

**Thomas Mailänder**

Geboