstellungen als dezent wirkende Linien mit in die Basiskarte aufgenommen werden.

2.2.5.5.3 Generalisierung von geographischen Objekten

Die Generalisierung bezeichnet den Vorgang, der die reale Topographie für die Kartendarstellung stark vereinfacht. Der Generalisierungsgrad bestimmt dabei, wie genau Topographie dargestellt wird. Eine Generalisierung der darzustellenden Geodaten ist ratsam, oftmals sogar erforderlich. Diese kann vor (dann würden die Daten bereits generalisiert in der Datenbank vorliegen), beziehungsweise während der "Kartenproduktion" auf dem Server erfolgen (On-the-fly Generalisierung). Letztere wird aufgrund der hohen Serverauslastung selten eingesetzt. Generalisierung bringt auch Datenreduzierung mit sich. Ein erfreulicher Nebeneffekt, solange die Datenmenge nicht stärker reduziert werden muss, als das optimale Generalisierung erfordert. Die Praxis zeigt (z.B. bei **NAVTECH** Daten), dass besonders bei Mikroregionen, wie beispielsweise Postleitzahlengebieten, eine Generalisierung bezüglich der Datenmenge nicht viel einbringt. Mikroregionen werden meist nur in großen Maßstäben angezeigt. Anders dagegen bei Bundesländern, die aufgrund der großen Datenmenge Generalisierung erfordert.

Im Zuge einer Geodatengeneralisierung ist auf folgendes graphisches Problem hinzuweisen, das ungelöst ist. Unterschiedliche Generalisierungsgrade der Polygongrenzen in verschiedenen Levels führen zu unschönen Überschneidungen, wie in Abbildung 20 zu sehen ist (vgl. STEINGRUBE W. 1997, S.45-47).

Abb. 20: Graphische Probleme im Zuge einer Generalisierung

Quelle: Eigene Darstellung nach STEINGRUBE W. 1997, S.45-47.

Hinweis: **Blau** stellen eine generalisierte Bundeslandgrenze, **rot** eine original Gemeindegrenze dar.

Zum Thema Generalisierung verbunden mit räumlicher Navigation ist noch das **adaptive Zoomen** zu erwähnen. Dabei passen sich Karteninhalte beim Zoomen automatisch an. Zum Beispiel werden bei der Vergrößerung einer Karte Details nachgeladen und visualisiert. Webmapping-Systeme schenken diesem Thema immer größere Bedeutung. Weitere Hinweise und Forschungsergebnisse zum adaptivem Zoomen geben BRÜHLMEIER und CECCONI & GALANDA (vgl. BRÜHLMEIER T. 2000, S.43-58 und CECCONI A. & GALANDA M. 2002).

2.2.5.6 Erläuternde Kartenzusätze

Um weitere wichtige Erklärungen bezüglich einer thematischen Karte zu erhalten, bedarf es noch folgender Zusatzelemente.

2.2.5.6.1 Der Titel einer Karte

Der Titel einer Karte sollte das Thema unmissverständlich nahe bringen. Zeitangabe (Zeitpunkt/-raum) und räumliche Angabe sind nicht immer erforderlich (z.B. Atlanten). Der Titel sollte nicht zu lang gewählt werden und kann gegebenenfalls mit einem Untertitel genauer spezifiziert werden (vgl. HAKE G. & GRÜNREICH D. 1994, S.451). Auch bei automatisch/dynamisch erzeugten Karten ist es wichtig einen aussagekräftigen und passenden Titel zu finden. Für diesen müssen aber andere Anforderungen gelten. Beispielsweise ist das Datum einer laufend aktualisierten Karte zu lang, um es im Titel zu platzieren.

2.2.5.6.2 Legende zur Erklärung des Inhaltes

Die Legende soll erklären, wie das in der Karte visualisierte Thema zu verstehen ist. Beispielsweise müssen Signaturen und Diagramme erläutert werden. Sie ist ein zwingender Bestandteil einer thematischen Karte und trägt wesentlich zur richtigen Interpretation dieser bei (KRAAK M.-J. 2001b, S.65). Legendenkonzepte in Abhängigkeit der Kartentypen behandelt beispielsweise OLBRICH et al (vgl. OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.88-90).

Bei Bildschirmkarten können Legenden als eigene "Fenster" realisiert werden. Zoomevents erfordern auch die Skalierung der Legende, so dass Größenrelationen beibehalten werden.

2.2.5.6.3 Maßstabsleiste als Relationsangabe

Der Maßstab ist eine Kenngröße für den Grad der Verkleinerung einer Darstellung zur Realität. Er ist zwingend notwendig und dient zur räumlichen Relation. Dies geschieht entweder mittels eines numerischen Maßstabs oder einer Maßstabsleiste. Diese stellt eine bestimmte Strecke mit deren zugehörigen Referenzwert dar. Die Maßstabsleiste sollte gerundete Zahlen verwenden, sich daher auch mitskalieren lassen. In der Bildschirmdarstellungen mit "Zoominteraktion" sollte ein Maßstabsbalken eingesetzt werden, da stufenloses Vergrößern die Maßstabszahl unrund werden lassen (OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.90/91).

2.2.5.6.4 Übersichtskarte zur räumlichen Orientierung

Übersichtskarten erleichtern die Orientierung und sind besonders bei dynamisch erzeugten Bildschirmkarten unabdingbar. Für den Karteninhalt reicht eine räumlich charakteristische Umrandung aus. In dieser sollte das aktuelle Kartenfenster, beispielsweise durch ein Rechteck oder Fadenkreuz, symbolisiert werden. Bei Bildschirmkarten kann die Übersichtskarte zusätzlich auch Funktionalität erhalten (vgl. Kapitel 2.3.2.2.2).

2.2.5.6.5 Sonstige Kartenzusätze

Autor-/Herstellerangaben:

Kartenautor und Hersteller müssen in einer Art Impressum angegeben sein. Erforderlichenfalls kann zusätzlich ein Copyright-Vermerk eine Darstellung vor Vervielfältigung schützen (vgl. IMHOF E. 1972, S.246). Bei automatisch generierten Karten wird auch die Institution/Firma, die die Technik für die Kartengenerierung zur Verfügung stellte, angegeben.

Quellenangaben:

Als Quellenangaben zählen alle Datenquellen mit, auch der zeitliche Bezug der Daten. Diese sind notwendig, damit Zeitpunkt und Datenherkunft bekannt sind und überprüft werden können. (vgl. IMHOF 1972, S.246-253).

Nordpfeil:

Ein Nordpfeil ist nicht zwingend notwendig, außer, die Nordrichtung zeigt nicht zum "oberen" Rand der Karte (vgl. IMHOF E. 1972, S.246 und OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.91).

Zusätzliche Darstellungen:

Zusätzliche Darstellungen können die Karte beziehungsweise das Thema in Form von Text, Tabelle und Graphik ergänzen (z.B. Tabelle der Bezirksabkürzungen mit zugehörigen Namen). Verschiedene Möglichkeiten werden in Abschnitt 2.3.2.2.1 diskutiert.

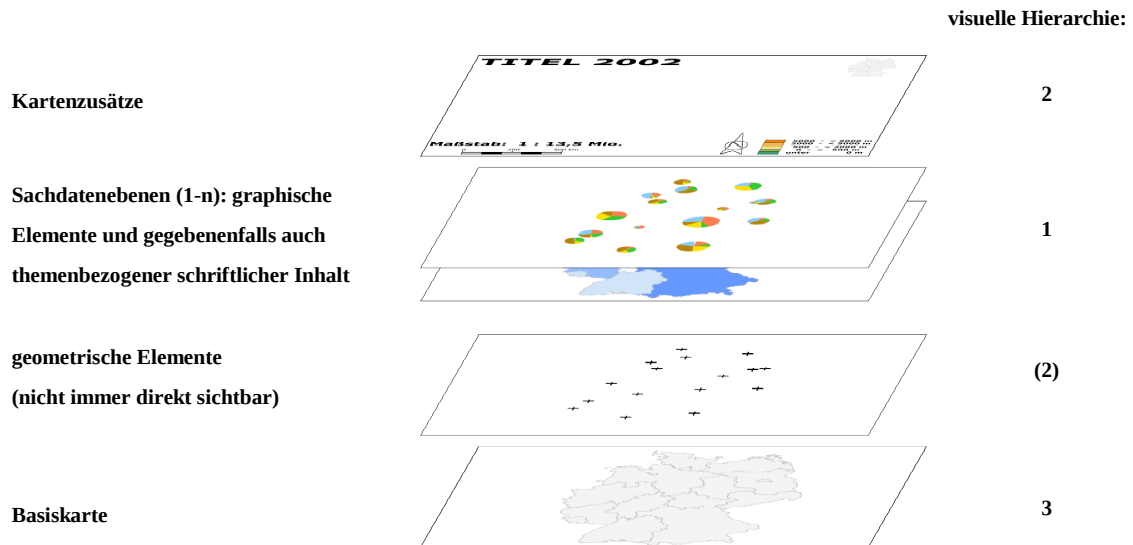
Werkzeuge:

Im Gegensatz zu **Hardcopy** Karten sind Werkzeuge, die einem Benutzer Interaktion mit dem Computer erlauben, bei kartographischen Viewern zwingend erforderlich.

2.2.6 Zusammenführung aller Kartenelemente

Die Zusammenführung aller Elemente ist besonders mit dem Hintergrund der optimalen visuellen Hierarchie (bezüglich des Themas und Zwecks) wichtig. Durch diese wird festgelegt, wo die Hauptaussagen einer Darstellung liegen. Dabei wird eine Karte nach ihrem Inhalt in verschiedene Ebenen zerlegt und ein Informationsschichtenmodell aufgebaut (Abbildung 21).

Abb. 21: Informationsschichten und visuelle Hierarchie thematischer Karten



Quelle: Eigene Darstellung nach OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S.24 und DENT B. 1996, S.256.

Die zusammengefügte Informationsschichten ergeben eine thematische Karte. Die graphische Prägnanz wird durch die visuelle Hierarchie der einzelnen Schichten bestimmt, wobei die Sachdatenebene am wichtigsten ist. Die Basiskarte und der thematische Layer können noch mal in sich in visuelle Schichten unterteilt werden (vgl. DENT B. 1996, S.256-269). Das Kartenlayout, genauer die Anordnung der Elemente, ist stark abhängig von verschiedenen Faktoren des Ausgabemediums, wie Größe oder optisches Zentrum der Ausgabefläche.

2.3 Gestaltung von Bildschirmkarten

Selbst bei Beachtung aller Grundlagen thematischer Karten ist trotzdem nicht gewährleistet, dass die Darstellungen hohe Aussagekraft besitzen. Das liegt daran, dass für Bildschirmkartographie noch zusätzliche Eigenschaften von Bedeutung sind. Auch neue Möglichkeiten der Informationsvermittlung gehören dazu. In den folgenden Kapiteln werden diese angesprochen.

2.3.1 Kriterien einer Bildschirmkarte

Aus dem Webdesign bekannte Kriterien zur Gestaltung von Bildschirmdarstellungen gelten auch für Bildschirmkarten. Jedoch müssen diese um kartographische Gesichtspunkte erweitert werden. Allgemein gilt, dass Inhalt und Graphik bei Bildschirmkarten reduziert werden muss, um das Kartenbild zu vereinfachen (DICKMANN F. 2001, S.161). Möglichkeiten dazu aufgrund der Eigenschaften des Mediums Bildschirm diskutieren die nächsten Abschnitte.

2.3.1.1 Format und Größe der Darstellungen

Das Format und die Größe einer Karte beziehungsweise der dargestellten Objekte muss für eine Bildschirmausgabe in relativen Werten erfolgen. Die daraus resultierenden absoluten Größen einer Karte richten sich nach dem sichtbaren Bildformat, der graphischen Auflösung des Systems, der Größe des Kartenfensters, der Software und den eventuell noch eingeblendeten Menüleisten. Daher kann fast hundertprozentig davon ausgegangen werden, dass die absolute Größe einer Karte bei jedem Anwender differiert. Im Vergleich zu gedruckten Karten ist eine Bildschirmkarte generell

"kleiner" als eine Papierkarte. Jegliche Definition der Größe sollte in "Pixel" angegeben werden. Selbst großflächige Darstellungen mittels **Beamern** sind lediglich wegen der optischen Vergrößerung als großformatig anzusehen. Da deren Auflösung meist gering ist, gelten gleiche Anforderungen an die Größe. Ausnahme stellen Spezialbildschirme dar, z.B. große **Plasmabildschirme**, die aber aufgrund der hohen Kosten selten anzutreffen sind. Daher muss davon ausgegangen werden, dass die tatsächliche Bildschirmgröße nicht größer als 22 Zoll ist. In Abbildung 22 werden wichtige Kenngrößen im Zusammenhang mit den beiden gebräuchlichsten Monitortypen, dem Röhrenmonitor (**CRT**) und dem Flachbildschirm (**LCD**), dargestellt:

Abb. 22: Größen elektronischer Bildanzeige an gebräuchlichen Monitoren

Bildschirm- typ	Bild- diagonale [zoll]	tatsächliche Bildschirmgröße [cm x cm]	Pixelgröße [mm] bei Auflösung von		
			1024 x 768	1280 x 1024	1600 x 1200
CRT	15	27 x 20	0.26	0.21	-
LCD	15	30.4 x 22.8	0.30	-	-
CRT	17	31.7 x 23.6	0.31	0.25	0.20
CRT	19	33.6 x 26.9	0.33	0.26	0.21
LCD	17	33.8 x 27.0	-	0.26	-
LCD	18,1	35.9 x 28.7	-	0.28	-
CRT	21	37.6 x 30.1	0.37	0.29	0.24
LCD	19	37.6 x 30.1	-	0.29	0.24

Quelle: Eigene Darstellung nach KUHLMANN U. 2002, S.126/127 und BRUNNER K. 2001, S.235.

Hinweise zu Monitoren: Die Bilddiagonale einer Röhre kann nicht voll ausgenutzt werden, daher ist die tatsächliche Bildschirmgröße kleiner als bei entsprechenden LCD Geräten. Weiter sei erwähnt, dass Flachbildschirme (**TFT**) nur in einer bestimmten empfohlenen Auflösung gute Darstellungen bringen, CRT Monitore dagegen bei verschiedener Auflösung sinnvoll eingesetzt werden können. Die tatsächliche Bildschirmgröße von Röhrengeräten kann bei unterschiedlicher Auflösung geringfügig schwanken, deswegen weichen die Werte in der Literatur voneinander ab (vgl. KUHLMANN U. 2002, S.126/127).

Als Richtmaß kann eine Bildschirmdiagonale von ca. 32 x 23 cm angesehen werden. Die allgemein verwendete Auflösung beträgt 1024 x 768 Pixel.

2.3.1.2 Farbe und Helligkeit

Die Anwendung der visuellen Variablen Farbe und Helligkeit sind ein weiterer entscheidender Unterschied zwischen Bildschirm- und Papierkarte. Bildschirmkarten verwenden im Gegensatz zu Papierkarten den **RGB**-Farbraum. Traditionelle Farbkonzepte müssen deswegen überdacht werden (vgl. TURTSCHI R. 1998, S.318). Auch der Aspekt, dass die gewählte Farbe in jeder Kartenausgabe gleich ist, gilt bei Bildschirmkarten nicht. Hauptgrund sind auch hier divergierende Hard- und Softwarestrukturen der Kartennutzer. Doch es gibt verschiedene Grundregeln, die möglichst vielen Anwendern eine annähernd gleiche Darstellung präsentieren. Zum Einen ist die Wahl der verwendeten **Farbpalette** zu nennen, die möglichst plattformübergreifend verbreitet sein sollte. Die websichere Farbpalette mit 216 Farben (*engl.* websafe) hat sich im Bereich des **WWW** etabliert. Sie ermöglicht einheitliche Darstellung auf unterschiedlichen Systemen. Diese, mit ihrer geringen Anzahl an Farben reicht aus, da am Bildschirm üblicherweise verschiedene Sättigungen einer Farbe nicht in dem Maße wahrgenommen werden können, als dies bei Papierkarten der Fall ist (vgl. FERINGA W. 2001, S.202). Zum anderen sind kartographische Einschränkungen zu machen. Eine Farbskala kann schon ab fünf Stufen am Bildschirm nicht mehr richtig interpretiert werden. Der Simultankontrast führt bei zu vielen Farbstufen dazu, dass Farben in Abhängigkeit

ihrer Umgebung wahrgenommen werden, und somit nicht mehr eindeutig zugeordnet werden können (vgl. CRÜGER I. 2002, Der Simultankontrast).

Dem ist noch hinzuzufügen, dass nicht alle Farben, die im Druck eingesetzt werden, auch für den Bildschirm geeignet sind (vgl. TURTSCHI R. 1998, S.318-323) und somit eine andere Farbgebung in Bildschirmkarten unabdingbar macht. Für einen Viewer ist zudem ein einheitliches Farbkonzept nötig.

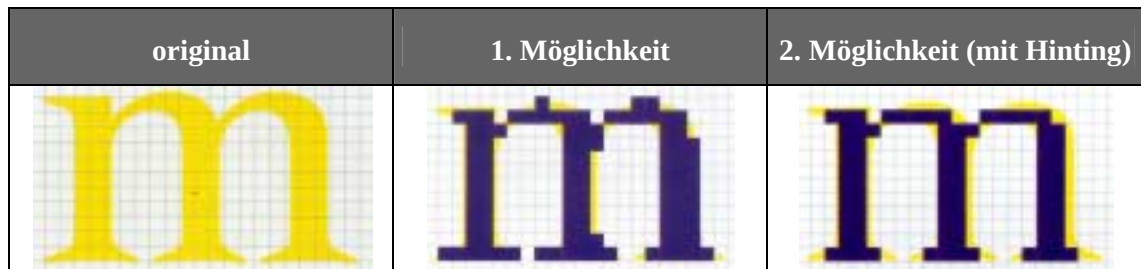
Abschließend ist zur Helligkeit noch zu sagen, dass diese sehr stark von den Monitoreinstellungen abhängig ist und der Kartograph deswegen keinen direkten Einfluss darauf hat.

2.3.1.3 Schrift und graphische Objekte

Schrift und graphische Objekte sind die filigransten Kartenelemente. Aufgrund der geringeren Auflösung von Bildschirmkarten müssen Konzepte zur Darstellung dieser erörtert werden. Bei der typographischen Gestaltung macht sich besonders der Unterschied zwischen der Auflösung eines Bildschirms (gering) und der einer Druckmaschine (hoch) bemerkbar. TURTSCHI weist darauf hin, dass zu kleine und zu feine Buchstaben, wie beispielsweise die einer kleinen Serifenschrift (→ **Serifen**), aufgrund der geringeren Auflösung am Bildschirm nicht mehr lesbar sind. TrueType-Schriften sind generell vorzuziehen, da sie **hinting** Verfahren einsetzen, die Buchstaben möglichst optimal in die Bildschirmmatrix konvertieren (Abbildung 23) (vgl. TURTSCHI R. 1998, S.312/313). Serifenlose ("sans-serif") oder Blockschriften eignen sich besser (beispielsweise Verdana und Tahoma), diese sind ab einer Schriftgröße von 6-8 Punkt am Bildschirm lesbar (ARLETH M. 1999, S.853).

Hinweis: Die Punktgröße einer Schrift ist nicht gleichzusetzen mit der Pixelgröße, da Punkte je nach Anwendung eine feste Größe besitzen; für SVG beispielsweise entspricht ein Punkt 1,25 Pixeln (vgl. FERRAILOLO J. et al 2001, 7.2 Units). Daher wären mit SVG Schriftgrößen ab 8 Punkt möglich, je nach Schriftart.

Abb. 23: Möglichkeiten der Schriftdarstellung an einem Bildschirm



Quelle: veränderte Darstellung aus TURTSCHI R. 1998, S.313.

Sobald Schriften nicht mehr auf einer horizontalen Grundlinie verlaufen - beispielsweise die Beschriftung von Flüssen - werden sie unleserlicher (vgl. ALRETH M. 1999, S.853). Begrenzte Abhilfe schafft dabei das sogenannte **Antialiasing** (vgl. WORM van den J. 2001, S.102/103). Im nächsten Kapitel wird detaillierter auf derartige Bildstörungen eingegangen.

Hinweis: Bei der Gestaltung von Bildschirmkarten muss zudem darauf geachtet werden, dass nicht jede Schriftart in das jeweilige Betriebssystem des Anwenders eingebunden sein muss. Daher empfiehlt es sich, nur allgemein übliche Schriften zu verwenden, und falls dies nicht möglich ist, notfalls die Schriftdefinition in die Anwendung zu implementieren, das aber eine Vergrößerung der Datenmenge nach sich zieht.

Für graphische Objekte, wie beispielsweise punktbezogene Signaturen gilt ähnliches. Auf zu detaillierte/dünne Linien, wie sie in der traditionellen Kartographie eingesetzt werden, gilt es zu verzichten. Damit Signaturen nicht zu dominant wirken, sollten sie die Größe 16 x 16 Pixel nicht überschreiten. Abhilfe schaffen hier Metasymbole, die mit einem Ereignis verbunden sind, beispielsweise einem Hyperlink (vgl. ARLETH M. 1999, S.853).

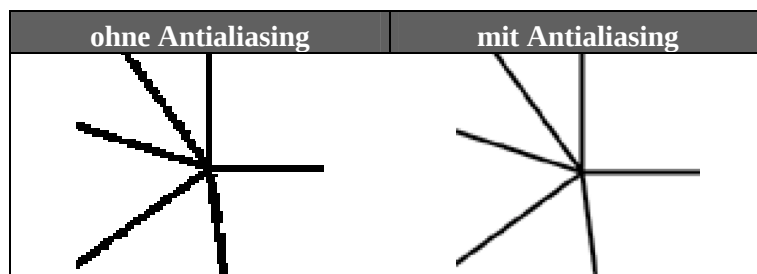
Abschließend noch Aspekte zu graphischen Mindestgrößen von Kartenelementen. Es ist schwierig, eine allgemeine Größenangabe aufgrund divergierender Hard- und Softwarestrukturen und auftretender Bildstörungen zu machen. Ein Punkt sollte jedoch nicht kleiner als 4 Pixel und eine Linie nicht schmäler als 3 Pixel sein. Wichtig ist auch der minimalste Abstand zwischen zwei Objekten, der nicht weniger als 3 Pixel betragen darf. Die Differenzierbarkeit beispielsweise von in der Breite variierenden Linien, ist, im Vergleich zu Papierkarten wesentlich geringer. Durch Kontrastierung können graphische Objekte besser differenziert werden (vgl. DICKMANN F. 2001, S.161).

Ausführlichere Gedanken über die Eigenschaften von graphischen Objekten sind bei BRÜHLMEIER und RÄBER & JENNY zu finden (vgl. BRÜHLMEIER T. 2000, S.23-24 und RÄBER S. & JENNY B. 2001, S.9-12). Über die Gestaltung von Icons in kartographischen Benutzeroberflächen spricht ANGÜSSER (vgl. ANGÜSSER S. 2002).

2.3.1.4 Auftretende Bildstörungen

Bildstörungen wirken sich negativ auf eine Karte aus. Sie werden vom Anwender als unruhig empfunden. Bildstörungen treten bei der Umrechnung der Graphikgeometrie in die Bildschirmmatrix auf. Vergleichbar ist dieser Prozess ungefähr mit dem einer Rasterung von Vektorgraphiken. Diese sinnvollerweise hardwareseitig ausgeführte Transformation ist von entscheidender Bedeutung (vgl. BRUNNER K. 2001, S.236-237). Wichtigste dabei auftretende Bildstörung ist der Alias-Effekt (*engl. Aliasing*). Dieser macht sich als Stufenbildung bei schrägen Graphikelementen bemerkbar. Gegenwärtig werden Störungen dieser Art durch die "Antialiasing"-Funktion der Software behoben. Diese ist zwar rechenaufwendig und bringt somit längere Renderzeiten mit sich, jedoch zahlt sich das, wie Abbildung 24 zeigt, aus (vgl. FERINGA W. 2001, S.203/204).

Abb. 24: Linien ohne und mit Antialiasing



Quelle: Eigene Darstellung.

TURTSCHI weist darauf hin, dass Antialiasing bei kleinen Objekten Unschärfe mit sich bringt. Dieser Effekt ist besonders bei kleinen Buchstaben zu erkennen, aber aufgrund der anderen Vorzüge von Antialiasing zu vernachlässigen (TURTSCHI R. 1998, S.317).

Eine andere Bildstörung resultiert aus der geometrischen Form der Pixel. Dies werden manchmal nicht in quadratischer Form (aufgrund ungeeigneter Graphikeinstellungen), sondern in Rechtecksform, dargestellt. Es kommt dadurch zu Verzerrungen, die beispielsweise Kreise ellipsenförmig erscheinen lassen.

Ein anderer Aspekt betrifft die Augenposition vor dem Bildschirm. Der Blickwinkel ist hier der entscheidende Faktor, wobei sowohl CRT-Monitore als auch LCD-Displays Fehldarstellungen produzieren. Man kann nicht davon ausgehen, dass jeder Anwender optimal vor dem Monitor sitzt. Dies ist besonders der Fall, wenn sich mehrere Personen gleichzeitig Einblick verschaffen wollen.

2.3.2 "Neue" Aspekte zur Kartengestaltung

Wie schon in den vorherigen Kapiteln erwähnt, können Bildschirmkarten die "altehrwürdigen" kartographischen Gestaltungsmöglichkeiten nicht voll ausschöpfen. Das Computerumfeld aber bildet den Rahmen für viele "neue" Darstellungsformen (vgl. WORM van den J. 2001, S.87). Dabei wird in "inneres" und "äußeres" Kartendesign unterteilt. Das innere Design beschreibt die Gestaltung der Kartengraphik selbst, das Äußere dagegen die Funktionalität der Karte (vgl. ARLETH M. 1999, S.850). Für beide gibt es viele sinnvolle neue Gedanken, die sich auf beide auswirken können. Im Folgenden werden diese im Hinblick auf die kartographische Nutzung beleuchtet.

2.3.2.1 Inneres Kartendesign

Das innere Kartendesign beschreibt den Inhalt einer Karte. Zum einen sind hier rein graphische Aspekte zu nennen, zum anderen interaktive Elemente, die sich auf den Inhalt einer Karte beziehen.

2.3.2.1.1 Filter

Filter die Schatteneffekte erzeugen, dienen zur Harmonisierung des Kartenbildes, indem beispielsweise Tortendiagramme mit einem Filter hinterlegt werden (vgl. WORM van den J. 2001, S.103). Dabei ist zu beachten, dass großflächige Filter sehr rechenaufwendig sind. Es ist daher ratsam, Filter gezielt und sparsam einzusetzen.

2.3.2.1.2 Farbverläufe

Farbverläufe repräsentieren den Übergang von einer Ziel- zu einer Quellfarbe. Dabei kann dieser in bestimmter Art und Weise definiert sein (z.B. linear) und verschiedene Farben durchlaufen. Farbverläufe dienen hauptsächlich zur visuellen Gestaltung von graphischen Objekten, wie beispielsweise Signaturen oder Buttons.

2.3.2.1.3 Füllmuster

Bei der Gestaltung von Füllmustern kann aufgrund der primitiven Realisierung am Computer auf umfangreichere Möglichkeiten zurückgegriffen werden. Altertümliche Texturen können im passenden Kontext beispielsweise verbildlicht werden und tragen somit zur einfacheren Zuordnung bei. Das führt letztendlich zu einer "lebendigeren" und interessanteren Karte (vgl. WORM van den J. 2001, S.98). Schraffuren sind aufgrund der unter 2.3.1.4 erwähnten Bildstörungen nicht geeignet.

2.3.2.1.4 Transparenz

Transparenz ermöglicht es, in der traditionellen Kartographie problematische Überlagerungen einfacher und besser darzustellen. Verteilungsgebiete können eingefärbt und zusätzlich mit beschränkter Transparenz versehen werden. Sie verdecken damit nicht die Kartengrundlage und es ist daher bessere Orientierung gegeben. Weiter drängt Transparenz die Basiskarte, vor allem wenn diese aus herkömmlichen Rasterdaten besteht, weiter in den Hintergrund (vgl. ARLETH M. 1999, S.852). Jedoch sollte auch mit Transparenz sparsam umgegangen werden, da oftmals rechenaufwendige Prozesse damit verbunden sind.

2.3.2.1.5 Animation

Mit Hilfe von Animationstechniken können zeitliche Abläufe/Veränderungen und Bewegungen im Raum besser umgesetzt werden. Hier ist noch sehr viel Forschung notwendig, um dem Anwender möglichst akzeptable Darstellungen zu präsentieren. KRAAK bietet dazu einen weitreichenden Einblick (vgl. KRAAK M.-J. 1999, S.173-180), der jedoch den Rahmen sprengen würde.

Zusätzlich kann Animation zur Hervorhebung von Karteninhalten eingesetzt werden. Bestes Beispiel ist nach einer Ortssuche die zugehörige blinkende Signatur in der Karte.

2.3.2.1.6 Interaktive Kartenelemente

Falls Kartenelemente, wie beispielsweise Signaturen, mit Funktionalität ausgestattet werden, stellen diese in gewisser Weise "Kartenbedienelemente" (*engl. map controls*) dar. Diese sind, wie im Screendesign üblich, auch entsprechend zu kennzeichnen (z.B. Verändern des Mauszeigers beim Überfahren). Diese Möglichkeit interaktiver Elemente stellt sich bisher als eine äußerst sinnvolle Erweiterung bei Bildschirmdarstellungen heraus, da sie fehlende Informationen trotzdem ermöglichen. Außerdem ist diese Art der Kommunikation sehr beliebt und steigert das Verständnis einer Bildschirmdarstellung, da Inhalte durch den spielerischen Umgang besser wahrgenommen werden. **Controls** sind allgemein abhängig von den definierten Funktionalitäten des äußeren Kartendesigns.

2.3.2.2 Äußeres Kartendesign

Das äußere Kartendesign beschreibt überwiegend interaktive Vorgänge, enthält aber auch inhaltliche Gedanken einer Karte. "Kartographische" **GUI** – Elemente sind wichtiger Bestandteil für Webmapping GUI's (vgl. Kapitel 2.3.3).

2.3.2.2.1 Weiterführende Information zu einer Darstellung

Wie im vorherigen Kapitel schon angesprochen können Kartenelemente zu Controls mutieren (vgl. Kapitel 2.3.2.1.6). Dabei ist es wichtig, die Art und Weise der Funktionalität im äußeren Kartendesign zu definieren. Folgende Möglichkeiten der zusätzlichen Informationsvermittlung sind üblich:

- **Text/Tabellen:**
Text und Tabellen dienen als weiterführende Erklärungen zum Thema oder auch zu verwandten Themen. Diese können durch Pop-Up Fenster, Hyperlinks oder Tool Tips ermöglicht werden. Sie sind besonders hilfreich und bei Computeranwendungen allgemein gebräuchlich.
- **Graphiken:**
Von Kartenbedienelementen ausgehend, werden zusätzliche Informationen in Diagrammform beispielsweise beim Überfahren mit dem Mauszeiger (*engl. mouseover*) eines Filialstandortes neben der Karte angezeigt.
- **Ton und Film:**
Die Einbindung von Ton und Film bringt ergänzende Information. Die zusätzliche Information kann man als "realistischer" bezeichnen. Oftmals unterschätzt wird die akustische Ausgabe. Ton und Film gestalten eine Anwendung "anspruchsvoller".

2.3.2.2.2 Räumliche Navigation in thematische Karten

Bezüglich der räumlichen Navigation stellt sich die Frage "welche intuitiven Möglichkeiten gibt es zum Navigieren in einer Bildschirmkarte?". Dieser Funktionalität sollte erhöhte Aufmerksamkeit geschenkt werden, da sie einerseits sehr beliebt ist. Andererseits verlangt sie dem Anwender hohe Orientierungsfähigkeiten ab. Dabei muss er durch die Anwendung, beispielsweise in Form einer Übersichtskarte, unterstützt werden. Verschiedene Ansätze, die in gewisser Weise immer mit dem Zweck und Anwenderkreis der Applikation verbunden sind, greifen auf folgende Grundfunktionalitäten zurück:

- **Zoom:** bezeichnet das Vergrößern und Verkleinern des Kartenausschnitts. Dies kann durch verschiedene Möglichkeiten realisiert werden, wie beispielsweise Zoombuttons neben der

Darstellung, die beim Anklicken die gewünschte Funktion ausführen. Weiter sind Zoomfunktionen mit der Maus beziehungsweise Tastatur, wie Vergrößern auf ein aufgezoogenes Rechteck oder automatisches Zoomen beim "Betreten" bestimmter Flächen gebräuchlich. Zudem gibt es meist eine Option, die schnell zur Originalansicht oder zur letzten Zoomstufe zurückführt. Außerdem finden in einigen Vieweranwendungen auch **Slider** zum stufenlosen Zoomen Einsatz.

- **Verschieben** (engl. Fachbegriff pan) bezeichnet das Verschieben des sichtbaren Kartenausschnittes. Lösungen bestehen aus Verschiebebuttons/-balken an den Kartenrändern in die jeweilige Richtung, die beispielsweise nur beim Überfahren mit der Maus sichtbar werden. Weiter kann auch ein externes Bedienelement, ähnlich den Cursortasten, zum Verschieben der Karte dienen. Mausfunktionalität dient zudem zum Verschieben der Darstellung, wie Zentrieren zum angeklickten Punkt oder in Verbindung mit Tastenfunktionalität. Eine weitere Möglichkeit, das Kartenfenster zu verschieben ist mittels einer interaktiven Übersichtskarte. Die üblichen "**Scrollbars**" eines Programmfensters werden bei Bildschirmkarten seltener eingesetzt.
- **Drill-Down:** "**Drill-Down** Mapping" beschreibt den Vorgang, der nach der Auswahl eines Objektes, beispielsweise durch Anklicken einer Region, die zugehörige Raumeinheit detailliert und den Kartenausschnitt vergrößert (vgl. MAPINFO Corp. 2002, S.152-162). Wird ein Drill-Down angeboten, sollte auch an einen **Roll-Up** gedacht werden, der das Gegenteil bewirkt. Diese Art der räumlichen Navigation hat im Business Mapping in Kombination mit dem Daten Drill-Down einen hohen Stellenwert (vgl. Abbildung 53).
- **Übersichtskarte:** Sie wird in passiver Form, nur zum Anzeigen einer Übersicht, oder auch als interaktives Bedienelement, beispielsweise zum Verschieben des Kartenausschnittrechtecks mit der Maus, realisiert.

2.3.2.2.3 Layerkontrolle

Eine Layerkontrolle regelt die Ebenenverwaltung einer Karte. Layer sind in den Informationsschichten einer Karte auszumachen, können aber auch zusätzliche Informationsebenen, wie beispielsweise Luftbilder, enthalten. Das Layermodell erfordert neues kartographisches Denken, beispielsweise kann durch das Ein- und Ausblenden von Kartenebenen schnell zwischen thematischen Inhalten gewechselt werden. Auch das individuelle Anfügen von topographischen Inhalten in die Basiskarte ist dadurch möglich. Die "Funktion" Layerkontrolle kann mit verschiedenen GUI Elementen realisiert werden, z.B. mit Menüs oder "**Checkboxes**".

2.3.2.2.4 Suchdialoge

Suchdialoge/-masken sind meist über Texteingabefelder realisiert und dienen dazu, beispielsweise Ortsnamen in einer Datenbank zu suchen. Die Kartenansicht sollte dann automatisch auf das gefundene Objekt, falls dieses eindeutig ist, zoomen.

2.3.2.2.5 Interaktive Kartenfenster

Auch das Kartenfenster kann Interaktivität erhalten und beispielsweise in der Größe veränderbar sein. Der Kartennutzer ist somit nicht ganz vom Standardlayout des Viewers abhängig, und kann bei Bedarf die Größe des Kartenfensters anpassen. Die Kartenansicht, sowie andere GUI-Elemente sollten sich dementsprechend anpassen.

2.3.2.2.6 Weiteres Gestaltungselemente

- **Animation/dynamischer Inhalt:** Animation/dynamischer Inhalt wird häufig zur Stilisierung eingesetzt. Beispielsweise Animation zum "angenehmeren Aufklappen" von **Drop-Down**

Listen dienen. Jedoch sollte auf übertriebenen Einsatz verzichtet werden, da dieser vom eigentlichen Thema ablenkt (vgl. DICKMANN F. 2001, S.164).

- **Schieberegler** (*engl.* Slider) dienen beispielsweise zur stufenlosen Skalierung von Kreisdiagrammen.
- **Ton und Akustik:** Dazu unterstützen Töne die Attraktivität eines Softwareproduktes. Sie kann besonders bei voll ausgeschöpften graphischen Möglichkeiten hilfreich sein. Beispielsweise findet Sprachausgabe bei Navigationssystemen oder zur Unterstützung Sehbehinderter schon lange Zeit erfolgreichen Einsatz.
- **Druckansicht/-funktion:** Eine Druckansicht ermöglicht, die Darstellung auf sinnvolle Art und Weise auszudrucken. Da ein Ausdruck der reinen Bildschirmansicht oftmals nicht ausreicht, muss erweiterte Druckfunktionalität vorhanden sein. Der Bildschirmansicht nicht enthaltene erklärende Elemente dürfen im Druck nicht fehlen.

2.3.3 Aufbau eines GUI für thematische Karten

2.3.3.1 Der Begriff graphische Benutzeroberfläche

Eine graphische Benutzeroberfläche (*engl.* Graphical User Interface; → **GUI**) ermöglicht es dem Nutzer, mit einem Computer in verschiedener Art und Weise zu kommunizieren. In großem Maße richtet sich diese nach den Erfahrungen und Untersuchungen des Human-Computer Interface (**HCI**) Designs, der Wissenschaft, die Interaktion zwischen Mensch und Computer untersucht. Wichtig ist hierbei, welche Ein- und Ausgabemöglichkeiten einem System zur Verfügung stehen, beziehungsweise welche graphischen Bedienelemente für die Kommunikation üblich sind (vgl. IBM EASE OF USE, Design Concepts). Der im Rahmen dieser Arbeit erstellte Prototyp eines kartographischen Viewers, ist nichts anderes als der Teil einer GUI, die sich hauptsächlich auf die dargestellte Karte bezieht. ANGÜSSER bezeichnet diesen als kartographische Benutzeroberfläche und liefert dazu folgende Definition. Eine kartographische Benutzeroberfläche ist eine "**Schnittstelle zwischen Benutzer und Präsentationen realer oder virtueller, räumlich (raum-zeitlich) oder quasi-räumlich (quasiraum-zeitlich) organisierter Informationen auf der Basis von interaktiven Funktionen (zooming, panning,)**" (ANGÜSSER S. 2002, Folie 6).

MENG beschreibt Anforderungen an ein gutes User Interface, das nutzer- und kundenorientiert sein sollte. Die verfügbaren interaktiven Funktionalitäten müssen deswegen dem Nutzerverhalten angepasst werden. Das Interesse des Anwenders soll dabei geweckt werden. Außerdem muss die Schnittstelle schnell erlernbar sein und der persönliche Handlungsbedarf effektiv unterstützt werden. Das ideale User Interface soll das erwartete Nutzerverhalten berücksichtigen können, dabei spielen Eigenschaften wie uneingeschränkte Funktionen, Konzentration auf das Wichtigste, Abschätzbarkeit von Wartezeiten (beispielsweise durch Statusbalken), eine wichtige Rolle (vgl. MENG L. 2002, S.14/15). IBM gibt in seinen "Design Principles" weitere Hinweise für die Gestaltung eines User Interfaces (vgl. IBM EASE OF USE, Design Basics). Zusammenfassend ist zu sagen, dass Anwenderfreundlichkeit ein entscheidender Faktor für den Erfolg einer Anwendung ist.

Obwohl es sich bei dem erarbeiteten Prototypen um eine Webapplikation handelt, ist die von NIELSEN beschriebene Eigenschaft "kurzlebig" nicht gültig (vgl. NIELSEN J. 2002). Der Viewer stellt dagegen eine Softwarekomponente dar. Mit dieser wird täglich gearbeitet. Die meisten Webapplikationen dienen oftmals nur zum einmaligen Gebrauch.

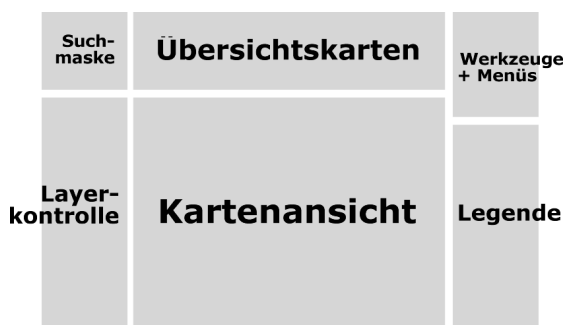
2.3.3.2 Bestandteile eines kartographischen Viewers

Um eine Benutzeroberfläche zu erstellen, müssen zuerst die notwendigen Inhalte definiert werden. Für einen kartographischen Viewer sind diese primär Karten-, Legenden- und Übersichtskartenfenster notwendig. Außerdem werden Navigationselemente und eine Layerkontrolle angeboten (vgl. DICKMANN F. 2001, S.46). Zu diesen Grundelementen kommen, je nach Funktionalität noch weitere Elemente, wie beispielsweise "Textfelder". Sie dienen zum Anzeigen zusätzlicher Informationen. Drop-Down Listen, andere Bedienelemente und Statusanzeigen runden den Umfang einer Benutzeroberfläche ab. Visuelle Akzente setzt man nach der in Abschnitt 2.2.6 beschriebenen visuellen Hierarchie.

2.3.3.3 Gesamtlayout der Benutzeroberfläche

Das Layout der Benutzeroberfläche trägt wesentlich dazu bei, den ersten Eindruck der Applikation zu prägen. Ist dieser schlecht, hat der Benutzer von Anfang an Probleme, sich mit dem Produkt anzufreunden (THISSEN F. 2000, S.162). Bezüglich des Rahmens einer Anwendung kann man behaupten, dass sich alles auf die üblichen Bildschirmrelation 4:3 bezieht. Da der Designer nicht genau weiß, welche absoluten Abmessungen der Monitor hat, werden alle Größenangaben darin relativ angegeben und die Elemente beispielsweise an einem Raster ausgerichtet (vgl. TURTSCHI R. 1998, S.324-325). Dabei müssen grundlegende Screendesignempfehlungen berücksichtigt werden, wie beispielsweise die Platzierung der wichtigsten Inhalte links oben auf dem Bildschirm. Ein Layoutvorschlag allgemein gebräuchlicher kartographischer Benutzeroberflächen ist in Abbildung 25 zu sehen.

Abb. 25: Layoutvorschlag für einen kartographischen Viewers



Quelle: Eigene Darstellung.

Die Abbildung soll die Oberfläche eines kartographischen Viewers nur schematisch wiedergeben. Wichtig ist, dass kartographische Benutzeroberflächen selten alleine stehen, sondern meist in andere Anwendungen integriert werden. Deswegen kann der Layoutvorschlag nur als Teil einer Benutzeroberfläche gesehen werden.

OLBRICH et al weisen darauf hin, dass es auf keinen Fall Sinn und Zweck sein kann, alle Bestandteile der kartographischen Benutzeroberfläche mit einem Extrarahmen zu versehen (vgl. OLBRICH G. & QUICK M. & SCHWEIKART J. 2002, S. 90).

3 Technologien

Damit Webmapping für Geomarketing betrieben werden kann, ist es besonders wichtig, die gegenwärtigen technologischen Möglichkeiten zu untersuchen. Zum einen spielt die Eignung des Ausgabeformates für kartographische Zwecke eine wichtige Rolle. Mit der spezifizierten XML Anwendung SVG ist eine optimale Darstellungsform geschaffen worden. Zum anderen muss einfache Integrierbarkeit in entscheidungsunterstützende Systeme gegeben sein. Die Vorteile des "Webstandards" XML ermöglichen diese. Die folgenden Abschnitte beschreiben die notwendigen Technologien und untersuchen diese auf deren Eignung.

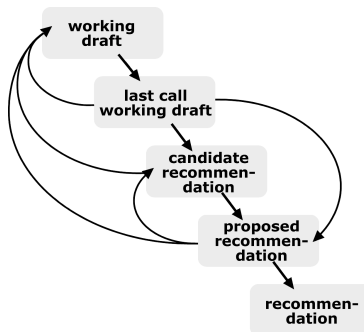
3.1 Webtechnologien

3.1.1 Vorteil standardisierter Technologien

Jeder, der professionell Software einsetzt, weiß wie schwierig es ist, eine optimale Lösung für die jeweiligen Bedürfnisse zu finden. Setzt man heute auf das proprietäre System eines Herstellers, so kann diese Entscheidung morgen schon fatale Folgen haben. Zum Beispiel wenn technologische Veränderungen stattfinden, die das Altsystem unbrauchbar machen (vgl. NITSCHKE M. 1998, S.124). **Modularität**, Erweiterbarkeit, Portabilität und Integrationsfähigkeit sind wichtige Faktoren, dem entgegen zu wirken. Dabei müssen die in Kapitel 2.1.6 behandelten Systemkomponenten einer Geomarketinganwendung sinnvoll vereinigt werden. Immer mehr Kunden zwingen mit diesen Anforderungen Systemhersteller, einheitliche Kommunikationsformen zwischen einzelnen Systemkomponenten zu finden. In der Praxis haben sich oftmals proprietäre Lösungen eines Herstellers durchgesetzt. Diese Tatsache führte zu großer Abhängigkeit von einem Hersteller. Dadurch wurde die weitere Entwicklung durch erhöhte Preise und Qualitätsminderung negativ beeinflusst.

Das offene Protokoll **HTTP** und die "Websprache" **HTML** haben mit der rasanten browserbasierten Nutzung des Internets gezeigt, welches Potential in herstellerneutralen Spezifikationen stecken kann (vgl. HAZAËL-MASSIEUX D. 2002). Um den eigenmächtigen Veränderungen konkurrierender Browserhersteller entgegenzuwirken, wurde 1994 das World Wide Web Consortium (kurz W3C) am **MIT** in Boston gegründet. Unter der Beteiligung einer zunehmenden Zahl renommierter Forschungseinrichtungen, wichtiger Software Hersteller beziehungsweise IT Firmen verabschiedet das Consortium von Jahr zu Jahr zahlreiche als Empfehlung geltende Spezifikationen (*engl. Recommendation*). Diese kann man deswegen als "herstellerneutral" bezeichnen. Spätestens seit der Übernahme der XML Spezifikation ist das W3C neben den offiziellen Nationalen und Internationalen Standardisierungsorganisationen (z.B. **DIN** und **ISO**) zu einem wichtigen Gremium aufgestiegen. Im Gegensatz zu diesen offiziellen Ausschüssen, bietet das W3C Spezifikationen kostenfrei an. Zudem reagiert es flexibler auf technische Veränderungen. Dies wird durch die Bildung von sogenannten Arbeitsgruppen erreicht, die eine Technologie, wie beispielsweise XML, bearbeiten. Um ein harmonisches Nebeneinander zu erreichen, arbeiten diese Fachkreise auch miteinander. Die verschiedenen XML Anwendungen zeigen diese Tatsache. Bevor Empfehlungen endgültig verabschiedet werden, muss ein breiter Konsens der Beteiligten vorliegen. Durch den definierten Prozess der Modifizierung und Anerkennung wird Übereinstimmung erreicht (vgl. Abbildung 26) (vgl. FISCHER H. 2000 und FISCHER H. & BIRKENBIHL K. 2002):

Abb. 26: Recommendation Process



Quelle: Eigene Darstellung in Anlehnung an JACOBS I. 2001, S.31.

Der in Abbildung 26 dargestellte Prozess (*engl.* Recommendation Process/Track) ist aufgrund der großen Anzahl von Beteiligten oft langwierig und aufwendig. Jedoch muss beachtet werden, dass für den Einsatz praktikabler Lösungen einige Zeit der Entwicklung beansprucht wird. Denn obwohl beispielsweise XML schon vielfältig im Einsatz ist, muss man die gewonnenen Erfahrungen erst verarbeiten. Es wird also noch einige Zeit vergehen, bis sich die "XML Welt" vollständig etabliert hat.

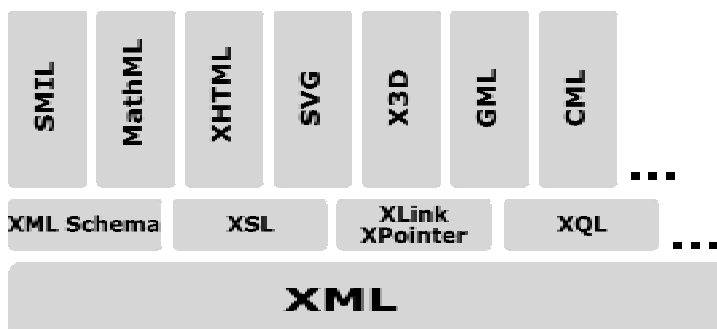
Am Besten ist diese Tatsache an der clientseitigen Umsetzung der Empfehlungen zu erkennen, die von Jahr zu Jahr voranschreitet. Kein Browser bietet gegenwärtig eine zufriedenstellende Umsetzung der "Standardtheorie". Oft sind nur Teile davon implementiert. Selbst schon lang spezifizierte Standards, wie Cascading Style Sheets (vgl. Kapitel 3.1.2.4), werden in der Praxis noch nicht vollständig unterstützt.

Trotzdem lassen Vorgehensweisen verschiedener Softwareunternehmen erahnen, dass W3C Spezifikationen immer erfolgreicher sind.

3.1.2 XML und seine Anwendungen

Sie Extensible Markup Language (XML) ist keineswegs nur ein weiterer "Webstandard", sondern vielmehr eine Familie von Techniken (vgl. FISCHER H. 2002). Das W3C hat mit XML eine Grundlage geschaffen, die Erweiterungen für fast jedes Bedürfnis zulässt. Eine Übersicht der entstandenen XML Welt ist in Abbildung 27 zu sehen.

Abb. 27: Übersicht der XML Welt



Quelle: Eigene Darstellung nach NEUMANN A. & WINTER A. 2001.

Wie unschwer zu erkennen ist, bildet XML die Basis für weitreichende Auszeichnungssprachen, sogenannte XML Anwendungen. Die nächsten Kapitel sollen Übersicht und Verständnis der bestehenden XML Technologien geben.

3.1.2.1 Extensible Markup Language

Die Extensible Markup Language (XML) definiert Regeln einer Auszeichnungssprache (*engl.* Markup Language) und ist daher eine Art "Meta-Auszeichnungssprache" im Textformat (vgl. HAROLD E. 2002, S.31). XML gilt zudem als webfreundliche Vereinfachung des ISO Standards **SGML**; der Erfolg von SGML als Datenbeschreibungssprache in der Druckbranche ist der Hauptgrund für die Entwicklung von XML. Ein XML Dokument besteht, wie bei allen Auszeichnungssprachen, aus "**Tags**", die in semantischer Weise Daten beschreiben. Dabei sei ausdrücklich erwähnt, dass XML im Gegensatz zu HTML nur Inhalte beschreibt. Für die Formatierung beziehungsweise die graphische Ausprägung sind andere Sprachen notwendig. Diese sind am geeignetsten ebenfalls XML basiert, müssen dies aber nicht sein. Abbildung 28 zeigt das vereinfachte XML Konzept. Inhalt, Struktur und graphische Beschreibung ergeben eine Darstellung.

Abb. 28: Vereinfachtes XML Konzept



Quelle: Eigene Darstellung nach SAMLAND W. & HINTERMEIER A. & DÜRR J. 2000, S.4.

Ein XML Element (auch Knoten oder *engl.* Node genannt) besteht aus einem "Anfangstag" und einem "Endtag", die den Inhalt (*engl.* nodeValue) eines Elementes einschließen. Die freie Deklarierbarkeit von Elementnamen hat den Vorteil, nicht nur vordefinierte Elemente verwenden zu müssen. Weiter können einem Element verschiedene Attribute zugewiesen werden, die Namen dieser sind auch frei wählbar.

Abb. 29: Beispiel eines XML Elementes

```
<textFeld sprache="de" name="definition">Das ist eine Definition!</textFeld>
```

Quelle: Eigene Darstellung.

Hinweis: Im Beispiel ist textFeld der Elementname mit den Attributen sprache und name, die Zeichenkette "Das ist eine Definition!" ist der Inhalt des Elementes.

XML gibt damit verschiedenen Nutzergruppen die Möglichkeit, eigene Auszeichnungssprachen zu entwickeln. Zum Beispiel die Geographic Markup Language (**GML**), die eine Syntax zur Beschreibung geographischer Vektordaten zur Verfügung stellt (vgl. Kapitel 3.1.2.6). Diese können dann beispielsweise dem effektiven Datenaustausch dienen und durch andere XML kompatible Sprachen zu XML Visualisierungsanwendungen wie SVG oder **X3D** transformiert werden.

Die Document Type Definition (kurz **DTD**) legt die Struktur eines XML Dokumentes fest. Sie ist eine Sammlung von Konstrukten, die der XML Datei eine baumartige Form geben beziehungsweise deren Entitäten (*engl.* **Entity**) deklariert. Eine DTD kann, muss aber nicht für eine **wohlgeformte** XML Datei vorhanden sein. Für eine **gültige** XML Datei dagegen schon. In Zukunft wird DTD von **XML Schema** abgelöst werden, die wiederum einen weiteren XML **Namespace** darstellt. Dieser lässt weitaus mehr Struktureigenschaften definieren, wie beispielsweise die Reihenfolge und Anzahl der Elemente (vgl. SPEERBERG-MCQUEEN C. M. & THOMPSON H. 2002).

Oftmals tauchen die Begriffe "XML Anwendung", "XML basiert/konform" oder "XML Derivat" auf, die für einen Namensraum stehen, der nach der XML Spezifikation erstellt worden ist.

Abb. 30: Beispiel eines XML Dokumentes

```
<?xml version="1.0"?>

<!DOCTYPE firma SYSTEM "firma.dtd">

<firma>
  <mitarbeiter>
    <vorname>hans</vorname>
    <nachname>dampf</nachname>
  </mitarbeiter>
</firma>
```

Quelle: Eigene Darstellung.

Prolog: XML Deklaration mit Versionsangabe

Struktur: gibt dem Dokument eine Struktur; hier in Form eines Dokumententyps

Inhalt: XML Fragmente, eingeschlossen in des Dokumentenfragment (hier "firma").

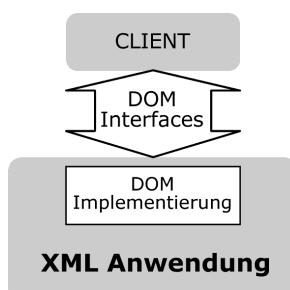
Vorteile von XML und seinen Anwendungen liegen auf der Hand. Es wird zum Einen eine einzigartige Schnittstelle geschaffen, um bestehende Anwendungen (z.B. verschiedenen GIS Architekturen) auf einfache Art und Weise kommunizieren zu lassen. Zum Anderen bewirkt ein harmonisches Zusammenspiel der verschiedenen XML Anwendungen, simples Generieren des Inhaltes. Wichtig dafür ist die volle Browserunterstützung.

Größter Nachteil von XML Anwendungen ist die beachtliche Datenmenge, da XML textbasiert vorliegt. **Binäre** Daten sind schlanker, jedoch nur von speziellen Anwendungen lesbar. Die weite Verbreitung von Komprimierungsverfahren, wie beispielsweise **gzip** in Applikationen und Serverumgebungen lassen eine schnellere Übertragung zu (vgl. FISCHER H. 2002 und BRAY T. & PAOLI J. & SPERBERG-MCQUEEN C. & MALER E. 2000).

3.1.2.2 Document Object Model

Das Document Object Model (DOM) ist ein baumartiges Objektmodell, das Dokumente als Objekte darstellt. XML und Anwendungen basieren auf einer DOM Struktur. Das DOM stellt weiter eine standardisierte Schnittstelle, das DOM Interface, mit spezifizierten Methoden zur Verfügung. Dieses kann daher als **API** angesehen werden. Alle Typen von Programmiersprachen, unabhängig der Plattform und des Betriebssystems, können auf das DOM zugreifen. DOM Methoden und Eigenschaften sind wichtig, um durch Scripte auf die verschiedenen Elemente zuzugreifen. Attribute und Werte können ausgelesen sowie durch Programmierung verändert oder gelöscht werden. Voraussetzung dafür ist, dass das DOM Interface in der Anwendung implementiert ist (vgl. LE HORS A. et al 2000 und HELFRICH C. 1998). Um Clientseitiges Scripting mit XML Anwendungen zu betreiben ist eine Integration der DOM Schnittstelle zwingend erforderlich (Abbildung 31).

Abb. 31: Funktionsweise des DOM



Quelle: Eigene Darstellung nach HELFRICH C. 1998.

Die Grundidee des DOM's ist sehr wichtig für das Verständnis von XML und seinen Anwendungen. Zur Entstehung des DOM's trug wesentlich die Tatsache bei, ein "Web" zu schaffen, dass von einem Browser dargestellt werden kann, und zwar ohne zusätzliche Plug-In's. Man spricht

in diesem Zusammenhang von "nativer" Browserunterstützung. Daher sollen Dateiformate für Webapplikationen auch auf Basis des DOM's beruhen (vgl. NEUMANN A. & WINTER A. 2001).

3.1.2.3 XHTML / HTML

XHTML stellt die XML basierte Neuformulierung von HTML 4.0 dar, und ist damit eindeutiger spezifiziert. Im Rahmen der verfeinerten Spezifikation entfallen einige Elemente und Attribute, die von HTML 4.0 bekannt sind. Das heißt nicht, dass XHTML eingeschränkt ist, denn durch andere XML Derivate ist es möglich, auf unterschiedliche Art und Weise diese fehlenden Attribute zu ersetzen. Gegenwärtig sind die Namensräume XHTML 1.0 Strict, XHTML 1.0 Transitional und XHTML 1.0 Frameset spezifiziert (vgl. PEMBERTON et al 2002).

3.1.2.4 Cascading Style Sheets

Cascading Style Sheets sind Definitionen, die es ermöglichen, Dokumente zu "stilisieren". Dabei werden "Styles" in Formatvorlagen definiert und XML Elementen zugeordnet. Stilvorgaben können intern, global für ein ganzes Dokument oder auch "tagweise" in internen und/oder externen CSS-Dateien deklariert werden. Cascading Style Sheets vererben sich außerdem durch den Dokumentenbaum. Auch das Anwenden von mehreren Formatvorlagen auf ein Element ist gegeben. Der Funktionsumfang reicht von Farbe über Größe bis zur Seitengestaltung. Die Formatierungsmöglichkeiten entsprechen denen von Textverarbeitungsprogrammen. Cascading Style Sheets sind schon lange in der Webentwicklung bekannt (CSS1 im Dezember 1996 verabschiedet) und werden beispielsweise in HTML Dokumenten verwendet. Sie sind, wie XML, offizieller Internetstandard des W3C, aber keine XML Anwendung. Aktuell wird an CSS level3 gearbeitet (vgl. BOS et al 1998 und BOS B. & FROUMETIN M. & ADLER S. 2002).

3.1.2.5 Extensible Stylesheet Language

Der W3C Standard XSL steht für Extensible Stylesheet Language und stellt eine XML basierte Stil-Sprache dar. Im Gegensatz zu CSS dient XSL aber auch zum Struktur- und Dokumenten Design und ist somit wesentlich vielseitiger (vgl. ADLER et al 2001). Wichtigste Bestandteile von XSL sind XSLT und XSL-FO, im Folgenden kurz erläutert.

3.1.2.5.1 XSLT

XSLT wandelt ein vorhandenes XML Dokument in ein anderes XML Dokument um. Das Ausgabeformat kann auch einem anderem "namespace" angehören. Grundsätzlich ist es möglich XSL Transformationen auf dem Client und/oder dem Server auszuführen, vorausgesetzt, XSLT wird unterstützt. Ein Beispiel, das XML mit XSLT zu SVG transformiert, ist bei IBM zu finden (vgl. IBM DEVELOPER WORKS 2001), wie zukunftssträchtig XSLT ist, zeigt auch Kapitel 3.1.2.6. Wie einfach mit XSLT aus XML eine SVG Graphik erzeugt wird, veranschaulicht das Beispiel im Anhang. Derzeit nehmen komplexere XSL Transformationen, vor allem auf dem Client, noch erheblich Rechenzeit in Anspruch.

3.1.2.5.2 XSL-FO

Für diese Arbeit weniger von Bedeutung ist "XSL Formatting Objects", da es eine sehr junge Technik darstellt und noch wenig Unterstützung erfährt. Es ermöglicht Formatierungen des Layouts auf XML Dokumente anzuwenden, in ähnlicher Art und Weise, wie es CSS tut. Ergebnis ist dann ein entweder eine formatierte XSL Datei, oder ein vollständig "designtes" Dokument, beispielsweise eine SVG Datei. Der wesentliche Vorteil gegenüber CSS ist aber, dass XSL-FO ein vorhandenes XML Dokument direkt in ein anderes Dateiformat umwandeln kann, und somit für

professionelle **Crossmedia** Zwecke bestens geeignet ist (vgl. BRAILSFORD D. 2002). CAGLE et al geben eine umfangreiche Übersicht zu CSS und XSL FO (vgl. CAGLE K. et al 2001, S.311-363). Erste serverseitig erzeugten Ergebnisse liefert **Apache's** FOP Projekt (vgl. APACHE 2002a).

3.1.2.6 Geographic Markup Language

Für den vorliegenden Prototypen direkt nicht von Bedeutung soll GML aber als ein Beispiel für eine XML Fachanwendung dargestellt werden. Durch GML kann vektorbasierte Geoinformation XML konform gespeichert werden. Wichtig ist, dass die vom **OGC** entwickelte herstellerneutrale Spezifikation schon sehr ausgereift vorliegt - die aktuelle Version 2.1.1 verwendet bereits XML Schema - sich in der Praxis jedoch erst noch profilieren muss (vgl. COX et al 2002). Besonders die einfache Art und Weise, GML zu transformieren, findet in der GIS/SVG Entwicklergemeinschaft großen Zuspruch und bringt besonders als Datenaustauschformat Vorteile (vgl. REICHARDT M. 2001, S.18-21). Im Zusammenhang mit GML ist vor allem das britische Ordnance Survey mit ihrem OS Master Map Programm zu erwähnen, das bereits digitale Kartendaten GML spezifiziert anbietet. Zudem wird versucht, die Daten mit XSL Transformationen in SVG zu visualisieren (vgl. ORDNANCE SURVEY 2002 und TRIGLAV J. 2002, S.12–15). Auch der Topographische Dienst der Niederlande mit dem TOP10NL Projekt experimentiert erfolgsversprechend daran, die in GML gehaltene digitale Topographische Karte mit XSLT nach SVG zu transformieren (vgl. TOP10NL 2002). Wie auch diese Beispiele zeigen, liegt es nahe, GML Daten mit XSLT nach SVG zu transformieren, um somit komplizierte Zwischenschritte einzusparen. Selbst kommerzielle Desktop GIS, wie MapInfo Professional 7, unterstützen GML Import (vgl. MAPINFO Corp. 2002b, S.1). Es ist zu erwarten, dass GML sich besonders als allgemeines Austauschformat in der Geoinformatik Branche durchsetzen wird.

3.2 SVG – Scalable Vector Graphics

SVG steht für skalierbare Vektorgraphiken. SVG ist die vom W3C spezifizierte XML Anwendung zur Darstellung von zweidimensionalen Vektorgraphiken. Skalierbar bedeutet einheitliches Vergrößern und Verkleinern einer Darstellung. Dabei ist die Graphik nicht von der Auflösung des Ausgabegerätes abhängig. Eine SVG Darstellung wird deswegen immer in der möglichen Auflösung des Ausgabegerätes angezeigt; unabhängig von den absoluten Ausmaßen der Darstellung. Wie oben erläutert trennt XML Inhalt, Struktur und Darstellung streng voneinander (vgl. Kapitel 3.1.2). Aufgrund dessen, dass noch keine Struktur für die Darstellung von zweidimensionalen Vektorgraphiken vorhanden war, wurde die SVG Spezifikation vom W3C ausgearbeitet. Folgende Abschnitte sollen SVG erklären und die Möglichkeiten dieser XML Anwendung aufzeigen.

3.2.1 Dateiaufbau – Dokumentenstruktur - Elemente

Da SVG eine XML Anwendung ist, können gültige SVG Dateien leicht verstanden werden. Das in Abbildung 28 vorgestellte XML Konzept gilt deswegen auch für SVG Dateien. Die Struktur mit ihren Elementen und Attributen wird in der vom W3C spezifizierten SVG-DTD vorgegeben. Eingebunden wird diese durch die DOCTYPE Deklaration, die aus dem Public Identifier und dem System Identifier für SVG besteht (→ **DTD**). Wie alle XML Elemente besitzt auch SVG genau ein **Root Element**. Dieses muss immer ein <svg> Element sein. Alle weiteren Elemente werden an

dieses angefügt. Man spricht von "Kindern" (*engl.* children). Diese können alle Grundelemente der SVG Spezifikation sein (Abbildung 32).

Abb. 32: Grundelemente der SVG Spezifikation

Grundtypen	SVG Elemente
Containerelemente	<code><svg></code> <code><g></code> <code><a></code> <code><defs></code> <code><symbol></code> <code><pattern></code> <code><clipPath></code> <code><mask></code> <code><marker></code> <code><switch></code> <code><view></code>
Graphikelemente	<code><path></code> <code><text></code> <code><rect></code> <code><circle></code> <code><ellipse></code> <code><line></code> <code><polyline></code> <code><polygon></code> <code><image></code> <code><use></code> , wobei die beiden letzten zur Referenzierung von Graphiken dienen.
Textauszeichnungselemente	<code><text></code> <code><tspan></code> <code><tref></code> <code><textPath></code> <code><altGlyph></code>

Quelle: Eigene Darstellung basierend auf der Ausführung in FERRAILOLO J. et al 2001, Kapitel 1.6.

Ein Containerelement fasst ein oder mehrere Elemente zusammen. Diese können alle Grundelemente der SVG Spezifikation sein. Die Graphikelemente sind dagegen für den graphischen Inhalt einer SVG Darstellung verantwortlich, wobei unterschieden wird in Elemente, die Geometrie definieren und Elemente, die allgemein Graphik referenzieren. Textauszeichnungselemente behandeln die Gestaltung von Text.

Es gibt zusätzlich Elemente, die als graphische Effekte dienen (Abbildung 33).

Abb. 33: SVG Elemente für zusätzliche graphische Effekte

Graphische Effekte		SVG Elemente
statisch	dynamisch	
Verläufe		<code><linearGradient></code> <code><radialGradient></code>
Füllmuster		<code><pattern></code>
Bescheidungspfade		<code><clipPath></code>
Masken		<code><mask></code>
Filter		<code><filter></code>
	Bewegung	<code><animate></code> <code><set></code> <code><animateMotion></code> <code><animateColor></code> <code><animateTransform></code>

Quelle: Eigene Darstellung

Die Anwendung der SVG Elemente für statische Darstellungen erfolgt immer in der gleichen Weise. Sie werden zuerst als Kinder des `<defs>` Elementes deklariert und nachfolgend im zu formatierenden Element instanziiert. Dabei spielt es keine Rolle, ob es sich um graphische oder Textauszeichnungselemente handelt. Beispielsweise füllt das `<linearGradient>` Element eine Fläche mit einem definierten linearem Farbverlauf. Derartige Instanzen werden in XML Anwendungen mit **XLink**, einer weiteren XML basierten Spezifikation für Linkdefinitionen realisiert.

Elemente die Bewegung bewirken, werden dagegen als Kinder des zu animierenden Elementes definiert. Außerdem sollte das `id` Attribut, dass zum eindeutigen Bezeichnen von Elementen dient, Erwähnung finden. Dieses ist auf alle Elemente anwendbar, jedoch muss beachtet werden, dass die XML Spezifikation nur eindeutige Identifizierungsschlüssel zulässt.

Erläuterungen zur Syntax anhand von Beispielen und zusätzliche Informationen sind in der Spezifikation der SVG Arbeitsgruppe des W3C's ausführlich beschrieben (vgl. FERRAILOLO J. et al 2001)

3.2.2 Graphische Möglichkeiten von SVG

Die Möglichkeiten mit SVG Graphiken zu erzeugen sind vielfältig und stehen denen anderer Vektorgraphikformate in nichts nach. Im Folgenden sind die wichtigsten Grundelemente und Darstellungsmöglichkeiten beschrieben.

3.2.2.1 Kartengeometrien

Geometrien einer Karte setzen sich aus den Grundelementen Punkt, Linie und Fläche zusammen. Verschiedene Implementierungsmöglichkeiten werden in den nächsten Kapiteln erläutert.

3.2.2.1.1 Punkte

Ein Punktelement oder –attribut ist in SVG nicht vorgesehen. Jedoch können Kreise/Rechtecke und Signaturen/Symbole mit den x / y Attributen als alternative Punktdarstellung platziert und mit dem `transform` Attribut verschoben, rotiert und skaliert werden.

3.2.2.1.2 Linien

Für linienartige Elemente gibt es ein weites Spektrum an Möglichkeiten. Neben `<line>` und `<polyline>` ist `<path>` das am häufigsten verwendete Element. Der Grund für die Verwendung des `<path>` Elementes ist dessen ausgesprochene Vielfältigkeit. Neben Pfaddaten aus aufeinanderfolgenden X/Y-Koordinaten, stellt die SVG Spezifikation noch weitere "Kommandos" (*engl. commands*) zur Liniendarstellung zur Verfügung. Diese erlauben das Zeichnen von elliptischen Formen sowie kubischen und quadratischen Bezierkurven. Linien werden mit dem `d` Attribut definiert (*engl. pathData*).

Abb. 34: Beispiel eines Pfadelementes als Linie

```
<path d="m50 50 15 0 10 5 1-5 0"/>
```

Quelle: Eigene Darstellung.

Das Kommando `m` springt zu der angegebenen Koordinate ("moveto") ohne zu zeichnen. `l` dagegen zeichnet eine Linie zwischen zwei Punkten ("lineto"). SVG stellt zudem über die Kommandos eine Möglichkeit zur Verfügung, relative oder absolute Koordinatenangaben zu verwenden. Ist ein Kommando klein "geschrieben" gelten die Koordinaten relativ zum Ausgangspunkt, groß dagegen absolut zum Element. Innerhalb einer Pfaddefinition kann immer nur eine der beiden Varianten definiert werden.

3.2.2.1.3 Flächen

Flächen können durch Kreise, Rechtecke oder Ellipsen repräsentiert werden. Weiter stehen ein eigenes `<polygon>` Element zur Verfügung, jedoch wird das `<path>` Element bevorzugt genutzt. Eine Pfaddefinition aus einem `moveto` und `lineto` Kommando entspricht einem `<polygon>` Element mit `points` Attribut. Voraussetzung dafür ist, dass der Pfad sich schließt. Ein Pfadelement, wie in Abbildung 34 zu sehen ist, wird mit einem abschließendem `z` Kommando zu einem Polygon geschlossen (vgl. Abbildung 35).

Abb. 35: Beispiel eines Pfadelementes als Polygon

```
<path d="m50 50 15 0 10 5 1-5 0z"/>
```

Quelle: Eigene Darstellung.

Auch die für GIS Anwendungen sehr wichtigen **multiplen Polygone**, finden in SVG indirekt Unterstützung. Die verschiedenen Bestandteile werden in einer Pfaddefinition beschrieben und

jeweils mit *m* begonnen und durch ein *z* geschlossen. Andere Ausprägungseigenschaften, wie **Doughnuts** können durch das *fill-rule* Attribut erzwungen werden.

Hinweis: Bei der Erstellung komplizierterer Geoobjekte ist trotzdem Vorsicht geboten. SVG ist kein GIS Vektorformat. Es sollte deswegen immer darauf geachtet werden, wie die Daten vorliegen. Ist ein Polygon beispielsweise alleine durch seine Koordinatenfolge definiert, muss es nicht mehr mit einem *z* Kommando geschlossen werden.

3.2.2.2 Text mit SVG

Für Text sieht die Spezifikation das `<text>` Element vor. Der eigentliche Text steht als Zeichenkette im Elementinhalt zwischen dem öffnendem und schließendem Tag. Die *x* und *y* Attribute platzieren den Text an die gewünschte Position. Formatierung erfolgt mit Stylesheets (vgl. nächster Punkt). Zum Beispiel kann die Basislinie eines `<text>` Elementes verschoben werden. Das Attribut *baseline-shift* in der Formatvorlage ermöglicht präzise Verschiebung. Das `<textPath>` Element lässt die für kartographische Zwecke wichtige Textausrichtung entlang eines Pfades zu. Das `<tspan>` Element kann Gestalt und Position jedes einzelnen Buchstaben beeinflussen. Beide werden als Kinder des Textelementes definiert.

Abb. 36: Beispiel eines SVG Textelementes

```
<text x="50" y="50">hello svg world</text>
```

Quelle: Eigene Darstellung.

Die SVG Textimplementierung hat neben den umfangreichen Formatierungsmöglichkeiten den großen Vorteil, dass Textstrings über Abfragen selektiert werden können. Ein sehr großes Manko sind die eingeschränkten Möglichkeiten mit Textblöcken zu arbeiten. Besonders die fehlende Unterstützung von Zeilenumbrüchen (*engl.* text-wrapping) ist zu erwähnen.

Hinweis: Quint bietet einen **Workaround** an, der das Arbeiten mit Textblöcken erleichtert (vgl. QUINT A. 2002b). Erweiterte Unterstützung von Textblöcken ist mit SVG Version 1.2/2 zu erwarten, wie dem gegenwärtigem Working Draft zu entnehmen ist (vgl. JACKSON D. 2002, 4.2).

Für den Einsatz verschiedener Schriftarten ist zu berücksichtigen, dass nur Fonts eingesetzt werden können, die dem Betriebssystem bekannt sind. Daneben gibt es die Möglichkeit, Schriften auch direkt in eine SVG Datei einzubinden. Dies bringt jedoch eine erhebliche Zunahme der Datenmenge mit sich. Damit Umlaute/Sonderzeichen richtig dargestellt werden, muss im XML Prolog, der entsprechende Zeichensatz eingebunden werden. Für West- und Mitteleuropäische Schriften beispielsweise der **"ISO-8859-1"** Standard.

3.2.2.3 Styles zur Gestaltung

Unter Styles versteht man die graphische Ausprägung von Elementen. Styles müssen den Elementen zugewiesen werden, da sonst ungeeignete Standardformatierungen angewendet werden. Für die graphische Formatierung sind CSS, XSL und weitere SVG-interne Stylingeigenschaften in der Spezifikation vorgesehen. Gegenwärtig werden Cascading Style Sheets am häufigsten für die Gestaltung graphischer Elemente eingesetzt. CSS können wie bei HTML über verschiedene Selektoren auf SVG Elemente angewendet werden. In Karten ist die Zuweisung über einen Identifikationsschlüssel üblich. Eine Stylesheetklasse wird mit dem *class* Attribut einem Knoten angefügt.

Styles können aber auch im *style* Attribute eines Elementes, oder den entsprechenden CSS2 Eigenschaften direkt als Attribut deklariert werden. Formatierungsattribute für SVG sind die meisten der CSS2- sowie SVG spezifische Eigenschaften (eine Liste dieser ist in FERRAILOLO J. et al 2001, Chapter 6.1 oder Appendix B.6.4, zu finden).

CSS @media Types erlauben verschiedene "style" Definitionen für unterschiedliche Ausgabetypen. Wird für den Druck und den Bildschirm unterschiedlich formatiert, weichen die beiden Darstellungen voneinander ab. Das hat den Vorteil, dass beispielsweise eine Farbskala am Bildschirm farbig erscheint. Beim Druck wird sie dagegen automatisch in eine echte Grauwertskala umgewandelt.

Allgemein können CSS sowohl intern als auch extern definiert werden (vgl. FIBINGER I. 2002 S.172-182).

Neben CSS werden auch Entitäten mit graphischen Attributen für die Formatierung von SVG Elementen eingesetzt. Zudem sind die zur Gestaltung von Elementen in Abbildung 33 aufgeführten Elemente noch zu erwähnen.

3.2.2.4 Koordinatensystem, Transformation und Kartenansichten

SVG verwendet ein, für die graphische Softwareentwicklung typisches, kartesisches Koordinatensystem mit negativer y-Achse und positiver x-Achse. Der Nullpunkt ist auf dem Bildschirm "links oben". Bei kartesischen Koordinatensystemen des Geoinformationswesens liegt der Ursprung dagegen "links unten". Deswegen müssen kartographische Darstellungen durch das transform Attribut verschoben und skaliert werden. Sphärische Koordinaten finden keine Unterstützung.

SVG unterstützt zusätzlich verschiedene Maßeinheiten. Unterschieden werden absolute Maße, in beispielsweise Inch oder Zentimeter, und relativen Maße, in Pixeln (default Einstellung) oder Prozenten.

Durch das viewBox Attribut oder das <view> Element lassen sich verschiedene Bildansichten erzeugen. Diese Tatsache ist im Hinblick auf kartographische Anwendungen von großer Bedeutung. Räumlich Navigation kann, wie im Prototypen realisiert, durch verschiedene Bildansichten berechnet werden (vgl. 4.1.2.2). Im Zusammenhang mit dem viewBox Attribut ist das preserveAspectRatio Attribut zu erwähnen. Dieses ist muss gesetzt werden, um die Proportionen der Darstellung zu erhalten (vgl. FERRAILOLO J. et al 2001, Chapter 7 und SCHONEFELD P. 2002, Part 1 – Viewing SVG).

3.2.2.5 Interaktion durch Clientseitiges Scripting

Damit in SVG Interaktion möglich wird, ist der Einsatz von clientseitigem Scripting notwendig. Clientseitiges Scripting zur Manipulation des SVG Baumes steht dabei im Vordergrund. So verwendet man Scripting beispielsweise um verschiedene Kartenebenen durch Checkboxes ein- und auszublenden. Meist wird dafür Javascript auf Basis des **ECMA** 262 Standards eingesetzt (vgl. ECMA 1999). Zudem unterstützen alle gängigen Browser das DOM (level 2) Interface mit dessen Methoden und Eigenschaften. Das als Standard anzusehende Adobe SVG Viewer 3 (→ **ASV3**) Plug-In beinhaltet zudem eine eigene Scriptingengine. Diese unterstützt neben ECMA 262 und DOM (level2) weitere Methoden und Eigenschaften, wie die "ECMAScript Language Binding for SVG". Zudem sind eigene Methoden und Eigenschaften implementiert, wie beispielsweise die `getURL()` Methode. Mit deren Hilfe können Anfragen an einen Server gesendet werden. Das Nachladen von Daten wird dadurch ermöglicht. Folgende Referenzen sind für die optimale Programmierung von besonderem Nutzen:

- **ECMA 262 Spezifikation (vgl. ECMA 1999)**
- **ECMAScript Language Binding for DOM2 Core (vgl. LE HORS A. et al 2000, Anhang E)**
- **ECMAScript Language Binding for SVG1 (vgl. FERRAILOLO J. et al 2001, Anhang E)**
- **Current Support for SVG – Adobe SVG Viewer (vgl. ADOBE 2001)**

Die Referenzen sind aufgrund der fehlenden Entwicklungsumgebung besonders hilfreich. Sie unterstützen den Entwickler in großem Maße.

Um Skripte verwenden zu können, müssen diese in den SVG Baum eingebunden werden. Das `<script>` Element erlaubt interne sowie externe Scriptdefinition. Dabei muss zusätzlich die Scriptsprache sowie der einzusetzende Scriptinterpreter angegeben werden.

Sind Skripte eingebunden, können diese durch Events aufgerufen werden. SVG unterstützt verschiedene Events, die das Ausführen bzw. Starten von Scriptroutinen auslösen. FERRAILOLO et al geben eine ausführliche Übersicht aller unterstützten Events sowie deren **Eventhandler** (vgl. FERRAILOLO J. et al 2001, 16.2 Complete list of supported events). Die Wichtigsten sind User Interface Events, die beispielsweise durch das Betätigen einer Taste oder der Maus, ausgelöst werden. SVG lässt das Deklarieren von Events als Attribut direkt in einem Element zu.

Treten bei der Interpretation der Funktionen Syntaxfehler auf, erscheinen diese in einer Nachrichtbox. Aufgrund der fehlenden Entwicklungsumgebung gestaltet sich das Auffinden von Fehlermeldungen äußerst schwierig.

Die Implementierung des Prototypen nutzt das Grundkonzept der Objektorientierung. Diese ist in JavaScript, im Vergleich zu dem Konzept der Klassen und dem der Objekte höherer Programmiersprachen (z.B. in C++ oder Java), durch das Konzept der Prototypen implementiert. Es dient also nicht eine Klasse als Vorlage für ein Objekt, sondern ein prototypisches Objekt. Jedes Objekt kann zur Laufzeit durch Attribute und Methoden erweitert werden. Eine derartige Erweiterung betrifft aber ausschließlich das referenzierte Objekt. Soll die Veränderung alle Instanzen modifizieren, so erfordert dies eine Erweiterung des Prototypen durch die Eigenschaft `prototype`. Diese Eigenschaft kann genutzt werden, die **Vererbung** von Methoden und Eigenschaften, die im Sprachumfang von Javascript nicht unmittelbar vorgesehen ist, nachzuempfinden (vgl. ZIEGLER C. 2001; LINDSEY K. 2002).

3.2.2.6 Animationsmöglichkeiten

Für zeitgemäßes Design ist es wichtig, auch dynamische Vorgänge beschreiben zu können. Dafür sind Elemente der W3C Empfehlung für Multimedia **SMIL** in SVG implementiert. Weitere SVG-interne Animationselemente erweitern die Möglichkeit, Bewegungen darzustellen. Animationselemente werden als Kinder des zu animierenden Elementes deklariert. Dabei werden fast alle Elemente und Attribute unterstützt. SVG definiert echte zeitbasierte Animation. Daher ist alleine der **User Agent** für die Renderingzeit verantwortlich.

Eine Vielzahl von Animationsmöglichkeiten mit zugehörigen Beispielen beschreibt QUINT (vgl. QUINT A. 2002a). Teilweise kann durch geschickt programmierte Animation auf Scripting verzichtet werden. Ein beeindruckendes Beispiel zeigt HIRTZLER. Dabei ist eine Drop-Down Liste nur mit Elementen der Animation realisiert worden. Auf Scripting konnte deswegen komplett verzichtet werden (vgl. HIRTZLER M. 2002).

3.2.2.7 Sonstiges

Rastergraphiken werden mit dem `<image>` Element in den SVG Baum eingebunden. Viele der gängigen Graphikformate (dazu zählt auch SVG selbst) werden unterstützt. Auch base64 kodierte

Graphiken zur internen Bilddefinition können eingebunden werden. Rastergraphiken können intern oder extern eingebunden werden.

Zusätzliche Daten, wie beispielsweise Metadaten, können durch das `<metadata>` Element oder durch die Deklaration eines individuellen Namensraumes an den SVG Baum angehängt werden (vgl. FERRAILOLO et al 2001, Chapter 21+23).

Optimierungsmöglichkeiten:

Wie bei jeder Anwendung, bedarf es auch bei dem Einsatz von SVG einer Optimierung. Im Folgenden sind die wichtigsten Optimierungsmöglichkeiten besonders auf kartographische Anwendungen bezogen dargestellt. Viele gelten auch für andere Anwendungszwecke. Sie resultieren überwiegend aus praktischen Erfahrungen mit dem ASV3 und ergeben meistens eine Datenreduzierung

Optimierung der Pfaddefinition: Besonders effektive Optimierungsmöglichkeiten bietet das `pathData` Attribut. Dieses nimmt bei kartographischen Anwendungen die größte Datenmenge ein und beansprucht deswegen auch am meisten Renderingzeit. Eine besonders wirkungsvolle, nicht aufwendige und verlustfreie Möglichkeit Pfaddefinitionen zu optimieren ist die Verwendung relativer Koordinaten, sowie das Entfernen unnötiger Separatoren zwischen den Werten. Ist das Kommando einheitlich, so kann dieses weggelassen werden. Mit einem zu in Kauf nehmenden Verlust an Lagegenauigkeit lassen sich Werte auch auf ganze Zahlen runden. Weiter ist die Verwendung von Bezierkurven, am besten der quadratischen, verkürzende Kommandos, wie `h` und `v`, zu erwähnen. Optimierte Pfaddefinitionen verkürzen natürlich auch die Downloadzeit. Derartige Optimierung muss in den für die Geometrieerzeugung verantwortlichen Anwendungen, wie beispielsweise **Servlets**, vorgenommen werden.

Wiederverwendbarkeit der graphischen Elemente: Alle Elemente eines `<defs>` Knoten werden nicht gerendert und können über eine Verlinkung im `<use>` Element instanziiert werden. Das `<symbol>` Element verhält sich äquivalent. Beispielsweise werden sich wiederholende Objekte, wie im Prototypen dass bei Geoobjekten der Fall ist, einmal definiert und können demzufolge beliebig oft mit dem "schlankem" `<use>` Element instanziiert werden. Die Datenmenge reduziert sich dadurch erheblich, die Optimierung ist direkt in der SVG Syntax vorzunehmen.

Wiederverwendbarkeit von Formatvorlagen: Damit "styles" nicht in jedem Element neu definiert werden, sollten diese in CSS oder Entities festgelegt werden. Dabei ist es sinnvoll möglichst einheitliche Styledefinitionen festzulegen. Der jeweilige Stil wird durch seine Instanz im zu formatierenden Element zugewiesen. Weiter ermöglicht die Vererbung der Zeicheneigenschaften, dass beispielsweise nicht für jedes Element einer Gruppe eine Formatvorlage instanziiert werden muss, sondern dies über das entsprechende Containerelement geschehen kann. Diese Optimierungsmöglichkeit hat den schönen Nebeneffekt, dass Veränderungen nicht für jedes einzelne Element getätigt werden müssen (vgl. FERRAILOLO J. et al 2001, Chapter 11.8).

Streaming: Aufgrund dessen, dass SVG keine **Streaming** Möglichkeiten vorsieht, muss nach anderen Wegen gesucht werden. Generell ist das Nachladen von beispielsweise SVG Fragmenten, die nicht primär benötigt werden, durch die ASV3 spezifische `getURL()` Methode möglich. Diese Tatsache kann Streaming vortäuschen und somit lange und unnötige Downloads vermeiden.

Komprimierung: bei der Datenübertragung macht sich die `gzip` Komprimierung der Dateien erheblich bemerkbar. Wie bei FERRAILOLO et al zu sehen ist, erzielt diese Komprimierungsquoten von rund 80% (vgl. FERRAILOLO J. et al 2001, Appendix J).

Antialiasing: Rendereigenschaften, wie Antialiasing, können bei überflüssigem Gebrauch zu zeitaufwendigen Renderungen führen. Durch die ASV3 spezifische `setDefaultAntialias()` Methode kann Antialiasing unterdrückt und dadurch die Renderingzeit erheblich verkürzt werden. Jedoch ist damit Qualitätsverlust verbunden. Weiter stellt auch die SVG Spezifikation durch

Attribute, wie beispielsweise `shape-rendering`, Optimierungsmöglichkeiten zur Verfügung (vgl. FERRAIOLO J. et al 2001, Chapter 11.7 und SPONA H. 2001, S.309/310).

Javascriptoptimierung: Optimierungsmöglichkeiten in verwendeten Javascripten ist bei umfangreichem Scripting in Betracht zu ziehen. Beispielsweise verlängern überflüssige Schleifen die Laufzeit und sind daher zu vermeiden.

Weitere Hinweise zur Optimierung sind bei SOROTOKIN, FIBINGER und FERRAIOLO zu finden (vgl. SOROTOKIN P. 2002, Performance issues; FIBINGER I. 2002, S.330/331 und FERRAIOLO J. et al 2001, Appendix J).

3.2.3 SVG und Kartographie

Es stellt sich nun die Frage warum in dem Medien SVG meist mit der Kartographie in Verbindung gebracht wird. SVG gibt eigentlich eine Spezifikation vor, die zur allgemeinen vektorbasierten Visualisierung dient. Besondere Webmapping Eigenschaften sind deswegen nicht implementiert. Trotzdem stellt SVG einige für die Kartographie besonders vorteilhafte Features zur Verfügung. Die folgenden Abschnitte zeigen Stärken und Schwächen von SVG auf.

3.2.3.1 Stärken

Wichtigste Stärke von SVG ist, dass es noch keine Andere so weitreichenden spezifizierten Standards zur Vektorendarstellung im WWW gibt. Die Tatsache, dass SVG XML-basiert ist, muss auch als Stärke aufgefasst werden. Diese Kapitel behandelt weitere, speziell für die Kartographie wichtige Stärken von SVG.

Eine für Kartographische Gestaltung wichtige Anforderung ist der Aufbau einer Layerhierarchie. SVG bietet für diesen Zweck zwar keine optimale Lösung, doch können durch Containerelemente, wie `<g>`, indirekt Ebenen erzeugt werden. Dazu werden in der gleichen Hierarchiestufe des SVG Baumes die verschiedenen Ebenen definiert; die oberste Ebene steht als Letztes. Durch das Attribut `visible` können Layer auf einfach Art und Weise ein- beziehungsweise ausgeblendet werden.

Die aus **DTP** Programmen bekannten graphischen Features für Linien sind in SVG auf einfache Art und Weise zu realisieren. Dazu zählen beispielsweise verschiedene Linienenden (*engl. **linecap***) sowie unterschiedliche Linienverbindungen (*→ **linejoin***). Linienmuster, wie Strichlierung können auch einfach werden.

Außerdem stellt SVG weitere graphische Features wie Transparenz zur Verfügung. Das `<clipPath>` Element ermöglicht zudem das Ausschneiden des Inhalts oder der Umgebung eines Polygons (*→ **clipping***). Zudem ermöglichen die umfangreichen Möglichkeiten des `<filter>` Elementes die visuelle wertvollere Gestaltung.

Für Text gilt ähnliches. Die aus der Textverarbeitung bekannten Formatierungsmöglichkeiten sind bis auf Zeilenumbruch, der für die Kartographie selbst nicht notwendig ist, vorhanden (vgl. Kapitel 3.2.3.2.). Das `<textPath>` Element ermöglicht die objektbezogene Schriftausrichtung, wie beispielsweise Flussbeschriftungen. Weiterhin können eigene Schriften eingebunden werden, was die gewünschte Darstellung - unabhängig von installierten Systemschriften - ermöglicht.

Das `<pattern>` Element erlaubt das Definieren von Füllmustern, wie beispielsweise Texturen. Obgleich Schraffierung, wie in der Computergraphik üblich, nicht ganz einfach zu realisieren ist und auch speicheraufwendig sein kann, ist sie möglich. Aufgrund dessen, dass Schraffierung für die Bildschirmkartographie nicht geeignet ist, spielt diese "Schwäche" auch keine besondere Rolle (vgl. Kapitel 2.3.2.1.3).

SVG besitzt zwar ein differierendes Koordinatensystem (vgl. Kapitel 3.2.2.4), trotzdem lassen sich durch einfache Transformationen metrische Koordinaten (wie UTM) verwenden. Daher ist

beispielsweise Geokodierung ohne komplizierte Berechnungen möglich. Demzufolge lassen sich wirkliche Koordinaten auf einfache Art und Weise extrahieren.

Kartenteile können dynamisch verändert werden, ohne die ganze Darstellung noch mal erzeugen zu müssen; ein erheblicher Vorteil von Vektor- gegenüber Rastergraphiken.

SVG, beziehungsweise verwandte Technologien, bieten verschiedene Möglichkeiten des Nachladens von Daten. Das ist entweder durch ein **Request** mit Hilfe der **Scripting Engine** des Browsers, oder Plug-In spezifisch, durch die ASV3 spezifische `getURL()` Methode möglich.

3.2.3.2 Schwächen

Da SVG nicht speziell für Geoinformation geschaffen wurde, bringt es für kartographische Anwendungen auch einige Schwächen mit sich.

Das Problem vieler Vektordarstellungen ist, dass diese im kartographischem Kontext objektunabhängig formatiert sind. Diese Tatsache ist besonders bei Signaturen aus zusammengesetzten Objekten, wie beispielsweise einer Straße mit Füllung, problematisch. Der Pfad muss in diesem Fall mindestens zweimal definiert werden (vgl. WINTER A. 2000, S.90/91).

Weiter stellt SVG nur unzureichende Möglichkeiten zur räumlichen Navigation zu Verfügung. Kompliziertes Scripting ist dabei unabdingbar. Zwar lässt sich Zoomen durch das `ZoomAndPan` Attribut unterdrücken. Doch ist dieses nur auf das äußerste SVG Element anwendbar und für einen kartographischen Viewer mit verschachtelten SVGs nicht hilfreich. Deswegen müssen andere aufwendige Möglichkeiten gefunden werden, räumliche Navigation für Viewierzwecke zuzulassen. Zum einen ist räumliche Navigation durch die ASV3 spezifischen Attribute `currentScale` und `currentTranslate` zwar möglich, jedoch lassen sich diese nur auf das äußerste Element einer SVG Datei anwenden. Eine Standalone SVG Implementierung, die die Skalierung verschachtelter `<svg>` Elemente erfordert, wird dadurch unmöglich. Bedingte Abhilfe schafft die Einbettung von SVG in HTML. Hierbei ist jedoch problematisch, dass Scripte in den verschiedenen Namensräumen miteinander kommunizieren müssen. Diese Tatsache stellt sich in der Praxis als schwierig heraus, obgleich das SVG-Wiki derartige Möglichkeiten beschreibt (SVG-WIKI 2002, InterDocumentCommunication). Zum Anderen ist räumliche Navigation durch die Anpassung des `viewBox` Attributs denkbar. Um die einzelnen Viewbox Werte beim Skalieren zu berechnen, ist ein globaler Skalierungsfaktor nötig. Diese Variante lässt eine Standalone Lösung zu. Dabei wird nur eine SVG Datei geladen, ohne in eine HTML Seite eingebunden zu sein. Der Prototyp zeigt eine derartige Lösung. Verbesserungen, die auch eine **"Level of Details"** Implementierung ohne komplexe Scripte zulassen, sind sehr wünschenswert.

Ein weiteres großes Problem ist die Datensicherheit. Diese kann bei einem textbasiertem Format gegenwärtig noch nicht vorhanden sein. Dies führt in der Entwicklergemeinde zu einer großen Barriere bezüglich des SVG Einsatzes. Obwohl besonders trickreiche Anwendungen dem ersten Anschein nach Einsicht des Quellcodes verhindern, können beispielsweise durch **"Sniffer"**, alle Daten, die vom Server zum Client übertragen werden nachvollzogen werden. Jedoch ist dieses Problem dem W3C bekannt und eine Empfehlung zur Ver- und Entschlüsselung von XML Dateien wurde verabschiedet (vgl. EASTLAKE D. & REAGLE J. 2002). Auch umfangreiche clientseitige Scriptroutinen können nicht verschleiert werden. Gründe und Gedanken sind bei LEY zu finden (vgl. LEY J. 2002).

Als weitere Schwäche gelten auch die fehlenden Ausdruckmöglichkeiten von SVG Graphiken. Die Druckfunktion des Browsers stellt gegenwärtig die einzige Möglichkeit dar, Karten auszudrucken. Jedoch kann diese nicht als allgemeine Möglichkeit angesehen werden, da dabei immer wieder Probleme auftreten.

Hinweis: Für kartographische Zwecke Ungeeignetes oder Fehlendes, kann durch Vorschläge an die SVG Arbeitsgruppe des W3C's vielleicht in der nächsten Version der SVG Spezifikation berücksichtigt werden. Dazu sollten zuerst, die für zukünftige SVG Versionen bereitstehenden Verbesserungsvorschläge, eingesehen werden. Die Praxis zeigt, dass die SVG Arbeitsgruppen für Vorschläge jeglicher Art offen ist und versucht, diese zu berücksichtigen.

3.2.4 Implementierung

Implementierung schließt benutzerfreundliche Erzeugung, "Editierbarkeit" und Möglichkeiten zur Betrachtung von SVG Dateien ein. Dieser Punkt ist sehr wichtig für die Akzeptanz von SVG. Die in diesem Kapitel folgenden Abschnitte sollen einen aktuellen Überblick über Implementierungsmöglichkeiten geben (Stand 11/2002). Eine umfangreiche und ständig aktualisierte SVG Implementierungssammlung ist außerdem bei LILLEY und im SVG-WIKI zu finden (vgl. LILLEY C. 2002 bzw. SVG-WIKI 2002, Viewer- bzw. Other Implementations).

3.2.4.1 Anwendungen zur Erzeugung von SVG

Anwendungen zum Erzeugen von SVG lassen sich in zwei Kategorien einteilen. Einerseits Texteditoren, die keinerlei graphische Unterstützung bieten. Andererseits WYSIWYG Autorenwerkzeuge für die Erstellung anhand einer graphischen Benutzeroberfläche.

3.2.4.1.1 Texteditoren

Da XML respektive SVG textbasiert sind, können sie in jedem beliebigen Texteditor erstellt werden. Texteditoren mit Syntaxhighlighting und anderen komfortablen Editorfunktionen sind optimal (z.B. TextPad, Ultraedit, Vim).

Besser eignen sich professionelle XML Editoren mit SVG Unterstützung, wie beispielsweise XMLSpy. Vorteile dieser sind zum einen das Syntaxhighlighting, sowie zum anderen die verschiedenen Ansichtsmöglichkeiten einer Datei. Unter verschiedenen Ansichten versteht man beispielweise reine Quellcodedarstellung, Baumdarstellung (DOM) oder Vorschaumodus. Auch Syntaxüberprüfung ist ein weiteres angenehmes Feature, jedoch gibt es noch keine Entwicklungsumgebung, die Projekte mit Scriptingfunktionalität validieren (→ **gültig**) kann (vgl. Kapitel 4.3.1).

3.2.4.1.2 WYSIWYG Autorensysteme

WYSIWYG Editoren zeichnen sich durch die graphische Unterstützung beim Editieren von Dateien aus. In diesem Fall sind das, da SVG ein Graphikformat ist, üblicherweise auch Graphikprogramme. Unterschieden werden Programme, die SVG als eigenes Dateiformat verwenden (z.B. Jasc Webdraw und SVG Studio) von Graphiksoftware, die eine Möglichkeit besitzen, SVG Dateien zu exportieren (z.B. Adobe Illustrator, Corel Draw, Mayura Draw, OpenOffice.org Draw). Gute Autorenwerkzeuge schließen einen Texteditor mit ein.

Eine ausführliche und aktuelle Übersicht von WYSIWYG Autorensystemen bietet SCHUMACHER (vgl. SCHUMACHER S. 2002, S.60–67).

Hinweis: Problematisch bezüglich automatisch erzeugtem Quellcode ist die oftmals daraus resultierende **Spagetti Struktur**. Diese erweist sich als ungeeignet, besonders an der großen Datenmenge und unbrauchbaren Datenstruktur zu erkennen. Für kleinere Projekte und zur reinen Geometrierstellung reichen WYSIWYG Editoren aber meist aus, obgleich bei manchen auch hier noch Abstriche zu machen sind. Jasc Webdraw beispielsweise kann in der aktuellen Version nicht mit relativen Koordinaten umgehen und unterstützt das <use> Element nicht. Sobald Javascripte implementiert werden sollen, sind Texteditoren der einzige Ausweg.

3.2.4.2 Exportmöglichkeiten

Ein Exportskript, das SVG konforme Dateien erzeugt, ist meistens auf einfache Art und Weise zu realisieren (vgl. Kapitel 3.2.4.4.2). Wie oben schon diskutiert, gibt es Graphikprogramme, die einen SVG Export ermöglichen (Kapitel 3.2.4.1.2). Im Folgenden werden Exportmöglichkeiten aus

Geoinformationssystemen besprochen, wobei diese zu unterschieden sind. Zum Einen gibt es Werkzeuge, die neben der Geometrie zusätzlich noch Funktionalität, wie Layerkontrolle oder Attributtabelle schreiben. Diese, meist kommerziell angebotenen Werkzeuge, sind noch am Anfang ihrer Entwicklung. Erfahrungen zeigen, dass sie gegenwärtig noch ungeeigneten Quellcode erzeugen und Darstellungen zudem kartographisch falsch sein können. Zum Anderen sind einfache Exportskripte zu nennen, die reine Geometrie erzeugen. Diese eignen sich besonders zum einfachen Auslesen von Koordinaten aus verschiedenen Datenformaten (z.B. **MID/MIF**).

Hinweis: Eine breite Palette an Export Tools ist für ArcView 3x (z.B. UISMEDIA, SVGMAPPER, SHP2SVG) und Mapinfo erhältlich (z.B. DBXGEOMATICS). Auch TNT unterstützt mittlerweile in der Basisversion guten SVG Export (vgl. MICROIMAGES 2002, SVG Gallery). Als weiteres Produkt mit kartographischem Bezug ist noch der Mappublisher von AVENZA zu erwähnen (vgl. AVENZA SYSTEMS Inc.). Die Entwicklung zeigt, dass die Zahl der Exportmöglichkeiten, ob kommerziell oder frei, fortwährend zunimmt.

Auch einige **UML** Editoren unterstützen mittlerweile SVG Export. Bekannte Beispiele sind Poseidon für UML (vgl. GENTLEWARE 2002) und Together (vgl. INTERNATIONAL SOFTWARE 2002).

3.2.4.3 Vektorisierungsanwendungen - Druckertreiber

Zudem gibt es Konverter die Rastergraphiken, wie jpeg Dateien, vektorisieren. Breite Anerkennung erfährt gegenwärtig das CELINEA Projekt, mit der Kommandozeilenanwendung CR2V (vgl. CELINEA 2000). Der Vollständigkeit halber ist noch der SVG Druckertreiber von SOFTWARE MECHANICS zu erwähnen, der dem **PDF** Writer ähnlich ist (vgl. SOFTWARE MECHANICS 2002, SVGMaker). Beide stellen sehr einfache Methoden dar, SVG Dateien zu erzeugen, wenngleich der erzeugte Code dementsprechend primitiv ist. Die daraus resultierenden Vektoren eignen sich nicht für eine Weiterverarbeitung, wie beispielsweise zur Bearbeitung durch Scripte.

3.2.4.4 Andere Erzeugungsmöglichkeiten

3.2.4.4.1 XSL zur Transformation nach SVG

Wie in Kapitel 3.1.2 diskutiert, lassen sich XML Dateien auf sehr einfache Art und Weise transformieren. Besonders die XSL Transformierung ist bestens dazu geeignet, SVG's zu erzeugen. Gegenwärtig gelten diese aufgrund zu hoher Rechenzeit nicht als wirkliche Alternative zu serverseitigen Implementierungen, wie sie im nächsten Kapitel beschrieben werden. Besonders clientseitig wird XSLT noch einige Zeit benötigen, da Implementierungen momentan fehlen. Großes Hindernis ist außerdem, dass nach SVG transformierte Dateien (clientseitig) noch immer in HTML Seiten eingebettet werden müssen. Die allgemein übliche Browser und Plug-In Implementierung mit dem Internet Explorer und dem ASV3 unterstützt keine Transformation in eigenständiges SVG. Diese Tatsache resultiert aus einem Konflikt zwischen dem verwendeten Browser und Plug-In (vgl. YAHOO SVG-DEVELOPERS GROUP 2002, Message 23927). Ausführlich beschriebene Beispiele zur XSL Transformation sind bei CAGLE et al zu finden (CAGLE K. et al 2001, S.433-461).

3.2.4.4.2 Programmierung

Wie unter 3.2.4.2 schon erwähnt wurde, bedarf es nicht besonderer Programmierkenntnisse, ein Exportwerkzeug zu erstellen, dass Daten als XML/SVG schreiben kann. Bestes Beispiel, wie simple SVG Code erzeugt werden kann, sind primitive Javascripte, die zur Laufzeit SVG Elemente erzeugen (vgl. Kapitel 3.2.2.5). Aufgrund dessen, dass SVG textbasiert ist, kann es sehr einfach geschrieben werden.

Auch serverseitig können mit vielen verschiedenen technologischen Möglichkeiten Klassen oder Skripte programmiert werden, die SVG Code schreiben. Besonders beliebt sind Servlets um SVG Dateien zu erzeugen. Auch **ASP** Seiten mit eingebundenem SVG erfreuen sich großem Zuspruch.

3.2.4.4.3 Toolkits

Ein Toolkit ist eine Sammlung von Anwendungen. Diese umfassen programmierte Bibliotheken zum Erstellen, Weiterverarbeiten und Betrachten von SVG Dokumenten (vgl. auch Kapitel 3.2.4.5.3). SVG Toolkits sind in einer Programmiersprache geschrieben, meist in Java.

Besonders populär ist der **Batik** SVG Toolkit des Apache Projekts (vgl. APACHE 2002b), ein anderer ist der CSIRO SVG Toolkit (vgl. CSRIO 2002). Der Vorteil liegt in der plattformübergreifenden Einsetzbarkeit. Jedes Betriebssystem, dass das Java Runtime Environment installiert hat, kann diese Klassen nutzen. Sie können deswegen eine Serverimplementierung aber auch eine Clientimplementierung unterstützen.

Hinweis: Die Bitmaps in Abbildung 6 "Visuelle Variablen in Abhängigkeit der Implantation" sind mit dem Batik-Rasterizer erstellt worden (vgl. Kapitel 2.2.3.2).

3.2.4.5 SVG Viewer

SVG Viewer sind Anwendungen, die SVG darstellen können. Sie müssen SVG gemäß der Spezifikation implementieren und sollten weiter Features, wie beispielsweise DOM oder CSS unterstützen. Gegenwärtig dienen vor allem Plug-In's dazu, SVG darzustellen. Doch auch über den Einsatz native Browserunterstützung und eigenen Vieweranwendungen sollte nachgedacht werden.

Hinweis: Großes Manko des SVG Standards sind mangelnde Viewerimplementierungen, denen vor allem Scriptingmöglichkeiten fehlen. Im Folgenden werden verschiedene Viewer Implementierungen diskutiert.

3.2.4.5.1 Plug-In Lösung

Browser Plug-In's dienen dazu, einem Browser zu erweitern, damit zusätzliche Dateiformate gerendert werden können. Der am weitesten verbreitetste für SVG, ist der Adobe SVG Viewer (zur Zeit in Version 3 vorliegend; vgl. ADOBE 2002). Weiter ist seit Kurzem auch ein Preview des Corel SVG Viewer Plug-In's erhältlich (vgl. COREL 2002), der dem ASV 3 sehr ähnelt.

Probleme diesbezüglich sind leicht auszumachen, da der Webentwickler nicht davon ausgehen kann, dass der Client diesen installiert hat. Man kann sich zwar programmiertechnischer Art mit Abfrage Statements behelfen, die den Client auf entsprechende Plug-In's untersuchen. Doch kann dieser Weg nicht als Lösung angesehen werden, einen Standard zu verbreiten. Auch eine eventuelle Installation kann nur als Zwischenschritt zur Etablierung eines Webstandards gesehen werden. Oftmals sind Administratorenrechte oder Spezialwissen notwendig, um das Plug-In einzurichten. Viele Produkte der Firma Adobe (z.B. Acrobat Reader 5) installieren den ASV3 vom Anwender unbemerkt mit (vgl. SVG-WIKI 2002, Adobe SVGViewerdistribution). Diese Tatsache trägt zwar zu einer größeren Verbreitung bei, doch sie ist nicht "im Sinne eines Webstandards". Zudem erlauben viele Systemadministratoren nicht, Plug-In's zu installieren. Die Praxis zeigt, dass sie ein Sicherheitsrisiko darstellen. Beispielsweise kann obwohl Scripting durch Browsereinstellungen nicht zugelassen wird, ein Plug-In Scriptinginterpreter Skripte ausführen. Lösen kann die fehlende Plug-In Akzeptanz eingeschränkt nur einfache Handhabung, wie Installation, kurze Downloadzeiten zur Installation und plattformübergreifende Verbreitung. Optimalen Zustand können aber nur native Browserimplementierungen erzielen (vgl. nächstes Kapitel).

Hinweis: Bei Windows kann über die **COM** Schnittstelle auf ASV3 Objekte aus jeder Applikation zugreifen. Dieser Möglichkeit bedient sich beispielsweise der Vorschaumodus von Jasc Webdraw.

3.2.4.5.2 Native Browserimplementierungen

Im Gegensatz zur Plug-In Lösung bei Webbrowsern stehen "native" SVG Implementierungen in Webbrowsern. Sie benötigen kein Plug-In, um SVG spezifizierte Dateien darstellen zu können - vergleichbar mit der HTML Implementierung von Webbrowsern. Noch gibt es keinen praktikablen nativen SVG Browser, doch ein **Mozilla** Projekt arbeitet daran und erste Beispiele sind zu sehen (vgl. MOZILLA SVG PROJEKT 2002). Bleibt abzuwarten ob native Browser in der Lage sind, komplexe SVG Graphiken zufriedenstellend zu verarbeiten. Es zeichnet sich ab, dass die Geschwindigkeit erheblich geringer sein wird, als bei standalone Viewern..

3.2.4.5.3 Standalone Viewer

Standalone Viewer sind als eigenständige Anwendung zu sehen. Sie bringen eine eigene Rendering Engine mit. Besondere Eigenschaft ist schnelleres Visualisieren von SVG Dateien. Zu erwähnen ist besonders der in Java programmierte **Squiggle** SVG Browser (vgl. APACHE 2002b, Squiggle SVG Browser). Gegenwärtig ist dieser die properitärste Implementierung. Fast alle SVG Elemente werden unterstützt, jedoch müssen noch Einschränkungen gemacht werden. Beispielsweise werden wichtige Scripting Methoden noch nicht voll unterstützt. Er ist Bestandteil des Batik Toolkits (vgl. 3.2.4.4.3). Außerdem erfolgsversprechend ist das SharpVectorGraphics Projekt, das einen SVG Viewer für Microsoft's **.NET** Framework erstellt. Dieser hat die wesentlichen SVG Elemente implementiert, wobei Scripting noch nicht möglich ist. Er befindet sich noch im Anfangsstadium der Entwicklung (vgl. SHARPVECTORGRAPHICS 2002).

3.3 System Integration – Architektur

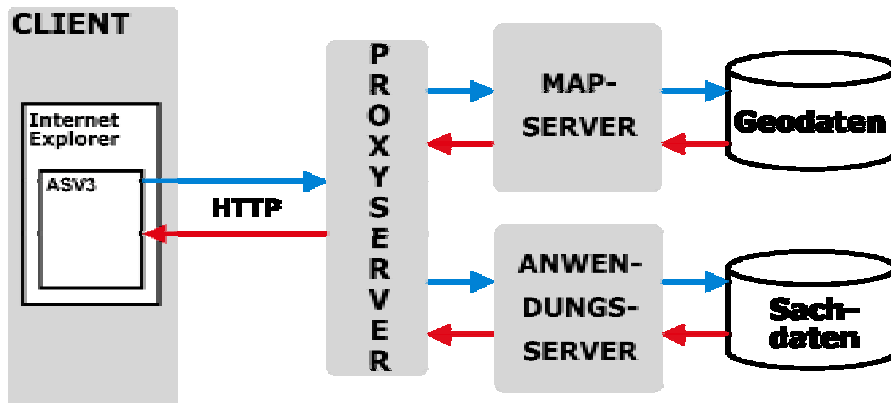
Um SVG für Webmapping/WebGIS als Darstellungsformat einzusetzen, ist es wichtig die Integrierbarkeit einer SVG basierten Clientlösung in eine Systemarchitektur zu untersuchen. Dabei wird anhand eines möglichen Systemaufbaus ein WebGIS mit SVG Darstellungskomponente erläutert. Die darauffolgenden Kapitel behandeln die einzelnen Komponenten sowie deren Funktion.

Hinweis: Es gibt viele verschiedene Systemarchitekturen. Hier wird nur eine mögliche beschrieben. Weitere Architekturen im Bezug auf Business Intelligence sind bei BANGE et al zu finden (vgl. BANGE C. & MERTENS H. & KELLER P. 2001, S.1-64)

3.3.1 Überblick

Um einen Überblick zu bekommen zeigt Abbildung 37 die Minimalanforderung an ein webbasiertes Geomarketingsystem für Business Intelligence. Es handelt sich dabei um ein typisches WebGIS System zur properitären Kartenherstellung.

Abb. 37: Vereinfachte Systemarchitektur



Quelle: Eigene Darstellung.

Wie in Abbildung 37 zu sehen ist, stellt ein Client eine Anfrage (Fachbegriff "Request"), die der **Proxy** an den jeweiligen Server weiterleitet.

Auf der einen Seite verfügt das System über einen Mapserver, der auch als Geokomponente bezeichnet. Dieser stellt die notwendigen Geodaten in Form von SVG Dateien bereit. Ein Anwendungsserver selektiert notwendige Sachdaten und gibt diese als XML Dokument zurück. In der Fachwelt bezeichnet man die Antwort des Servers als "**Response**". Der Client, der aus einem XML/SVG fähigen Anwendung besteht, verarbeitet beide Datensätze zu thematischen Karten. Zur Kommunikation zwischen Client und Server dient das in der Webwelt übliche HTTP Protokoll.

Neben einer erfolgreichen Integration der einzelnen Komponenten ist die Zuverlässigkeit aller Systemkomponenten der wichtigste Faktor.

3.3.2 Datenhaltung

Datenbanken dienen zur sinnvollen strukturierten Haltung der Sach- und Geodaten. Datenbankmanagementsysteme (DBMS) schaffen die Voraussetzung, um mit Datenbanken zu arbeiten. Mit ihnen lassen sich Datenbanken definieren, kreieren und manipulieren. Dafür wird der allgemein übliche **SQL** Standard eingesetzt.

Im Gegensatz zur Lagerung von komplexen und großen Datenmengen in Dateien, bringen Datenbanken verschiedene Vorteile mit sich. Zum einen ist die hohe Performance von Datenbanken zu nennen. Zum anderen die besonders einfachen Methoden, mit Datenbanken zu arbeiten (vgl. RIGAUX P. & SCHOLL M. & VOISARD A. 2002, S.3-8).

3.3.2.1 Geodaten in Datenbanken

Geodaten erfordern aufgrund des Raumbezuges andere Ansprüche an ein Datenbanksystem als Sachdaten. Deswegen ist es wichtig, sich ausführlich Gedanken über die Geodatenhaltung zu machen. Allgemein gibt es zwei Typen von Datenbanksystemen, die mit (Spatial DBMS) und die ohne Raumbezug.

"Räumliche" Datenbanken ersetzen den Einsatz von GIS spezifischen Dateiformaten in gewisser Weise, da auch topologische Eigenschaften der Information gespeichert werden (z.B. **Bounding Box** oder **Zentroid**). Ein Datenbanksystem mit "Spatial Erweiterung" hat zudem den großen Vorteil, GIS Funktionalität in eine Client-Server Struktur einzubringen. GIS-Analysen können eingeschränkt mit vom Datenbankmanagementsystem bereitgestellten Methoden bewältigt werden. Datenbanksysteme ohne Raumbezug besitzen dagegen keine zusätzlichen Methoden und Eigenschaften zur räumlichen Bearbeitung. Andere Anwendungen, wie beispielsweise der Mapserver, sind dann gefordert, GIS Funktionalität zur Verfügung zu stellen.

Weiter können kommerzielle (z.B. **Oracle** + **Oracle Spatial**, DB2, MS Access) von Opensource Datenbanken (z.B. **MySQL**, **PostgreSQL** + **PostGIS**) unterschieden werden. Datenbanken aus freien Entwicklergemeinschaften stellen sich als preiswerte Alternativen heraus.

Für ein Testsystem sind Datenbanken ohne Raumbezug ausreichend. Die Koordinaten werden bereits in SVG Fragmenten in der Datenbank gespeichert. Diese werden vom Mapserver ausgelesen und an den Client weitergegeben.

3.3.2.2 Sachinformationen

Sachdaten werden in relationalen Datenbanken mit Geomarketing- sowie Unternehmensdaten gehalten. Dabei wird auf vorhandene Datenbanksysteme beispielsweise eines Unternehmens zurückgegriffen. Zum Beispiel setzt Business Intelligence **multidimensionale Datenbanken** ein. Die Anwendungslogik eines entsprechenden Systems greift auf diese zu und verarbeitet sie durch multidimensionale Analysen weiter.

3.3.3 Serverkomponenten

Die Serverkomponenten eines WebGIS dienen dazu, auf Anfrage eines Clients die gewünschte Karte zu erstellen und an den Client zurückzugeben. Im Folgenden werden die wichtigsten Serverkomponenten eines für die SVG Ausgabe geeigneten WebGIS beschrieben.

3.3.3.1 Mapserver

Die Aufgabe eines Mapservers besteht darin, die geforderten Geodaten bereitzustellen. Als Mapserverimplementierung dient ein Servletcontainer, wie beispielsweise der Apache Tomcat (vgl. APACHE 2002c). Die Anfrage des Clients richtet sich an ein Java Servlet. Dieses ist als Anwendung auf dem Mapserver zu sehen. Dabei selektiert es Daten aus der Geodatenbank und gibt die Informationen in Form von SVG Fragmenten an den Client zurück.

Für den Prototypen entsteht beispielsweise folgendes Szenario:

Beim Drill-Down Mapping wird der Identifikationsschlüssel des Geoobjektes als Parameter an ein Servlet übergeben. Dieses selektiert die Objekte der neuen Kartenebene aus der Geodatenbank. Der Client erhält die SVG Fragmente und fügt diese dem SVG Baum an.

Hinweis: Wird ein serverseitig generiertes SVG an den Client übertragen, sollte dabei immer der MIME Type mitgeschickt werden, da ansonsten Browser Probleme haben, SVG Dateien eindeutig zu identifizieren. Dateitypidentifizierung anhand der Extension funktioniert nur bei Microsoftprodukten (vgl. SVG-WIKI 2002, MimeType).

3.3.3.2 Anwendungsserver

Auf einem Anwendungsserver befindet sich die Sachdatenkomponente. Eine konkrete Implementierung wird über Datenbankzugriffe die gewünschten Informationen aggregieren. Anschließend werden durch eine XSL Transformationen die Daten im vom Client geforderten XML Format bereitgestellt. Gegenwärtig ist der Einsatz von Web Services auf dem Anwendungsserver prädestiniert. Diese unterstützen in einem hohen Maße proprietäre Webtechnologien.

Die Praxis zeigt, dass in Business Intelligence Anwendungen meist mehrere Anwendungsserver zum Einsatz kommen. Es entsteht eine Anwendungslogik die komplexe Analysen durchführen kann.

3.3.3.3 Proxyserver

Aufgrund von Sicherheitsrichtlinien des Clients, können weitere Daten nur von einem Server geladen werden. Da mindestens immer zwei Server eingesetzt werden, wird ein transparenter

Proxyserver benötigt. Dieser gibt die Anfragen an die verschiedenen Server weiter. Somit ist der **Host** immer gleich (vgl. YAHOO SVG-DEVELOPERS GROUP 2002, Message 22311). Zudem steigert ein Proxy unter Umständen Systemgeschwindigkeit, da Anfragen gespeichert (umgangssprachlich "gecached" → **Cache**) und bei Wiederholung aus dem Cache geladen werden.

Hinweis: Sind Web- und Applikationsserver auf einem Host, so ist ein Proxyserver nicht zwingend notwendig.

3.3.4 Der Client zur Darstellung

Als Client wird der Teil des Systems bezeichnet, der die Anfrage an einen Server stellt. Weiter sorgt er für die Visualisierung der Daten. Je nach dem, welches Darstellungsformat verwendet wird, sind verschiedenen Anforderungen zu stellen. Für das vorliegende Konzept zum visualisieren mit SVG ist dabei folgendes notwendig.

Zum einen muss der Client SVG spezifizierte Dateien unterstützen. Zum anderen muss dieser einen ECMA 262 konformen Javascriptinterpreter besitzen. Beim Test des Prototypen wird ein Microsoft Internet Explorer 6 mit ASV3 Plug-In verwendet. Betriebssystem ist Windows 2000.

Hinweis: Im Zusammenhang mit der Wahl der Technologie des Clientes wurde auch der Einsatz von Java Applet's in Betracht gezogen (→ **Applet**). Derartige Tools diskutieren CECCONI in seiner Diplomarbeit (vgl. CECCONI A. 1999), ANDRIENKO & ANDRIENKO im deutschen Forschungsprojekt "descartes" (vgl. ANDRIENKO G. & ANDRIENKO N. 1999) und bezüglich Geomarketing, CORBLEY (vgl. CORBLEY K. 2002). Die vorgestellten Lösungen können in gewisser Weise als Pendants zu diesem Prototypen betrachtet werden; jedoch mit anderer Technik.

3.3.5 Datenübertragung

Die Übertragung von Geodaten erfordert schnelle Datenübertragung. Diese Tatsache resultiert aus dem großen räumlichen Informationsgehalt von Karten. Vor allem Datenreduzierung mit akzeptablem Informationsverlust trägt zum schnelleren Laden des Viewers bei. SVG hat zudem den großen Vorteil, nur zusätzlich benötigte Daten nachladen zu müssen. Einmal übertragene Daten, dies gilt auch für Sachdaten, sind solange die Anwendung geöffnet ist verfügbar. Zudem tragen proprietäre Netzwerktechnologien dazu bei, die Geschwindigkeit des ganzen Systems zu steigern. Zur Kommunikation zwischen Webserver und Client dient das in der "Netzwelt" übliche HTTP Protokoll. Dieses ist vom W3C spezifiziert und sehr weit verbreitet (vgl. LAFON I. 2001).

4 Der Prototyp

Weitere Aufgabe der Arbeit war es, die theoretischen Ausführungen des zweiten Kapitels anhand der behandelten Technologien (vgl. Kapitel 3) durch einen Prototypen zu realisieren. Dieser muss aufzeigen, wie ein vollständiger Viewer entwickelt werden könnte. Dabei sind Darstellungen aus dem Geomarketing berücksichtigt worden. Zudem zeigt er wichtige Funktionalität eines Web-mapping-Clients auf. Die folgenden Kapitel besprechen den realisierten Prototypen umfassend.

Hinweis: Ein kompletter Viewer unterscheidet sich von einem Prototypen hauptsächlich in einem Punkt. Mit ihm muss gearbeitet werden. Ein prototypischer Viewer dient dagegen nur zu Anschauungszwecken und soll den zukünftigen Viewer gerechtfertigen.

4.1 Anforderungen an die Anwendung

Um die theoretischen Grundlagen anhand des Prototypen vermitteln zu können, müssen verschiedene Anforderungen gestellt werden. Dabei stehen folgende Gesichtspunkte an erster Stelle.

Ein prototypischer Viewer für thematische Karten muss die von einem Server transferierten Daten geeignet visualisieren können. Die Wissenschaft der Thematischen Kartographie gibt dafür die Grundlagen. Sowohl im Geomarketing übliche Karteninhalte als auch erforderliche Funktionalität gilt es zu realisieren.

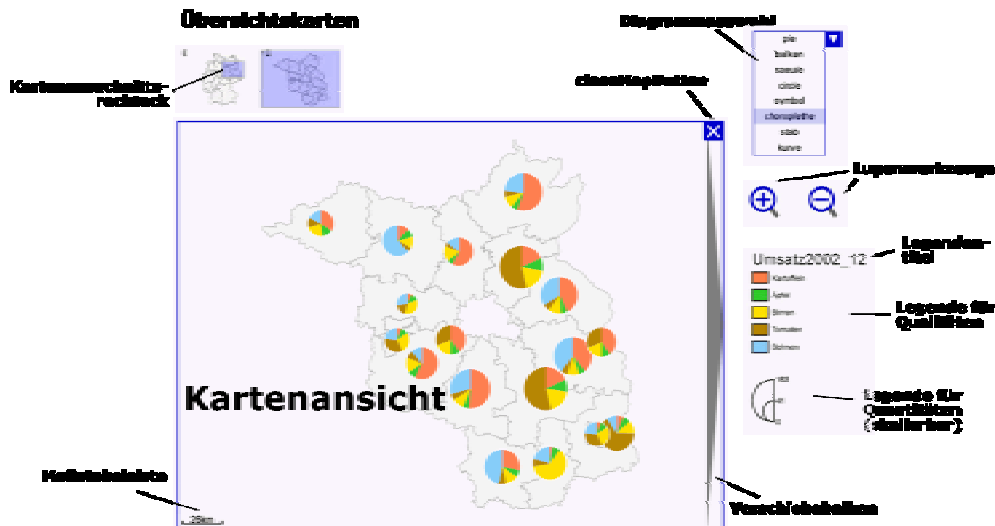
Bezüglich des Karteninhaltes sind besonders Diagrammimplementierungen von Bedeutung. Diese müssen Kreissektoren, Balken, Säulen, Kurven, Stäbe und Symbole beinhalten. Weiter spielen Choroplethenkarten eine wichtige Rolle. Zu den verschiedenen Darstellungstypen sind außerdem Legenden erforderlich. Eine Maßstabsleiste sowie eine Übersichtskarte müssen zur räumlichen Orientierung vorhanden sein. Die erforderliche Funktionalität schließt besonders räumliche Navigation ein. Dazu gehören Zoomen und Verschieben der Ansicht durch entsprechende Werkzeuge. Auch das im Geomarketing sehr beliebte Drill-Down Mapping beziehungsweise Roll-Up Mapping müssen realisiert werden. Dafür benötigte Geo- und Sachdaten sind nachzuladen. Eine kartographische Benutzeroberfläche gibt dafür den Rahmen.

Um eine derartige Anwendung zu erweitern und zu personalisieren, ist zeitgerechte Programmierung notwendig. Dazu soll die Anwendung auf den drei Komponenten Geoinformation, Thematik und Eigenschaften aufbauen. Entwickelte Javascriptmodule in Form von Objekten dienen zur Verarbeitung dieser. Die Integration in vorhandene Systeme muss besonders einfach zu realisieren sein. Deswegen ist der Einsatz von spezifizierten Techniken notwendig. Anwenderfreundlichkeit muss gegeben sein. Dazu zählen einfaches und zuverlässiges Benutzen sowie gutes "Bechmarking" (→ **Benchmark**). Letztendlich soll die vorgestellte Lösung ein Demonstrationsprototyp sein, dessen Ziel es ist, dem Anwender detaillierte Vorteile eines zukünftigen Viewers aufzuzeigen.

4.2 Funktionalität des Prototypen

Die Funktionalität beschreibt die realisierten Anforderungen des Prototypen. Die Beschreibung lehnt sich an die implementierte kartographische Benutzeroberfläche (Abbildung 38).

Abb. 38: Die implementierte kartographische Benutzeroberfläche



Quelle: eigene Darstellung.

4.2.1 Diagrammauswahl im Prototypen

Im Falle des Prototypen steht eine Drop Down Liste zur Diagrammauswahl in der Anfangsansicht zur Verfügung. Es sind Kreissektoren-, Balken-, Säulen-, Kreis-, Symbol-, Kurven- und Stabdiagramme implementiert, wobei Kreissektorendiagramme die Startansicht darstellen. Weiter ist eine Choroplethenkarte realisiert, die aus Sicht der Programmierung zu den Diagrammtypen zählt. Die Diagramme sind, wie ihre zugehörigen Geoobjekte, mit Funktionalität ausgestattete Kartenelemente. Sie dienen dem Drill-Down Mapping. Falls ein Diagrammtyp neue Sachdaten erfordert, werden diese nachgeladen. Ein mal geladene Daten bleiben der Anwendung erhalten. Außerdem steht eine Legende für den jeweiligen Diagrammtyp zur Verfügung.

4.2.2 Räumliche Navigation und Orientierung in der Karte

Die räumliche Navigation ist wichtiger Bestandteil eines kartographischen Viewers.

Zum Einen dienen virtuelle Balken an den Kartenrändern zum Verschieben der Karte. Beim Überfahren mit der Maus werden diese sichtbar und können betätigt werden. Betätigt der Anwender die Balken in kurzen Abständen mehrmals mit der Maus, ändert sich der Verschiebefaktor auf den fünffachen Wert. Das hat für den Benutzer den Vorteil, schneller navigieren zu können.

Zum Anderen ermöglichen interaktive Lupen das Vergrößern und das Verkleinern der Ansicht. Soll eine Karte vergrößert/verkleinert werden, ist die entsprechende Lupe ("+" / "-") zu betätigen. Dabei wird um einen konstanten Faktor gezoomt. Ähnlich den Balken zum Verschieben einer Karte, können auch Lupen in kurzen Abständen mehrmals vom Anwender mit der Maus bedient werden. Der Zoomfaktor ändert sich dabei auf den fünffachen Wert. Der Schwerpunkt des Rechtecks verhält sich beim Zoomen statisch.

Zusätzlich ist das geforderte Drill-Down Mapping implementiert. Beim Anklicken eines Geoobjekts oder dessen Diagramms werden die benötigten Geo- und Sachdaten der neuen Kartenansicht von den Servern des Systems nachgeladen. Zudem zoomt die Ansicht auf die neuen Geoobjekte. Anhand der neuen Sachdaten werden die Diagramme gezeichnet. Der Diagrammtyp sowie die Datenart ändert sich beim Drill-Down Mapping nicht. Gleichzeitig passt sich der Skalierungsfaktor und alle verbundenen Elemente, wie der Maßstabsbalken, die Legende und die

Übersichtskarte an. Über einen Button im Kartenfenster ("closeMapButton") und durch die Übersichtskarte einer höheren Ebene ist auch das Roll-Up Mapping möglich. Die Daten vorheriger Kartenebenen bleiben wie auch beim Drill-Down Mapping der Anwendung erhalten und müssen bei wiederholtem Aufruf nicht noch mal vom Server geladen werden.

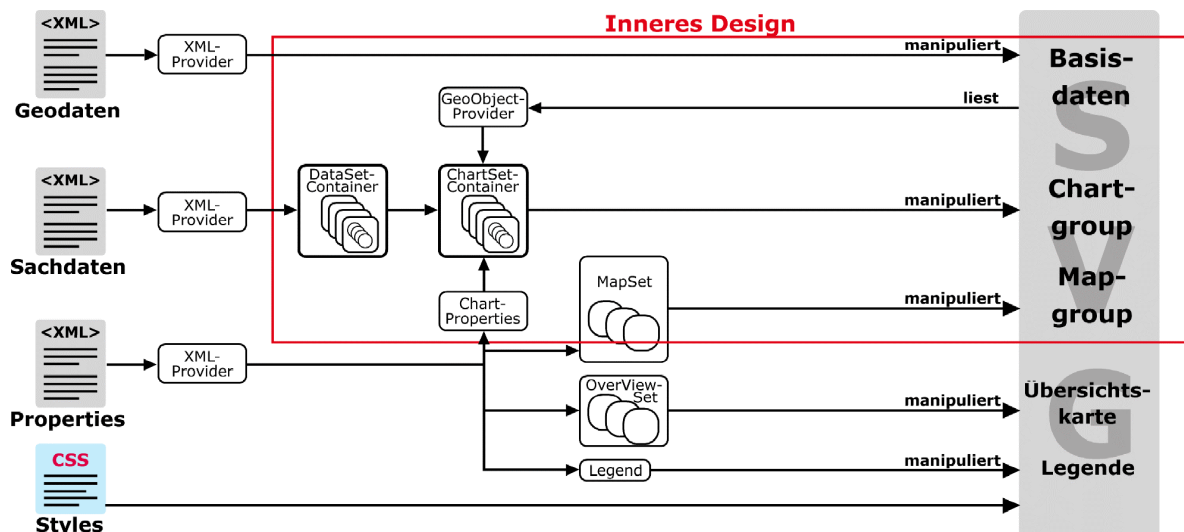
Übersichtskarten dienen zur räumlichen Orientierung. Ein Kartenausschnittsrechteck in der aktuellen Übersicht zeigt die Lage und Größe der Kartenansicht. Die Kartenausschnittsrechtecke in den Übersichtskarten höherer Kartenebenen stellen dagegen die Größe der untergeordneten Detailebene da. Sekundär dienen Übersichtskarten auch als Navigationselemente zum Roll-Up Mapping. Man spricht von interaktiven Übersichten (vgl. Kapitel 2.3.2.2.2). Der Titel einer Übersichtskarte zeigt die Kennziffer des dargestellten Gebietes (z.B. Landeskennziffer). Die Legende wird beim verändern des Gebiets oder der Daten entsprechend aktualisiert und beim Zoomen angepasst.

Als Testgebiete des im Anhang befindlichen Prototypen sind Brandenburg und Potsdam-Mittelmark (südöstlich Berlins/Potsdams) realisiert. Zur genauen Lage der Gebiete vergleiche auch Abbildung 53.

4.3 Realisierung des Prototypen

Um den Anforderungen gerecht zu werden ist es besonders sinnvoll, mögliche Komponenten des Prototypen zu definieren und diese in geeigneter Art und Weise mit Funktionalität auszustatten. Dabei müssen die verschiedenen Komponenten klar voneinander abgegrenzt werden aber trotzdem miteinander harmonieren. Abbildung 39 zeigt die wichtigsten Komponenten der Anwendung, sowie deren Zusammenhänge/-spiel.

Abb. 39: Vereinfachter Funktionsablauf des Prototypen



Quelle: Eigene Darstellung.

Wie zu erkennen ist, bezieht die Anwendung ihre Informationen aus Geodaten-, Sachdaten- sowie Property-/Stylekomponenten. Diese "drei Komponentenimplementierung" macht den Prototypen besonders leicht konfigurierbar. Die Daten werden in wohlgeformten XML Dokumenten übermittelt und von Javascripten weiterverarbeitet. Diese, wie auch die CSS Datei, sind extern in das SVG Dokument eingebunden. Um den Anforderungen zeitgemäßer Programmierung gerecht zu werden, ist es unabdingbar, das Konzept der Objektorientierten Programmierung einzusetzen. Dabei werden die Objekte mit Methoden und Attributen in Javascriptroutinen zur Verfügung

gestellt und zur Laufzeit instanziiert. Wichtig ist, dass Objekte, bevor sie instanziiert werden, dem Javascript Interpreter bekannt sein müssen. Darum sind alle externen Javascriptdateien, die Objekte beinhalten, als erstes im SVG DOM eingebunden. Durch DOM Methoden in den Befehlsabfolgen der Funktionen können zur Laufzeit sowohl Attribute des SVG Baumes gelesen als auch dieser manipuliert werden. Geschieht letzteres, wird die neue oder veränderte graphische Ausprägung vom Client gerendert.

Abbildung 40 verdeutlicht eine kreierte SVG Datei (vereinfacht) einer thematischen Darstellung in der Startansicht mit einem thematischen Layer (hier Kreissektorendiagramme).

Abb. 40: Mögliche SVG Datei mit einem thematischen Layer

SVG DOM	Erklärung / Bedeutung
<code><?xml version="1.0" standalone="no"?></code>	XML Header
<code><!DOCTYPE svg ...></code>	DTD
<code><?xmlstylesheet ...></code>	CSS stylesheet
<code><svg id="viewer" ...></code>	äußerstes SVG Element (Root-Element)
<code><script .../></code>	Einbindung von Scripten
<code>...</code>	
<code><defs id="definitions"></code>	Definitionsteil
<code><symbol id="mann" .../></code>	Symboldefinition
<code><point id="p_01" x="540" y="929"/></code>	Basisdaten: Punktelement
<code>...</code>	
<code><g id="defs_0"></code>	Basisdaten: Gruppe für Polygone
<code><path id="01" d="M...z"/></code>	Pfadelemente - Polygone
<code>...</code>	
<code><rect id="0"></code>	bounding box der äußersten Kartenebene
<code><svg id="magnifier" ...></code>	SVG Element für Lupenwerkzeuge
<code><svg id="map" ...></code>	SVG Element für den Karteninhalt
<code><g id="map_0" ...></code>	Kartengruppe eines Levels
<code><use xlink:href="#01"/></code>	Instanzierung der Pfade
<code>...</code>	
<code><g id="Umsatz2002_0_pie" ...></code>	Diagrammgruppe
<code><g id="Umsatz2002_16" .../></code>	Gruppenelemente für die Diagramme
<code>...</code>	
<code><rect ...></code>	Rechteck als Kartenfensterrahmen
<code><svg ...></code>	Verschiebepfeiler
<code><g id="scaleBar" .../></code>	Maßstabspfeiler
<code><svg id="legend" ...></code>	SVG Element für die Legende
<code><g .../></code>	Gruppe für "Qualitätenlegende"
<code><text .../></code>	Legendentitel
<code><svg .../></code>	SVG Element für legendSize
<code><svg id="overview_0" ...></code>	SVG Element für Übersichtskarte
<code></code>	onclick Events
<code><g id="overview_0_mapGroup" ...></code>	Gruppe für Übersichtskarteninhalt
<code><use xlink:href="#01"/></code>	Instanzierung der Pfade
<code>...</code>	
<code><rect id="overview_0_rect" ...></code>	Rechteck
<code><text .../></code>	Titel der Übersichtskarte
<code><rect .../></code>	Rechteck zum Triggern des onclick Events
<code></code>	closeMapLayer Button
<code><g id="popUpMenu" .../></code>	Drop Down List

Quelle: Eigene Darstellung.

Hinweis: Für das Verständnis muss zudem darauf geachtet werden, die Javascript Objekte nicht mit den unter Umständen namentlich ähnlichen SVG Elementen zu verwechseln.

Im Folgenden werden die wichtigsten Knoten des SVG Baum's erläutert:

Grundsätzlich dient der Definitionsteil des SVG's zur clientseitigen Bereitstellung der Basisdaten und der Symbole. Alle dort deklarierten Elemente werden nicht gerendert.

Das `<svg>` Element mit dem Identifikationsschlüssel "map" ist das äußerste Element des Kartenfensters. Dieses beinhaltet das `<g>` Element der Kartenebene. Dessen Kinder sind `<use>` Elemente, die die Geometrien für die Kartengrundlage instanzieren. Letztes Kind des Kartenfensters ist ein `<g>` Element, dass alle Diagramme beinhaltet.

Ein weiteres `<svg>` Element dient als Containerelement für die Legendeninhalte. Zur Legendendarstellung der Qualitäten dient ein `<g>` Element. Danach folgt der Legendentitel. Der quantitative Teil der Legende ist in einem neuen `<svg>` Element zusammengefasst.

Auch Übersichtskarten werden mit `<svg>` Elementen dem Root-Element angefügt. Dieses beinhaltet alle für die Übersichtskarte notwendigen Elemente.

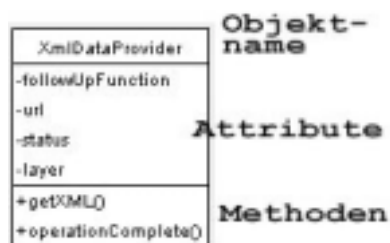
Basierend auf dem Prinzip des Äußeren beziehungsweise Inneren Designs (vgl. Kapitel 2.3.2) werden in den folgenden Kapiteln die verschiedenen Objekte des prototypischen Viewers erläutert. Zum Darstellen der Objekte dienen UML-ähnliche Klassendiagramme.

4.3.1 Äußeres Designmodul des Prototypen

4.3.1.1 Datenversorgung - Datenbereitstellung

Die Datenversorgung und die Bereitstellung dieser sind die zentralen Punkte des Prototypen. Die Datenübermittlung erfolgt durch Client-Server-Kommunikation. Ein Request wird durch die Instanzierung des `XMLDataProvider` Objekts ausgelöst. Die `getXML()` Methode dieses verwendet die `getURL()` Methode des `ASV3` zum Laden von Daten. Da `getURL()` **asynchron** arbeitet, dürfen nachfolgende Funktionen (hier `followUpFunction` genannt) erst nach vollständigem Parsen der geladenen XML-Fragmente aufgerufen werden ($\rightarrow$ **Parser**). Parser ist die `ASV3` interne `parseXML()` Methode. Diese gibt ein XML Fragment zurück. Nachdem die `operationComplete()` Methode das XML Dokument erfolgreich "geparst" hat, ruft diese die `followupFunction` auf. Als Parameter wird das "geparste" XML Fragment übergeben. Abbildung 41 zeigt das Objekt zur Datenversorgung mit seinen Attributen und Methoden.

Abb. 41: Das `XMLDataProvider` Objekt



Hinweis: Aufgrund fehlender Client-Server Möglichkeiten auf der Cd-Rom, sind ausgewählte Gebiete als Dateien abgespeichert. Diese könnten einfach durch eine **URL**, die beispielsweise ein Servlet aufruft, ersetzt werden. Ebenso bei Sachdaten, diese sind rein fiktiv und dienen nur der Anschaulichkeit

Quelle: Eigene Darstellung, vgl. Anhang `XmlDataProviderObject.js`.

Wie im vereinfachten Funktionsablauf aus Abbildung 39 zu erkennen ist, wird dieses Objekt zum Laden aller Daten benutzt. Einmal geladene Daten müssen nicht wiederholt transferiert werden. Diese bleiben der Anwendung erhalten.

4.3.1.1.1 Laden der Anwendungseinstellungen

Anwendungseinstellungen beschreiben das Layout, die Startansicht und Darstellungseigenschaften der Anwendung. Sie sind notwendig, um Konfigurierbarkeit des Prototypen zu gewährleisten. Dabei sendet der Server einen Einstellungscontainer (auch "Property Bag" genannt) an den Client. Dieser beinhaltet Daten in Form von XML Fragmenten. Abbildung 42 zeigt einen Auszug des Inhaltes einer Property Bag wie sie beim Starten der Anwendung aussehen könnte.

Abb. 42: Auszug aus der XML Property Bag

```

<defaultLayer>0</defaultLayer>
<defaultTheme>Umsatzentwicklung</defaultTheme>
<defaultChartType>stab</defaultChartType>
<map>
  <mapX>20</mapX>
  <mapY>150</mapY>
  <mapWidth>700</mapWidth>
  <mapHeight>525</mapHeight>
</map>

```

Defaulteinstellungen, die für das Laden der ersten Kartenansicht wichtig sind.

Kartenfenstereigenschaften, Platzierung und Größe des Kartenfensters

Quelle: Eigene Darstellung, vgl. Anhang *prop.xml*.

Hinweis: Neben der Definition der Startansicht werden beispielsweise auch Informationen zu den größtmöglichen Diagrammen in Abhängigkeit des Diagrammtypen bereitgestellt.

Eine Instanz des `xmlDataProvider` Objektes stellt alle Anwendungseinstellungen zur Verfügung, indem sie dem `MapSet` übergeben werden.

4.3.1.1.2 Laden der Basisdaten

Die Basisdaten sind alle topographischen Informationen der Anwendung. Die Basisdaten der Startansicht, im Falle des Prototypen die Bundesländer, sind bereits im SVG DOM eingebunden. Neue Basisdaten werden durch eine Instanz des `xmlDataProvider` Objektes nachgeladen und dem SVG Baum angefügt. Abbildung 40 zeigt, dass Basisdaten generell im Definitionsteil des SVG's zu finden sind.

Wichtig ist, dass die transferierten SVG Fragmente mit Basisdaten standardisiert vorliegen. Ansonsten kommt es zu Fehlinterpretationen durch die Javascriptroutinen. Abbildung 43 zeigt ein Beispiel eines Geodatenfragmentes einer Kartenebene.

Abb. 43: Vereinfachtes Beispiel eines spezifizierten XML Dokumentes

XML DOM	Erklärung
<code><?xml version="1.0" standalone="no"?></code>	XML Header
<code><g></code>	Gruppe des gesamten Layers
<code><point id="p_12051" x="740324" y="5812831"/></code>	Zentroide der Flächen
<code><g id="defs_12"></code>	Gruppe der Flächenobjekte
<code><path id="12051" d="M ... z"/></code>	Pfadelemente der Geoobjekte

Quelle: Eigene Darstellung, vgl. Anhang *12.xml*.

Zur Identifizierung der Geoobjekte wird das `id` Attribut verwendet. Im Falle des Prototypen setzen sich diese aus den jeweiligen Kennziffern der Gebiete zusammen. Da XML sinnvollerweise keine doppelten Bezeichner zulässt, erhalten Punkte das Präfix "p_" sowie Linien "l_". Flächen behalten den ursprünglichen Identifikationsschlüssel, sind jedoch in einer Gruppe zusammengefasst. Diese erhält den Identifizierungsschlüssel des übergeordneten Geoobjektes mit dem Präfix "defs_" (vgl. auch Abbildung 53 in Kapitel 4.2.2.3).

Hinweis: Zu Testzwecken diente ein MS Access Datenbanksystem. Die benötigten Daten sind mit Scriptroutinen aus dem MapInfo Austauschformat MID/MIF gelesen und als kommaseparierte Textdateien in die Access Datenbank importiert worden. Als Projektion und Koordinatensystem wurde UTM Zone 32 nördliche Hemisphäre gewählt.

4.3.1.1.3 Bereitstellen der Sachdaten zur Diagrammgenerierung

Die Sachdaten zur Diagrammgenerierung enthalten die thematischen Informationen. Diese lassen sich für den Prototypen hierarchisch zusammenfassen. Der Datencontainer, die Datensammlung und das Datenmodell bilden eine baumartige Objekthierarchie, wobei der Datencontainer die Wurzel ist. Ein Datenmodell beschreibt Werte eines Geoobjektes. Alle Datenmodelle einer Informationsschicht werden einer Datensammlung zugeordnet. Der Datencontainer beinhaltet alle Daten.

Anhand der thematischen Information werden Diagramme berechnet und gezeichnet. Eine Instanz des `XMLDataProvider` Objektes lädt die Sachdaten nach. Eine Datensammlung liegt in Form von XML Fragmenten der Anwendung vor (Abbildung 44). Diese sind in Abhängigkeit der verschiedenen Diagrammtypen zu spezifizieren.

Abb. 44: Vereinfachtes Beispiel eines spezifizierten XML Dokumentes für Sachdaten

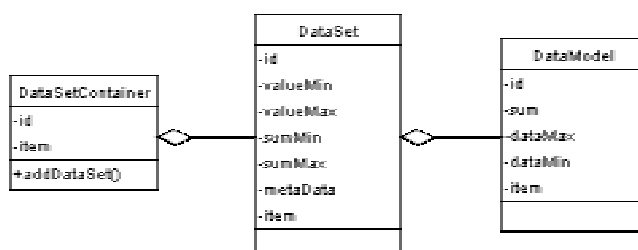
XML DOM	Erklärung
<code><?xml version="1.0" standalone="no"?></code>	XML Header
<code><dataSets></code>	root Element
<code><dataSet id="Umsatz2002_0"></code>	Datensammlung für einen Layer
<code><sumMin>51</sumMin></code>	minimale Summe
<code><sumMax>740</sumMax></code>	maximale Summe
<code><valueMax>300</valueMax></code>	absolut größter Wert
<code><metaData></code>	Metadaten für die Diagramme
<code><data></code>	Metadatenansammlung
<code><desc>Kartoffeln</desc></code>	Thema 1
<code><desc>&#196;pfel</desc></code>	Thema 2
<code><desc>Birnen</desc></code>	Thema 3
<code><desc>Tomaten</desc></code>	Thema 4
<code><desc>Bohnen</desc></code>	Thema 5
<code><dataModel id="16"></code>	Datenmodell eines Geoobjekts
<code><sum>740</sum></code>	Summe der Daten
<code><data></code>	Datensammlung
<code><value>300</value></code>	Wert 1
<code><value>50</value></code>	Wert 2
<code><value>130</value></code>	Wert 3
<code><value>60</value></code>	Wert 4
<code><value>200</value></code>	Wert 5
<code><dataModel ...></code>	weitere Datenmodelle
<code>...</code>	

Quelle: Eigene Darstellung, vgl. Anhang *Umsatz2002_0.xml*, *Umsatz2002_12.xml*, *Umsatz2002_12069.xml*.

Zusätzlich zu der Struktur des XML-Baumes spielen andere Faktoren, wie beispielsweise die Reihenfolge der Werte (`<value>`), eine große Rolle. Diese muss immer mit der Abfolge der Metadaten übereinstimmen. Ist das nicht der Fall, kommt es zu fehlerhaften Darstellungen in den Diagrammen. Im Beispiel aus der Abbildung 44 ist zudem notwendig, alle "dataModel's" nach ihrem Summenwert (`<sum>`) absteigend zu sortieren. Diese Ordnung legt die visuelle Anordnung der Diagramme im SVG DOM fest. Der kleinste Kreis muss im Vordergrund liegen (vgl. Kapitel 2.2.5.3).

Die `addDataSet()` Methode des `DataSetContainer` Objektes fügt diesem eine Instanz des `DataSet` Objektes zu. Der Datensammlung wird dabei ein entsprechendes XML Fragment als Parameter übergeben. Für die Werte der Datenreihe werden Attribute deklariert. Die Datenmodelle der Geoobjekte sind im `DataSet` Attributfeld `item` gesichert. Es entsteht ein dem XML ähnlicher Javascript Objektbaum. Abbildung 45 zeigt das JavaScript Objektmodell zur Bereitstellung thematischer Informationen.

Abb. 45: Das Objektmodell für Sachdaten



Quelle: Eigene Darstellung, vgl. Anhang *Data.js*.

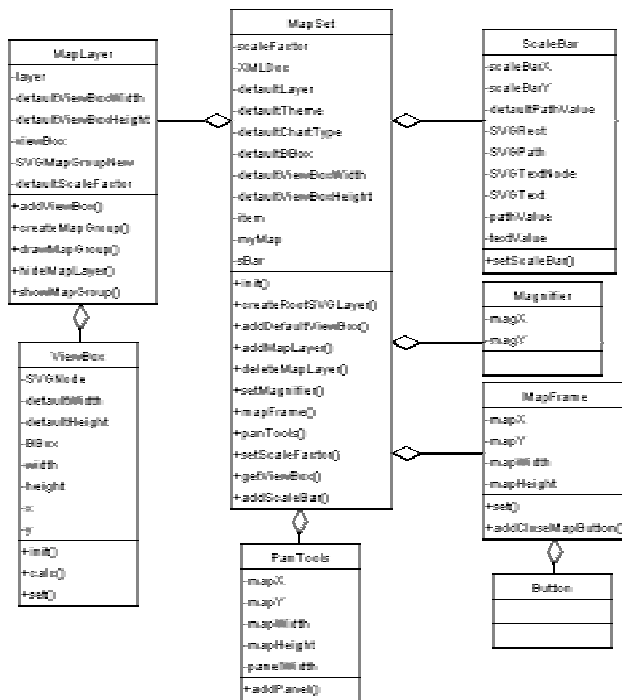
Hinweis: Obwohl beim Zeichnen der Diagramme auch direkt auf das XML Fragment zugegriffen werden kann, wurde dieser Zwischenschritt durch schnellere Javascript Objekte gewählt.

Die globale Variable `dsc` deklariert den Datencontainer der Anwendung.

4.3.1.2 MapSet

Ein MapSet speichert Eigenschaften der Anwendung und kreiert durch seine Methoden alle für die Kartenansicht wichtigen Bestandteile. Dazu gehören die Werkzeuge zur Navigation, der Maßstabsbalken sowie der Kartenrand. Im Prototyp ist dafür das MapSet Objekt mit dessen Attributen beziehungsweise Methoden vorgesehen. Abbildung 46 gibt einen Überblick darüber und zeigt alle mit dem MapSet verbundenen Objekte.

Abb. 46: Darstellung der mit dem MapSet verbundenen Javascript Objekte



Quelle: Eigene Darstellung, vgl. Anhang *MapSet.js*.

Wie in der Abbildung zu erkennen ist, sind weitere Methoden vorhanden, die der Kartenansicht Rahmenfunktionalität geben. Dazu zählen die `addMapLayer()` Methode, die ein neues MapLayer Objekt instanziiert (vgl. Kapitel 4.2.2.1) und dem Attributfeld `item` hinzufügt. Zudem regelt die `setScaleFactor()` Methode den globalen "Scalefactor" der Karte, der zum Berechnen der Zoomstufen von Bedeutung ist. Das MapSet ist in einer globalen Variable deklariert, damit dessen Methoden und Attribute der Anwendung zur Verfügung stehen.

4.3.1.2.1 Der interaktive Kartenrahmen

Als den Kartenrahmen bezeichnet man die Umrandung des Kartenfensters. Er beinhaltet einen graphischen Rahmen und einen Button zum Roll-Up. Im Prototyp steht zur Erstellung des Kartenrahmens das MapFrame Objekt zur Verfügung (vgl. Anhang *MapFrame.js*). Durch die `addCloseMapButton()` Methode wird ein eigenes Button Objekt instanziiert. Die Geometrie des Buttons ist im Definitionsteil des SVG DOM's vorhanden. Das Button Objekt fügt dem SVG Baum ein `<use>` Element zu, das den Button instanziiert und platziert (vgl. Anhang *Button.js*).

4.3.1.2.2 Räumliche Navigationselemente zur Kartentranslation

Räumliche Navigationselemente zur Kartentranslation dienen zum Verschieben der Kartenansicht. Das PanTools Objekt mit dessen `addPanel()` Methode erstellt die interaktiven Balken zum Verschieben der Kartenansicht und fügt diese dem SVG DOM an (vgl. Anhang *PanTools.js*).

4.3.1.2.3 Maßstabsbalken

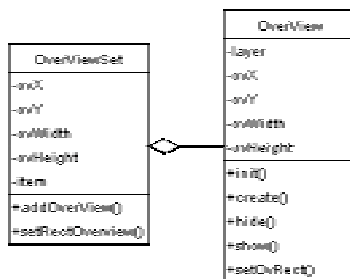
Damit der Anwender den Bezug zur Realität behält, ist dem Prototypen ein Maßstabsbalken angefügt worden. Dieser zeigt bei konstanter Länge die reale Distanz auf gerade Kilometerwerte gerundet.

Das MapSet sieht für eine Maßstabsbalkenimplementierung das `scaleBar` Objekt vor (vgl. Abbildung 46). Es kreiert den neuen Maßstabsbalken. Die `setScaleBar()` Methode fügt diesem die reale Distanz zu. Weiter ermöglicht sie es, beim Skalieren der Kartenansicht diesen Wert zu aktualisieren (vgl. Anhang *ScaleBar.js*).

4.3.1.3 Übersichtskarten zur räumlichen Orientierung

Die Übersichtskarten unterstützen den Anwender bei der räumlichen Orientierung. Das Drill-Down Mapping des Prototypen erfordert eine besonders sinnvolle Übersichtskartenimplementierung. Ähnlich dem Programmierprinzip bei Datensammlungen, gibt es ein `overviewSet` Objekt, das als Sammlung aller Übersichten gilt (Abbildung 47).

Abb. 47: Die Objekte der Übersichtskarten



Quelle: Eigene Darstellung, vgl. Anhang *Overview.js*.

Wird eine neue Kartenansicht erzeugt, fügt die `addOverview()` Methode der Übersichtskartensammlung eine neue Übersicht hinzu. Diese instanziert ein `overview` Objekt. Dessen `create()` Methode kreiert eine Übersichtskarte und fügt diese dem SVG Baum an. Dabei dienen die im Definitionsteil des SVG Baumes deklarierten Basisdaten als Kartengrundlage. Weitere Methoden des `overview` Objektes sind `hide()` und `show()`, die das Verschwinden und das Erscheinen der Übersicht ermöglichen. Die Methode `setOverviewRect()` dient zum Skalieren des Kartenausschnittsrechteckes auf die Originalgröße einer Kartenansicht (vgl. Anhang *Overview.js*).

Beim räumlichen Navigieren passt die `setRectOverview()` Methode der Übersichtskartensammlung das übergebene Kartenausschnittsrechteck an.

Die Übersichtskarten sind zudem Akteure, die Ereignisse zur räumlichen Navigation ausführen (vgl. Abschnitt 4.3.1.6).

4.3.1.4 Containerelement für Legenden

Die Legende erklärt Karteninhalte und ist deswegen unabdingbar. Im SVG DOM ist beim ersten Laden kein Containerelement für Legenden vorgesehen. Um ein derartiges in Form eines `<svg>` Elementes zu erzeugen, dient das `Legend` Objekt (Abbildung 48).

Abb. 48: Das Legend Objekt

Legend
-leg%
-leg%
-legWidth
-legHeight
+createSVGLegend()
+scale()
+setLegendSizeAndViewBox()

Quelle: Eigene Darstellung, vgl. Anhang *Legend.svg*.

Die `createSVGLegend()` Methode erstellt das `<svg>` Element für Legenden. Es besitzt weitere Methoden `scale()` und `setLegendSizeAndViewBox()`, die beim Zoomen die Legende transformieren. Im Gegensatz zu den Übersichtskarten gibt es im Objektbaum des SVG's immer nur eine Legende. Die Legendeninhalte werden von den Methoden des `chartSet` Objekts erzeugt und auch von diesen wieder gelöscht (vgl. Kapitel 4.3.2.2.2).

4.3.1.5 Drop-Down Liste zur Diagrammauswahl

Eine Drop-Down Liste ist ein Auswahlelement einer Benutzeroberfläche. Bedient der Anwender diese, klappt das Auswahlménü auf. In der Startansicht der Anwendung besteht die Möglichkeit, verschiedene Diagrammtypen auszuwählen, die vom Diagrammgenerator gezeichnet werden. Die Drop Down Liste zeigt auch einen möglichen **Drill-Across** auf. Verschiedene Diagrammtypen erfordern gegebenenfalls neue Sachdaten, die nachgeladen werden.

Eine Drop-Down Liste wird durch die Instanz des `DropDownList` Objektes erzeugt (Abbildung 49).

Abb. 49: Das DropDownList Objekt

Drop Down List
-item
-x
-y
-width
-height
-defaultValue
-SVGMenuGroup
+hide()
+show()

Quelle: Eigene Darstellung, vgl. Anhang *DropDownList.js*.

Hinweis: Die Drop Down Liste heißt im Quellcode teilweise auch `PopupMenu`.

Die `show()` und `hide()` Methoden dienen zum Zeigen und Verstecken der Liste beim Drill-Down Mapping (vgl. Anhang *DropDownList.js*).

4.3.1.6 Interaktionsmöglichkeiten der kartographischen Benutzeroberfläche

Die Interaktion zwischen dem Nutzer und der Anwendung ist unverzichtbarer Bestandteil einer Benutzeroberfläche eines Webmapping-Systems beziehungsweise WebGIS. Realisiert wird Interaktion durch Ereignisse (*engl.* Events), die die SVG Spezifikation zur Verfügung stellt. Sogenannte Eventhandler überwachen dabei, ob ein Ereignis ausgelöst wird. Das auslösende Element bezeichnet man als Akteur. Ihm ist der entsprechende Event mit auszulösender Funktion zugewiesen. Events werden aber nicht nur durch Mausaktivitäten ausgelöst, sondern beispielsweise auch bei der Initialisierung der Anwendung. Abbildung 50 gibt eine Übersicht aller kartographisch relevanten Interaktionsmöglichkeiten des Prototypen.

Abb. 50: Übersicht der kartographischen Interaktionsmöglichkeiten

Vorgang	Ereignis	Funktion	Akteur
Initialisierung der Anwendung	onload	initViewer()	äußerstes SVG Element
Skalierung der Kartenansicht	onclick	zoom()	Lupen
Verschieben der Karte	onclick	pan()	Verschiebepalken
Drill-Down Mapping	onclick	zoomToBox()	Kartenobjekte
Originale Kartenansicht wiederherstellen	onclick	thisView()	aktuelle Übersichtskarte
Roll-Up	onclick	thisView()	übergeordnete Übersichtskarte
Roll-Up	onclick	closeMapLayer()	Button im Kartenfenster
Diagrammtyp ändern	onclick	selection()	Drop-Down Liste

Quelle: Eigene Darstellung, vgl. Anhang *events_ "Funktion".js*.

Hinweis: Die Akteure eines Events können noch weitere User Interface Events besitzen, die sind aber im kartographischen Kontext nicht von großer Bedeutung und dienen nur als Stilmittel.

Zusätzlich sind noch die Events der Drop Down Liste zu erwähnen, die deren Auf- und Zuklappen realisieren (vgl. Anhang *event_dropdownlist.js*).

4.3.1.7 Formatvorlagen zur individuellen Stilisierung

Die Formatvorlagen dienen zur Stilisierung der geometrischen SVG Elemente. Im Prototypen werden sie global in einem externen Cascading Style Sheets Dokument deklariert. Generell machen die Formatvorlagen Sinn, wenn ein "Style" mehrere Male verwendet wird. Weiter tragen sie zur einfachen Konfigurierbarkeit der Anwendung bei. Die Vorlagen dienen nicht nur dem äußeren Design, sondern stellen auch "Styles" für den Karteninhalt zur Verfügung (vgl. Anhang, *style.css*).

4.3.2 Inneres Designmodul des Prototypen

Das Innere Designmodul, auch Kartenkomponente genannt, behandelt alle für die Kartengraphik wichtigen Module.

4.3.2.1 Kartenebene

Die Kartenebenen drücken alle kartographischen Bestandteile einer Kartenansicht aus. Im SVG Baum wird eine Kartengruppe durch ein Containerelement, hier durch ein `<g>` Element, realisiert. Die Inhalte einer Kartenebene sind dessen Kinder (vgl. dazu Abbildung 40). Abbildung 46 zeigt das `MapLayer` Objekt als Bestandteil des `MapSet`'s (vgl. Kapitel 4.2.1.2). Die `createMapGroup()` Methode erzeugt das Containerelement dieser Kartenebene und instanziert die nötigen Basisdaten des SVG Definitionsteils. `drawMapGroup()` stellt die Karte mit zugehöriger Legende und Übersichtskarte dar. Die Methoden `hideMapLayer()` und `showMapLayer()` verstecken und zeigen eine entsprechende Kartenebene (vgl. Anhang *MapLayer.js*).

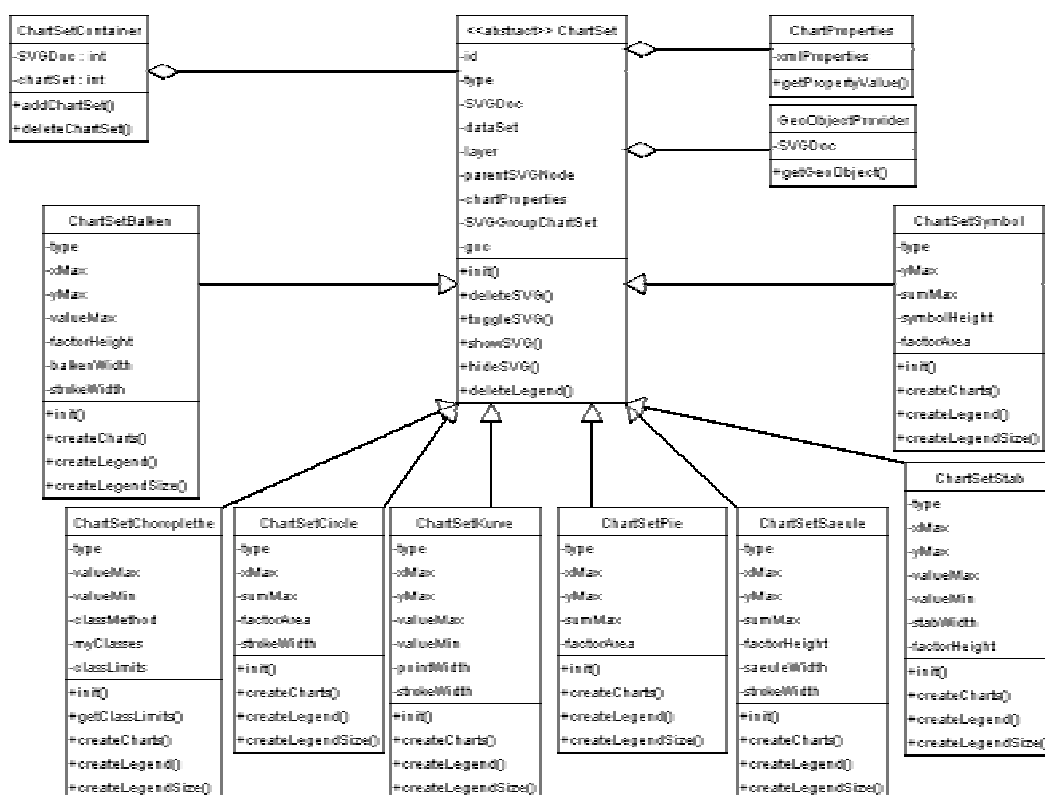
Mit der `addViewBox()` Methode wird ein neues `viewbox` Objekt instanziiert. Dessen Methode `calc()` berechnet und speichert die originalen Viewboxwerte einer Kartenebene. Zur Wiederherstellung der originalen Kartenansicht eines Layer's dient die `set()` Methode (vgl. Anhang, *viewbox.js*).

4.3.2.2 Der Diagrammgenerator

Das Herzstück der Kartenkomponente ist der Diagrammgenerator, bestehend aus verschiedenen Objekten mit Methoden zum Berechnen sowie Zeichnen von Diagrammen und Legende. Zudem lesen weitere Hilfsobjekte die Diagrammeigenschaften aus den Anwendungseinstellungen. Zur

Sammlung aller Diagrammlayer dient eine global deklarierte `chartSetContainer` Instanz. Ein Diagrammlayer beinhaltet alle Diagramme des Themas einer Kartenebene und wird durch den jeweiligen Diagrammtypen, allgemein als `chartSetType` bezeichnet, ausgedrückt.

Abb. 51: Die Objekte des Diagrammgenerators



Quelle: Eigene Darstellung, vgl. Anhang *Chart.js*.

4.3.2.2.1 ChartSetContainer

Das `chartSet` Attribut des `chartSetContainer` Objekts enthält alle `chartSet` Objekte, die durch die `addChartSet()` Methode angefügt werden. Es dient als Container aller Diagrammsammlungen (`chartSet`) (vgl. Anhang *Chart.js*).

4.3.2.2.2 ChartSet

Ein `ChartSet` ist eine Sammlung aller Diagramme eines Themas für eine Kartenebene. Die verschiedenen `chartSetType` Objekte sind Unterklassen des `chartSet` Objektes; man spricht von Vererbung. Die Unterklassen erben die Methoden und Attribute der zugehörigen Oberklasse.

Ein `chartset` Objekt besitzt verschiedene Methoden zum Ein- und Ausschalten der Diagramme einer Kartenebene im SVG. Das `goc` Attribut instanziiert ein neues `GeoObjectProvider` Objekt welches dem `ChartSet` Geodaten zur Verfügung stellt. Das `chartProperties` Attribut instanziiert ein neues `ChartProperties` Objekt, das für die Referenzierung der Zeicheneigenschaften der Anwendungseinstellungen zuständig ist.

Die Objekte aller Diagrammtypen sehen strukturell ähnlich aus. Sie besitzen alle die Methoden `createCharts()`, `createLegend()` und `createLegendSize()`. Diese berechnen und erzeugen die Diagramme und die zugehörige Legende. Zusätzlich sind für manche Diagrammtypen weitere Hilfsmethoden nötig. Choroplethenkarten erfordern zum Beispiel eine Berechnung der Klassengrenzen. Derzeit ist die Klassifizierung nach gleichen Werteintervallen implementiert. Die Attribute der verschiedenen Diagrammmethoden variieren aufgrund der unterschiedlichen Werte- sowie Zeicheneigenschaften. Abbildung 52 zeigt diese.

Abb. 52: Diagrammtypen und der Eigenschaften

Diagramm- typ	Wertebereich	Werteigen- schaften	Zeicheneigen- schaften
Kreis	$]0;\infty[$, Nullwerte werden nicht "gezeichnet".	sumMax	xMax oder yMax
Balken	$]0;\infty[$, Nullwerte werden nicht "gezeichnet".	sumMax, valueMax	xMax, yMax
Säulen	$]0;\infty[$, Nullwerte werden nicht "gezeichnet".	sumMax	xMax, yMax
Stab	$] -\infty;\infty[$, Nullwerte werden "gezeichnet".	valueMax, valueMin	xMax, yMax
Kurven	$] -\infty;\infty[$, Nullwerte werden "gezeichnet".	valueMax, valueMin	xMax, yMax
Kreis	$]0;\infty[$	sumMax	xMax oder yMax
Symbolfigur	$]0;\infty[$	sumMax	symbolHeight, xMax, yMax
Choroplethe	$] -\infty;\infty[$, Nullwerte werden "gezeichnet".	valueMax, valueMin	classMethod, myClasses

Quelle: Eigene Darstellung, vgl. Anhang `cs_ "ChartsetType".js`.

Hinweise zu den Begriffen: SumMax entspricht der Maximalsumme der Werte eines DataModels in einem DatenSet, sumMin die Minimalsumme. valueMax ist der absolut größte Wert eines DataSets, valueMin der kleinste. xMax/yMax sind die größtmöglichen Ausmaße eines Diagramms. symbolHeight beschreibt die Höhe des verwendeten SVG Symbols, classMethod die Klassifizierungsmethode bei Choroplethenkarten; myClasses entspricht der Anzahl der Klassen.

Der Kartenebene im SVG DOM ("mapGroup") wird ein Diagrammlayer als Kind angehängt.

4.3.2.2.3 Diagrammeigenschaften

Die Diagrammeigenschaften sind in den Anwendungseinstellungen gespeicherte Attribute zur Diagrammberechnung. Dazu zählt beispielsweise der größtmögliche Radius für Kreissektoren.

Um die Diagrammeigenschaften in der Propertybag zu referenzieren, ist das `chartProperties` Objekt vorhanden (vgl. Abbildung 51). Dieses enthält das Attribut `xmlProperties`, welches das zugehörige XML Fragment enthält. Die Methode `getPropertyValue()` werden als Parameter eine Eigenschaft übergeben, die diese aus dem XML Fragment liest und zurückliefert (vgl. Anhang `Chart.js`).

4.3.2.3 Bereitstellung der Basisdaten

Basisdaten werden vom Definitionsteil des SVG Dokumentes bereitgestellt (vgl. Kapitel 4.1.2.2.1). Sie bestehen aus Punkten und Polygonen. Linien spielen bisher keine Rolle, da streckenbezogene Diagramme noch nicht implementiert sind.

Der `GeoObjectProvider` mit seiner `getGeoObject()` Methode ist dafür zuständig, ein Geoobjekt im SVG DOM zu referenzieren, die nötigen Daten zu selektieren und dem Diagrammgenerator zur Verfügung zu stellen. Das `GeoObjectProvider` Objekt ist in der Abbildung 51 graphisch abgebildet (vgl. auch Anhang `Chart.js`). Damit keine Fehlinterpretationen auftreten, werden Basisdaten wie in Abbildung 53 spezifiziert.

Abb. 53: Format der SVG Basisdatenfragmente

GeoObjekt	Beispiel eines zugehörigen XML Fragments
Punkt	<code><point id="p_12" x="123456789" y="123456789"/></code>
Linie	<code><path id="l_12" d="M1000 1000 156789 34567 1543245 197686"/></code>
Fläche	<code><path id="12" d="M1000 1000 156789 34567 1543245 197686z"/></code>

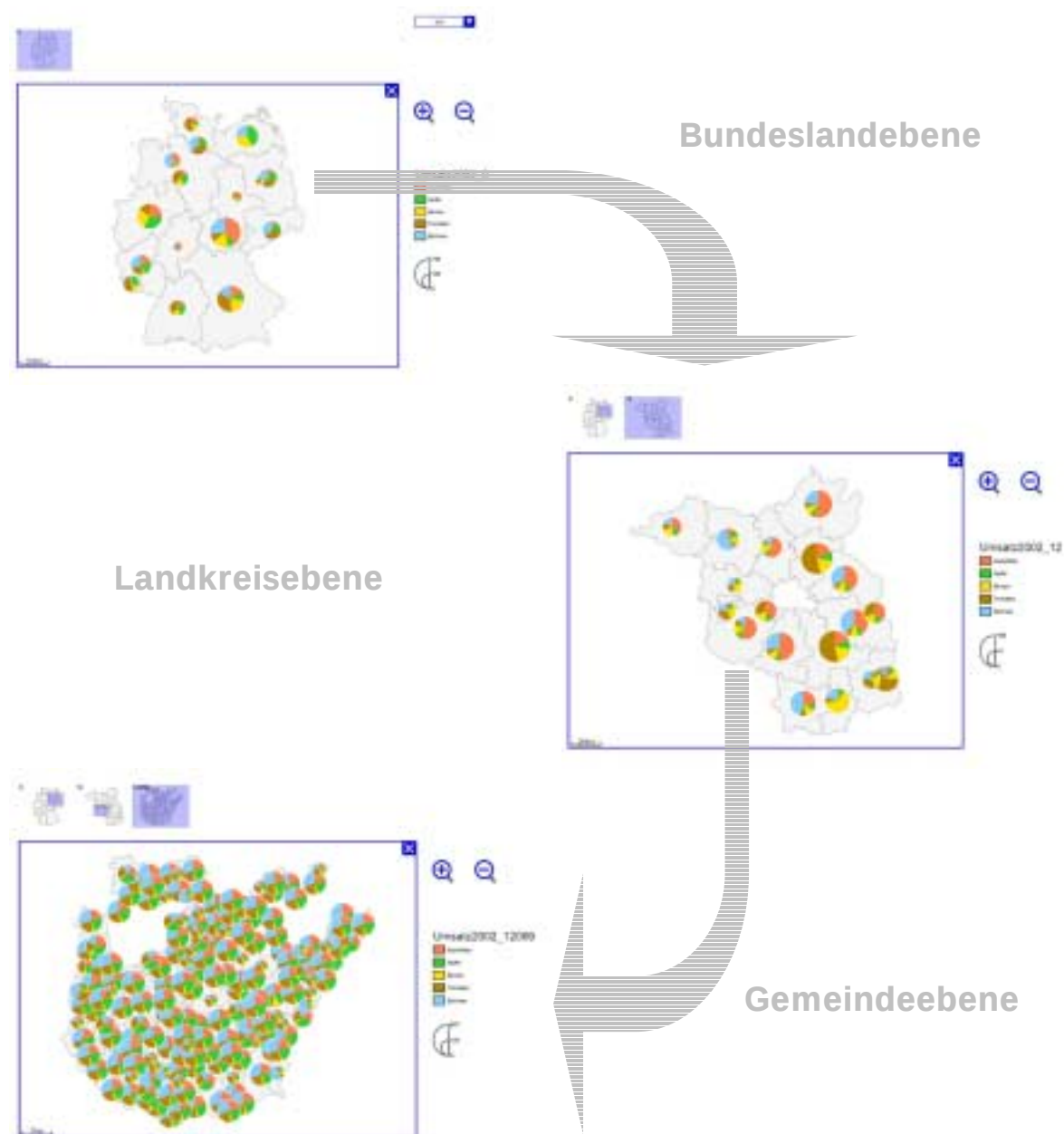
Quelle: Eigene Darstellung.

Hinweis: Die im Prototypen vorgestellte Realisierung ist ungültiges XML, da das `<point>` Element mit Attributen nicht in der DTD deklariert ist, obgleich eine Erweiterung um dieses kein Problem darstellt. Die Funktionalität schränkt diese Tatsache nicht ein.

4.3.3 Gesamtansichten des Prototypen

Abbildung 53 zeigt drei Gesamtansichten des Prototypen. Die Thematik ist mit Kreissektorendiagrammen dargestellt. Mit dieser hierarchischen Abfolge wird das Drill-Down Mapping erläutert. Die Gebietsaufteilung erfolgt administrativ. Die Gebiete der Abbildung entsprechen den Testgebieten des im Anhang beigefügten Prototypen.

Abb. 54: Drill-Down Mapping mit dem Prototypen



Quelle: Eigene Darstellung.

4.4 Bewertung des Prototypen

Die Bewertung gibt konkrete Beispiele, in welchen Punkten sich der realisierte Prototyp für einen entsprechenden kartographischen Viewer noch nicht eignet. Dazu zählen sowohl Probleme im

Umgang mit SVG, also auch kartographische Mängel des Prototypen. Im Folgenden eine Übersicht der aufgetretenen Schwierigkeiten.

Zum einen sind allgemeine Faktoren zu nennen, die besonders beim praktischen Einsatz von SVG auftreten. Gegenwärtig gibt es noch zu wenige Erzeugungs- beziehungsweise Editiermöglichkeiten. Das macht SVG fast ausschließlich nur für Experten zugänglich. Entwicklungstools mit grundlegenden Features, wie Javascript Syntaxüberprüfung fehlen. Weiter gibt es noch erheblichen Nachholbedarf bei der Wahl des User Agent's. Die einzige augenblicklich einsetzbare Viewer Implementierung, ist Microsoft's Internet Explorer mit installiertem Adobe SVG Viewer 3. Die Browserabhängigkeit sowie die Installation des mächtigen Plug-In's tragen nicht zur weiten Verbreitung des Standards bei. Der softwareseitige Zugang fehlt deswegen vielen Anwendern. Ein weiteres Problem ist die Offenheit des Quellcodes. Die Einsicht des Quellcodes ist vom großer Bedeutung, da jeder, der die Anwendung startet, darauf Zugriff hat. Besonders bei umfangreichen clientseitigem Scripting macht sich diese Tatsache negativ bemerkbar. Die "Intelligenz" einer Anwendung kann von jedem nachvollzogen werden. Auch die Möglichkeit, den erzeugten SVG Code auf einfache Art und Weise zu interpretieren, stellt sich bei sensiblen Daten als nachteilig heraus.

Zum anderen treten Probleme kartographischen Ursprungs auf. Bei der automatischen Kartenerzeugung aus Vektordaten treten schon seit langem Komplikationen auf. Viele dieser sind tiefgreifender Art und stellen auch Informatiker vor schwierige Aufgaben. Beispielsweise ist die automatische Platzierung von Schrift noch in fast keiner Anwendung sinnvoll gelöst und daher in den Karten des Prototypen darauf verzichtet worden. Ein ähnliches Problem ist die Platzierung von flächentreuen Diagrammen, wobei hier die Berechnung des Referenzpunktes von großer Bedeutung ist. Im Prototypen kann bei Bundesländern wie Niedersachsen sofort ein Konflikt erkannt werden. Auch die Darstellung von Bewegung durch gerichtete Banddiagrammen ist in dessen Anfang- und Endpunkt nicht auf einfache Art und Weise zufriedenstellend zu lösen.

Andere Probleme resultieren aus zu primitiver Realisierung des Prototypen. Diese können aber mit "einfachen" Mitteln behoben werden.

Zu einen tritt ein Problem bei der Diagrammskalierung auf den verschiedenen Ebenen eines Drill-Downs auf. Gegenwärtig gibt die Property Bag einen Maximalwert vor, der für einen Diagrammtypen auf allen Layern gilt. Anhand des globalen Skalierungsfaktors wird aus diesem die Maximalgröße für die jeweilige Ebene berechnet. Der Darstellungskonflikt entsteht aufgrund unterschiedlicher Objektdichte auf verschiedenen Layern. Wie Abbildung 54 zeigt, besitzt die Kartenansicht Brandenburg weniger Landkreise als die Kartenansicht Potsdam-Mittelmark Gemeinden. Die Diagrammdichte auf Gemeindeebene ist deswegen höher. Demzufolge sind diese zu groß skaliert. Deswegen sollte für jede Kartenebene die notwendigen Eigenschaften zur Diagrammgröße in Form einer ebenenspezifischen Property Bag nachgeladen werden.

Zum anderen tritt ein Problem bei der Darstellung der Nachbargebiete auf. Derzeit sind alle Karten sogenannte Inselkarten, in denen nur die gewünschte Region dargestellt wird. Eine Einbeziehung der angrenzenden Regionen ist für eine optimiertere räumliche Navigation ratsam. Ein räumlicher Drill-Across wäre dann auch gegeben.

Weitere Nachlässigkeiten ist das Fehlen einer konkreten Standortkartenimplementierung mit zugehörigen Events. Beispielsweise könnte eine Karte implementiert werden, die Filialstandorte anzeigt. Beim Überfahren der zugehörigen Symbole wird eine Art Filialsteckbrief angezeigt. Zudem ist eine bessere Identifizierung der Geoobjekte notwendig. Bis jetzt kann man nur erraten, welche Gebiete dargestellt werden. Eine derartige Erweiterung kann beispielsweise durch Tool-Tips gelöst werden.

4.5 Verbesserungsvorschläge für den Prototypen

Um aus dem Prototypen einen Viewer zu realisieren, bedarf es noch einiger Verbesserungen. Zuerst sind die im vorherigen Kapitel angesprochene Schwierigkeiten zu verbessern. Weiter lassen sich allgemeine Korrekturen zur Erarbeitung einer Viewerbasis und Vorschläge zur kartographischen Aufwertung unterscheiden. Im Folgenden dies wichtigsten Vorschläge zur Aufwertung der Anwendung.

Vorschläge zur Erarbeitung einer Viewerbasis:

Folgende allgemeine Gedanken optimieren beziehungsweise erweitern den Prototypen zu einem Basisprodukt. Dieses dient als Grundstock für eine individuelle Anpassung der Anwendung.

Für die Anwendung muss ein grundlegendes Konzept für Metadaten entwickelt werden. Metadaten sind zum Beispiel namentliche Bezeichnungen der Geoobjekte oder weitere Informationen zu Standorten. Diese dient als Grundlage für die Implementierung verschiedener Events zum Anzeigen von zusätzlichen Informationen.

Außerdem ist ein Plan zur automatischen Auswahl des besten Kartentypen für eine Datenreihe notwendig. Dafür müssen Sachdaten für Geomarketing mit Attributen versehen werden, die eine Aussage über den geeigneten Darstellungstypen treffen. Dies muss anhand der in Abschnitt 2.2.3.2 besprochenen Eigenschaften von Informationen erfolgen. Daten werden deswegen nach ihrer Implantation und ihres Skalenniveaus attribuiert. Weiter ist zwischen absoluten und relativen Werten zu unterscheiden. Die zur Darstellung notwendigen graphischen Variablen sind nun vorgegeben. Weiter muss der Darstellungstyp attribuiert werden.

Der Anwender ist deswegen nicht auf kartographisches Know-how angewiesen. Trotzdem darf ein fortgeschrittener Benutzer nicht dieser Darstellungsmethode ausgeliefert sein. Individuelle Anpassung muss gewährleistet werden.

Die Benutzerschnittstelle des Prototypen muss um weitere Elemente, wie beispielsweise einem Statusbalken, erweitert werden. Dieser zeigt den Downloadstatus beim Nachladen von Daten an. Während dessen ist es zudem sinnvoll, alle Events durch `pointer-events="off"` zu unterdrücken.

Außerdem müssen die Javascriptroutinen der Anwendung weiter optimiert werden. Es gilt, beste Konfigurierbarkeit zu erreichen. Verschiedene Objekte und Methoden bedarf es der Umstrukturierung, um bessere Scripting Performance zu erreichen.

Für jedes Softwareanwendung ist die Implementierung einer Aussagekräftigen Hilfe notwendig. Diese muss den Anwender unterstützen und gegebenenfalls weiterführende Erklärungen liefern.

Weitere Vorschläge zur kartographischen Aufwertung:

Von großer Bedeutung ist die Implementierung weiterer Klassifizierungsmethoden für Choroplethenkarten. Als "default" ist eine Klassifizierung nach Quantilen allgemein gebräuchlich. Zusätzlich können individuelle Klassifizierungsmöglichkeiten ins Auge gefasst werden (vgl. Kapitel 2.2.5.4). Zum Beispiel ermöglicht ein Schieberegler, dass die Klassengrenzen individuell eingestellt werden können.

Die Implementierung weiterer kartographischer Elemente wie streckenbezogene Diagrammkarten muss angedacht werden. Zusätzlich können eine Integration für Business Intelligence übliche Darstellungsformen den Viewer umfangreicher gestalten. Dazu zählen beispielsweise Tachometerdiagramme die auch unter dem Begriff "Cockpit" bekannt sind.

Analog zur Berücksichtigung des größten Diagrammausmaßes ist die Einbeziehung eines Größenwertes für minimal erkennbare Kartenelemente anzudenken. Die Einschließung derartiger Werte in die Anwendungseigenschaften vermeidet zu kleine Diagrammdarstellungen.

Bezüglich der räumlichen Navigation sollten weitere, intuitive Zoommöglichkeiten berücksichtigt werden. Zoomen auf die Größe eines mit der Maus aufgezogenen Rechtecks wäre eine allgemein Anerkannte. Diese ist in heutigen kartographischen Anwendungen der Standard.

Die Maßstabsleiste muss immer "runde" Zahlen darstellen. Bei Zahlenrundungen ist auch die Skalierung des Maßstabsbalkens notwendig.

Um die Benutzeroberfläche individueller gestaltbar zu machen, kann ein skalierbares Kartenfenster realisiert werden. Der Anwender ist somit in der Lage, die Größe der Karte nach eigenen Bedürfnissen zu bestimmen.

Die derzeitige Implementierung erlaubt unendliches Verschieben einer Karte. Dieses ist in einem ausgereiften Viewer zu unterdrücken.

Weiter ist über eine OGC konforme Geodatenversorgung nachzudenken. Diese könnte in Form der vom OGC spezifizierten XML Anwendung GML ablaufen. Durch eine XSL Transformation der GML spezifizierten Informationen kann eine SVG Graphik erzeugt werden (vgl. Kapitel 3.1.2.6). Dadurch würden bestimmte serverseitige Zwischenschritte zur Generierung von SVG Fragmenten wegfallen.

5 Zusammenfassung der Arbeit

Das Ziel dieser Arbeit war es, die Theorie zur Erstellung einer browserbasierten Anwendung für thematische Karten des Geomarketings aufzuarbeiten. Anhand eines Prototypen sollten die theoretischen Aspekte mit XML-basierten Technologien getestet werden. Die Ergebnisse werden in den nächsten Abschnitten noch mal zusammengefasst.

Die Darstellungen des Geomarketing beinhalten raumbezogene statistische Zahlen, die anhand thematischer Karten visualisiert werden. Moderne Kartenherstellung in Form von Webmapping und WebGIS stehen bei der Erzeugung im Mittelpunkt. Die Information kann in ihrer räumlichen Ausprägung punkt-, strecken-, oder flächenbezogen vorliegen. Die statistischen Eigenschaften der Werte werden in nominal-, ordinal- und metrische Skalen eingeteilt. Basierend auf diesen Attributen werden Darstellungsmöglichkeiten in Form der visuellen Variablen BERTIN's eingegrenzt. Sie spielen für die Gestaltung der Karteninhalte eine entscheidende Rolle. Die Kartengrundtypen des Geomarketing sind Standort-, Verbreitungs-, Diagramm-, Choroplethen- und Dynamische Karten. Die Kartenelemente sind vorwiegend Diagramme. Weiter spielen die Klassifizierungsmethoden für Choroplethenkarten für das Aussehen diese Kartentypen eine entscheidende Rolle. Auch ergänzende Elemente, wie eine Legende und eine Übersichtskarte sind von großer Bedeutung, um möglichst aussagekräftige Entscheidungsgrundlagen zu erstellen.

Zudem müssen die Faktoren des Ausgabemediums Bildschirm berücksichtigt werden. Divergierende Soft- und Hardwarestrukturen bei Anwendern führen zu unterschiedlichen Darstellungen. Die Grundregeln, wie die geringere Auflösung und Größe eines Bildschirms im Vergleich zu Papierkarten, sind bei der Kartengestaltung zu beachten. Die zusätzlichen Gestaltungsmöglichkeiten eines kartographischen Viewers lassen sich in inneres und äußeres Kartendesign einteilen. Dabei werden die Karteninhalte (inneres Design) von der Viewerfunktionalität (äußeres Design) abgegrenzt. Besonders Karteninhalte als Bedienelemente und animierte Karten bilden die wichtigsten Parameter des inneren Designs. Die Räumliche Navigation sowie eine Layerkontrolle zählen zu den wichtigsten Funktionalitäten des äußeren Kartendesigns. Zudem muss ein Viewer über eine funktionale und anwenderfreundliche Benutzeroberfläche verfügen.

Für die Praktische Umsetzung dienen XML-basierte Technologien. Diese stehen für zeitgemäße standardisierte Webentwicklung. Das für die Visualisierung eingesetzte XML-basierte Vektorformat SVG stellt umfangreiche Möglichkeiten zur kartographischen Darstellung zur Verfügung. Alle wichtigen Elemente einer Karte können mit Javascript zur Laufzeit auf dem Client erzeugt werden. Auch komplexe Interaktionen sind möglich, jedoch nur durch umfangreiche Scripting-routinen. Jedoch fehlt es noch an verschiedenen SVG Implementierungen. Gegenwärtig kann die volle Stärke von SVG nur mit dem aktuellen Internet Explorer von Microsoft mit installiertem Adobe SVG Viewer 3 Plug-In genutzt werden. Jedoch eignen sich XML-basierte Technologien für eine einfache Systemintegration. Da XML textbasiert ist, lassen sich entsprechende Dateien mit verschiedenen Serverimplementierungen erzeugen.

Der erstellte Prototyp zeigt, dass XML-basierte Technologien geeignete Möglichkeiten für vektorbasierte Darstellungen thematischer Karten in einem Browser darstellen. Diese sind auch der Komplexität von Business Intelligence Anwendungen gewachsen. Mit der XML Anwendung SVG ist es möglich, thematische Karten des Geomarketing darzustellen. Konkret beinhaltet der Prototyp Funktionalität in Form von räumlicher Navigation, umfangreichen Diagrammimplementierungen sowie Übersichtskarten und Legenden. Die zeitgemäße objektorientierte Programmierung der clientseitigen Javascriptroutinen ermöglicht die einfache Erweiterung und Optimierung des

Prototypen. Diese sind notwendig, da der Anwendung noch entscheidende Konzepte fehlen. Verbesserungsvorschläge sind die funktionelle Erweiterung durch die Implementierung eines Metadatenkonzeptes. Eine automatische informationsabhängige Selektion des besten Darstellungstypen ist zu integrieren. Die kartographische Aufwertung muss durch die Implementierung weiterer Darstellungsmöglichkeiten erfolgen.

Abschließend noch eigene Gedanken zur zukünftigen Entwicklung von SVG:

Die theoretischen Ausführungen der Arbeit und der entwickelte Prototyp zeigen, dass SVG als Visualisierungsformat für thematische Karten geeignet ist. Es ist allen bisher vorhandenen Lösungen überlegen und bietet aufgrund der Offenheit Perspektiven. Auch die Tatsache, dass XML als Basis für die Kommunikation zwischen Softwaremodulen immer mehr an Gewicht gewinnt, verhilft SVG zu einer weiteren Verbreitung. Auch für graphische Darstellungen auf mobilen Geräten kann sich SVG etablieren. Die entsprechenden Empfehlungen - SVGTiny für Mobiltelefone sowie SVGBasic für *Handheld's* - sind vom W3C bereits verabschiedet. Im Bezug auf kartographische Darstellungen und Anwendungen kann SVG als optimale Darstellungsmöglichkeit gesehen werden. KIEPKE & RICHTER bekräftigen dies mit der Realisierung der Bundestagswahlpräsentation im Reichstagsgebäude (vgl. KIEPKE G. & RICHTER E. 2002). Trotzdem hängt ein zukünftiger Erfolg SVG's von weiteren Faktoren ab.

Der Standard muss besser in Anwendungen Implementiert werden. Dazu zählen die Entwicklung weiterer SVG Viewer und die bessere Integration in GIS, CAD und Graphikprogramme. Auch spezielle Entwicklungsumgebungen für SVG Lösungen würden dazu beitragen, die SVG Programmierung auch für unerfahrene Programmierer zu vereinfachen. Weiter bedarf es einer Verbesserung der SVG Spezifikation besonders beim Arbeiten mit Textblöcken und der Seitennavigation. Ein automatischer Zeilenumbruch sowie spezifizierte Methoden und Attribute zur Transformation des Koordinatensystems sind zwingend erforderlich.

Die besprochenen Probleme sind der SVG Arbeitsgruppe des W3C's bekannt. Es bleibt zu hoffen, dass diese in der nächsten Spezifikation und in zukünftigen Anwendungen gelöst werden. Allgemeine Tendenzen zeigen eine erfolgsversprechende Zukunft SVG's auf.

Literaturverzeichnis

- ADLER S. et al 2001: Extensible Stylesheet Language (XSL) Version 1.0,
(im Internet: <http://www.w3.org/TR/xsl/>).
- ADOBE 2001: Current Support for SVG – Adobe SVG Viewer (vers. 3, build 76),
(im Internet: <http://www.adobe.com/svg/indepth/pdfs/CurrentSupport.pdf>).
- ADOBE 2002: svg zone (im Internet: <http://www.adobe.com/svg/main.html>).
- ANDRIENKO G. & ANDRIENKO N. 1999: Interactive Maps for Visual Data Exploration,
(= International Journal Geographic Information Science, vol. 13 (4), June 1999, S.355-374;
oder im Internet: <http://ais.gmd.de/and/icavis/>).
- ANGSÜSSER S. 2002: Icons in (karto)graphischen Benutzeroberflächen (= Vortrag im Rahmen
des Intergeo-Kongress, Frankfurt am Main, 16. bis 18. Oktober 2002;
im Internet: http://129.187.92.218/publications/angsuesser/icons_intergeo_20021017.pdf).
- APACHE 2002a: FOP – Formatting Object Processor, (im Internet: <http://xml.apache.org/fop/>).
- APACHE 2002b: Batik SVG Toolkit, (im Internet: <http://xml.apache.org/batik/>).
- APACHE 2002c: Apache Tomcat, (im Internet: <http://jakarta.apache.org/tomcat/index.html>).
- ARLETH M. 1999: Problems in screen map design, (= OTTAWA ICA / ACI 1999 – Proceedings /
Actes, S.849-857).
- ARNBERGER E. 1997: Thematische Kartographie, 4.Aufl. 1997, Braunschweig,
(in: Das geographische Seminar).
- AVENZA SYSTEMS Inc. 2002: MapPublisher (im Internet: <http://liam.avenza.com/svg/svgmain.html>).
- BAHRENBURG G. & GIESE E. & NIPPER J. 1999: Statistische Methoden in der Geographie,
Bd.1 univariate und bivariate Statistik, 4.Auflg. (in: Teubner Studienbücher zur Geographie).
- BANGE C. & MERTENS H. & KELLER P. 2001: OLAP und Business Intelligence, 2. Aufl.,
(= BARC Studie; <http://www.barc.de/>).
- BERTIN J. 1974: Graphische Semiologie – Diagramme Netze Karten, Berlin New York.
- BILL R. 1999: Grundlagen der Geoinformationssysteme - Band 2: Analysen, Anwendungen und
Entwicklungen, 2. Aufl., Heidelberg.
- BOS B. et al 1998: Cascading Style Sheets level 2 – CSS2 Specification,
(im Internet: <http://www.w3.org/TR/REC-CSS2/>).
- BOS B. & FROUMETIN M. & ADLER S. 2002: Style Activity Statement,
(im Internet: <http://www.w3.org/Style/Activity>).
- BOUNFORD T. & CAMPBELL A. 2001: Digitale Diagramme, Info-Graphiken professionell
gestalten, München.
- BRAILSFORD D. 2002: Reconciling Web-based and Page-based Digital Documents – is SVG the
way ahead?, (im Internet: http://www.xmluk.org/slides/RAL_2001/Brailsford/ahead.pdf).
- BRAY T. & PAOLI J. & SPERBERG-MCQUEEN C. & MALER E. 2000: Extensible Markup
Language (XML) 1.0 (Second Edition), (im Internet: <http://www.w3.org/TR/REC-xml>).
- BRÜHLMEIER T. 2000: Interaktive Karten – adaptives Zoomen mit Scalable Vector Graphics,
Zürich (= Diplomarbeit am geographischen Institut der Uni Zürich, im Internet:
http://www.carto.net/papers/tobias_bruehlmeier/2000_10_tobias_bruehlmeier_diplomarbeit_adaptives_zoomen.pdf).
- BRUNNER K. 2001: Kartengraphik am Bildschirm – Einschränkungen und Probleme, Bonn,
(= kartographische Nachrichten 2001, Heft 5, S.233–239, Bonn).
- CAGLE K. et al 2001: Professional XSL, Birmingham.
- CARTWRIGHT W. & PETERSON M. & GARTNER G. 1999: Multimedia Cartography, Berlin
Heidelberg New York.

- CARTWRIGHT W. & PETERSON M. 1999: Multimedia Cartography (= CARTWRIGHT W. & PETERSON M. & GARTNER G. 1999, S.1-10).
- CECCONI A. 1999: Kartographische Darstellung von statistischen Daten im Internet, Msc Thesis, (im Internet: <http://www.geo.unizh.ch/publications/acecconi/pdf/diplom.pdf>).
- CECCONI A. & GALANDA M. 2002: Adaptive Zooming in Web Cartography, (in: SVGOpen 2002 Conference Proceeding; im Internet: http://www.svgopen.org/papers/2002/cecconi_galanda_adaptive_zooming/).
- CELINEA 2000: CR2V, raster to vector converter (im Internet: <http://www.celinea.com/>).
- CORBLEY K. 2002: Employing Advanced Geomarketing – How an Italian Start-up Telco Becomes Europe's Fastest Grower (= Geoinformatics, Magazine for Geo-IT Professionals, June/2002, S.30-33; im Internet: http://www.geoinformatics.com/issueonline/issues/2002/june_2002/pdf_june/30_33_WIND.pdf).
- COREL 2002: Corel SVGViewer preview Edition, (im Internet: <http://www.corel.com/svgviewer>).
- COX S. et al 2002: OpenGIS Geography Markup Language (GML) Implementation Specification, Version 2.1.1 (im Internet: <http://www.opengis.org/techno/specs/02-009/GML2-11.html>).
- CRÜGER I. 2002: Farbentheorie und Farbgestaltung, (im Internet: <http://www.darmstadt.gmd.de/~crueger/farbe/index.html>).
- CSIRO 2002: CSIRO SVG Toolkit, (im Internet: <http://sis.cmis.csiro.au/svg/>).
- DBXGEOMATICS 2002: SVGMapMaker (im Internet: <http://www.dbxgeomatics.com/SVGMapMaker.asp>).
- DENT B. 1996: Cartography – Thematic Map Design, 4. Aufl., Dubuque, IA Boston Buenos Aires.
- DICKMANN F. 2001: Web-Mapping und Web-GIS, Braunschweig.
- DIERCKE WELTATLAS 1988, 1. Aufl. der Neubearbeitung, Braunschweig.
- DRANSCH D. 1997: Computeranimation in der Kartographie – Theorie und Praxis, Berlin Heidelberg.
- EASTLAKE D. & REAGLE J. 2002: XML Encryption Syntax and Processing, (im Internet: <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>).
- ECMA 1999: Standard ECMA262 - ECMAScript Language Definition, 3rd edition, (im Internet: <ftp://ftp.ecma.ch/ecma-st/Ecma-262.pdf>).
- ELZAKKER C.P.J.M. 2001: Use of maps on the Web, (= KRAAK M.-J. & BROWN A. 2001, S.21-36).
- EMC3 2002: Carte thématique n° 1 (im Internet: <http://www.emc3.fr/carte1b.html>).
- FERINGA W. 2001: Design, colour, images, fonts, file-size, (= KRAAK M.-J. & BROWN A. 2001, S.187-207, Appendix B).
- FERRAILOLO J. et al 2001: Scalable Vector Graphics (SVG) 1.0 Specification, (im Internet: <http://www.w3.org/TR/SVG/>).
- FIBINGER I. 2002: SVG – Scalable Vector Graphics, München.
- FISCHER H. 2000: W3C in sieben Punkten, (im Internet: <http://www.w3.org/Consortium/Offices/Germany/sieben.html>).
- FISCHER H. 2002: XML in 10 points, (im Internet: <http://www.w3.org/Consortium/Offices/Germany/Misc/XML-in-10-points.html.de>).
- FISCHER H. & BIRKENBIHL K. 2002: Informationen zum W3C, (im Internet: <http://www.w3.org/Consortium/Offices/Germany/info.html>).
- FRÜHLING J. & STEINGRUBE W. 1997: Hinweise zur Konzeption, Gestaltung und Interpretation der Karten, (= LEIBERICH P. 1997, S.117-133).
- GENTLEWARE 2002: Poseidon for UML, (im Internet: <http://www.gentleware.com/products/index.php3>).
- HAKE G. & GRÜNREICH D. 1994: Kartographie, 7.Aufl., Berlin NewYork.

- HAROLD E. 2002: Die XML-Bibel, 2.Aufl., Bonn, (= Übersetzung der amerikanischen Originalausgabe: XML Bible 2nd Edition).
- HAZAËL-MASSIUEX D. 2002: Buy standards compliant web sites, (im Internet: <http://www.w3.org/QA/2002/07/WebAgency-Requirements>).
- HELFRICH C. 1998: Document Object Model – Interoperabilität für dynamische Web-Seiten & XML-Anwendungen, (im Internet: http://www2.informatik.uni-wuerzburg.de/staff/wolfram/lehre/9899ws_semdok/dom/ausarbeitung/index.html).
- HELD S. 2002: Bewertung des GIS-Einsatzes bei der Planung von Filialstandorten im deutschen Einzelhandel, München, (= unveröffentl. wissenschaftliche Arbeit zur Erlangung des Grades eines Diplom-Geographen).
- HIRTZLER M. 2002: Drop down list without javascript, SMIL only, (im Internet: http://pilat.free.fr/english/animer/list_smil.svg).
- HURNI L. & NEUMANN A. & WINTER A. 2001: Aktuelle Webtechniken und deren Anwendung in der thematischen Kartographie und der Hochgebirgskartographie, (in: Proceedings zum 50. Kartographentag in Berchtesgaden; im Internet: http://www.carto.net/papers/svg/articles/paper_dgfk_berchtesgaden_2001.pdf).
- IBM DEVELOPER WORKS 2001: Introduction to Scalable Vector Graphics, (im Internet: <http://www-106.ibm.com/developerworks/education/transforming-xml/xmltosvg/>).
- IBM EASE OF USE: Design, (im Internet: http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/561).
- IMHOF E. 1972: Thematische Kartographie, Berlin New York, (in: Lehrbuch der Allgemeinen Geographie).
- JACKSON D. 2002: SVG 1.1/1.2/2.0 Requirements, (im Internet: <http://www.w3.org/TR/SVG2Reqs/>).
- JACOBS I. 2001: World Wide Web Consortium Process Document, (im Internet: <http://www.w3.org/Consortium/Process-20010719/>).
- JACOBS I. 2002: W3C Activities, (im Internet: <http://www.w3.org/Consortium/Activities>).
- KIEPKE G. & RICHTER E. 2002: Kundenorientierte Webmapping Lösungen: Das Beispiel Bundestagswahl 2002, (= Proceedings zum WEB.MAPPING.2002 Symposium in Karlsruhe, Vortrag 8).
- KRAAK M.-J. 1999: 17 Cartography and the Use of Animation (= CARTWRIGHT W. & PETERSON M. & GARTNER G. 1999, S.173-180).
- KRAAK M.-J. 2001a: Setting and needs for web cartography, (= KRAAK M.-J. & BROWN A. 2001, Chapter1 S.1-7).
- KRAAK M.-J. 2001b: Cartographic Principles, (= KRAAK M.-J. & BROWN A. 2001, Chapter5 S.53-71).
- KRAAK M.-J. & BROWN A. 2001: Web Cartography – developments and prospects, London New York (im Internet: <http://kartoweb.itc.nl/webcartography/webbook/>).
- KUHLMANN U. 2002: Plan, flach, digital – Flachdisplays und Röhrenmonitoren auf der Spur, (= c't – Magazin für Computer Technik 23/2002, S.126/127).
- LAFON I. 2001: HTTP – Hypertext Transfer Protocol (im Internet: <http://www.w3.org/Protocols/>).
- LANGENSCHIEDTS FREMDWÖRTERBUCH ONLINE: (im Internet: <http://www.langenscheidt.aol.de/>).
- LATZ J. 1997: Außendienstreorganisation als Voraussetzung für die Marktausschöpfung, (= LEIBERICH P. 1997, S.173-197).
- LE HORS A. et al 2000: Document Object Model Core (level2), (im Internet: <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/core.html>).
- LEIBERICH P. 1997: Business Mapping im Marketing, 1. Aufl., Heidelberg.
- LEY J. 2002: Frequently Asked Questions about Source Code Obfuscation, (im Internet: <http://jibbering.com/faq/obfuscate.html>).

- LILLEY C. 2002: SVG Implementations, (im Internet: <http://www.w3.org/Graphics/SVG/SVG-Implementations.htm8>).
- LINDSEY K. 2002: JavaScript Object Inheritance, (im Internet: <http://www.kevindev.com/tutorials/javascript/inheritance/index.htm>).
- MACEACHREN A.M. 1994: Visualization in modern Cartography: Setting the Agenda, (= MACEACHREN A.M. & TAYLOR D.R.F. 1994, S.1-12).
- MACEACHREN A.M. & TAYLOR D.R.F. 1994: Visualization in modern Cartography, Oxford, (= Modern Cartography, volume 2).
- MAC GILLAVRY E. 2000: Cartographic aspects of WebGIS-software, Master of Arts Thesis, (im Internet: <http://www.webmapper.net/thesis/>).
- MAP2SVG: (im Internet: <http://www.gis-news.de/svg/map2svg.htm>).
- MAPINFO Coop. 2002a: MapX 5 Developer Guide, (im Internet: http://www.mapinfo.com/free/docs/mapx/MapX_50_DevGuide.pdf).
- MAPINFO Coop. 2002b: Comparison Chart – What's new with MapInfo 7, (im Internet: http://www.mapinfo.com/products/mipro/comparison_chart.pdf).
- MENG L. 2002: Personalisierung der Kartenherstellung und Mobilität der Kartennutzung, (= Kartographische Schriften 2002: Kartographie als Baustein moderner Kommunikation, Band 6, S.10-15; im Internet: http://129.187.92.218/publications/meng/paper/pers_ks2002.pdf).
- MICROIMAGES 2002: SVG Gallery, (im Internet: <http://www.microimages.com/documentation/SVG.htm>).
- MOZILLA SVG PROJECT 2002: Mozilla SVG Project, (im Internet: <http://www.mozilla.org/projects/svg/>).
- MUFF S. 2001: GeoSolutions – Skalierte, webbasierte Rauminformation der Schweiz, (= Proceedings zum WEB.MAPPING.2001 Symposium in Karlsruhe, Vortrag 9).
- NEUMANN A. & WINTER A. 2001: Kartographie im Internet auf Vektorgraphik, mit Hilfe von SVG nun möglich, (im Internet: http://www.carto.net/papers/svg/index_d.html).
- NIELSEN J. 2002: Flash and Web-Based Applications, (im Internet: <http://www.useit.com/alertbox/20021125.html>).
- NITSCHKE M. 1998: Micromarketing, Wiesbaden.
- OBJECT INTERNATIONAL SOFTWARE 2002: Together UML Modellierungswerkzeug, (im Internet: <http://www.oisoft.com/produkte/together/index.htm>).
- OLBRICH G. & QUICK M. & SCHWEIKART J. 2002: Desktop Mapping – Grundlagen und Praxis in Kartographie und GIS, 3.Aufl., Berlin Heidelberg New York.
- ORDNANCE SURVEY 2002: OS Master Map, (im Internet: http://www.ordsvy.gov.uk/os_mastermap/home/home.htm).
- PETERSON M. 1995: Interactive and Animated Cartography, New Jersey.
- PEMBERTON S. et al 2002: XHTML 1.0 The Extensible HyperText Markup Language (Second Edition), (im Internet: <http://www.w3.org/TR/xhtml1/>).
- QUINT A. 2002a: Digging Animation, (im Internet: <http://www.xml.com/pub/a/2002/01/23/svg/>).
- QUINT A. 2002b: Simple Text Wrapping, (im Internet: <http://www.xml.com/pub/a/2002/09/11/quint.html>).
- RÄBER S. & JENNY B. 2001: Attraktive Webkarten - ein Plädoyer für gute Kartengraphik, (= Proceedings zum WEB.MAPPING.2001 Symposium in Karlsruhe, Vortrag 3).
- REICHARDT M. 2001: OGC's GML2.0, (= Geoinformatics, Magazine for Geo-IT Professionals, July/August 2001, S.18-21; im Internet: http://www.geoinformatics.com/issueonline/issues/2001/07_2001/pdf_07_2001/18_21_gml.pdf).
- REICHENBERGER T. & ANGSÜSSER S. & MENG L. 2002: Mobile Kartographie – eine offene Diskussion, (= Kartographische Nachrichten, 52.Jg., H.4. Bonn, 2002: S.164-166; im Internet: http://129.187.92.218/publications/reichenbacher/Mobile_Kartographie_KN.pdf).
- RIGAUX P. & SCHOLL M. & VOISARD A. 2002: Spatial Databases with Application to GIS, San Diego.

- SAMLAND W. & HINTERMEIER A. & DÜRR J. 2000: Extensible Markup Language (XML) und Scalable Vector Graphics (SVG), (= Workshop im Rahmen des Web.mapping 2000 Symposiums).
- SCHONEFELD P. 2002: An SVG Tutorial In Pictures, (im Internet: <http://www.digitalcraft.com.au/svg/tutorials/viewingSVG.asp>).
- SCHÜSSLER F. 1999: Geomarketing, (in: GIS Tutor Online; im Internet: <http://www.gis-tutor.de/praxis/anwendg/marketin/marketin.htm>).
- SCHÜSSLER F. 2000: Geomarketing: Anwendungen Geographischer Informationssysteme im Einzelhandel, Marburg.
- SCHUMACHER S. 2002: SVG: Viewer und Editoren – Bildlich gesprochen, (= iX – Magazin für Professionelle Informationstechnik 12/2002, S.60–67).
- SHARPVECTORGRAPHICS 2002: SVG# + SVG.NET Viewer, (im Internet: <http://sourceforge.net/projects/svgdomcsharp/>).
- SHP2SVG (im Internet: <http://arcscrips.esri.com/details.asp?dbid=10362>).
- SOFTWARE MECHANICS 2002: SVGMaker, (im Internet: <http://www.svgmaker.com/>).
- SOROTOKIN P. 2002: SVG Secrets, (in: SVGOpen 2002 Proceedings; im Internet: http://www.svgopen.org/papers/2002/sorotokin_svg_secrets/).
- SPEERBERG-MCQUEEN C. M. & THOMPSON H. 2002: XML Schema, (im Internet: <http://www.w3c.org/XML/Schema>).
- SPONA H. 2001: SVG - Webgraphiken mit XML, Bonn (in: Das Einsteigerseminar).
- STEINGRUBE W. 1997: Geodaten für das Business Mapping: Die Karten, (= LEIBERICH P. 1997, S.39-65).
- SVG-WIKI 2002: SVG–Wiki Front Page, (im Internet: <http://www.protocol7.com/svg-wiki/>).
- SVGMAPPER 2002: (im Internet: <http://www.svgmapper.com/>).
- THISSEN F. 2000: Screen-Design-Handbuch, Effektiv informieren und kommunizieren mit Multimedia, Berlin Heidelberg New York.
- TOP10NL 2002: The Top10NL demonstration Website, (im Internet: http://kartoweb.itc.nl/Top10nl/TOP10NL_eng/index2.htm).
- TRIGLAV J. 2002: OS MasterMap – Heralding a New Era in GIS Applications, (= Geoinformatics, Magazine for Geo-IT Professionals, January/February 2002, S.12-15; im Internet: http://www.geoinformatics.com/issueonline/issues/2002/01_janfeb_2002/pdf/12_15_ordnance.pdf).
- TURTSCHI R. 1998: Mediendesign, Sulgen.
- UISMEDIA 2002: MapViewSVG, (im Internet: <http://www.mapview.de/>).
- WILHELMY H. 1996: Kartographie in Stichworten, Zug.
- WINTER A. 2000: Internetkartographie mit SVG – Prototyp für einen thematischen Atlas, (= Diplomarbeit zur Erlangung des Magistergrades der Naturwissenschaften; im Internet: http://www.carto.net/andre.mw/projects/atlas/internetkarto_svg_atlas_001206.pdf).
- WITT W. 1967: Thematische Kartographie - Methoden und Probleme, Tendenzen und Aufgaben, 1.Aufl., Hannover, (in: Veröffentlichungen der Akademie für Raumforschung und Landesplanung, Abhandlungen Band 49).
- WORM van den J. 2001: Web map design in practice, (= KRAAK M.-J. & BROWN A. 2001, S.87-107).
- YAHOO SVG-DEVELOPERS GROUP 2002: SVG-Developers, (im Internet: <http://groups.yahoo.com/group/svg-developers/>).
- ZIEGLER C. 2001: Gralsuche – Javascript und Objektorientierung, (im Internet: <http://www.heise.de/ix/artikel/2001/04/194/>).

Hinweis: Alle Links sind im Rahmen dieser Arbeit immer wieder aufgerufen und letztmalig am **15.1.2003** auf Gültigkeit überprüft worden. Soweit möglich, sind Internetressourcen auch im Anhang zu finden.

Anhang

Inhaltsverzeichnis der beiliegenden Cd-Rom:

- Abstract: Kurze Zusammenfassung der Arbeit.
- Referenzen: Alle Online Referenzen sind, wenn möglich, lokal von der Cd-Rom abrufbar.
- SVG Beispiele: Verschiedene anschauliche SVG Beispiele zu den relevanten Kapiteln.
- Prototyp: Der vor allem in Kapitel 4 vorgestellte Prototyp.
- Digitales Das Glossar.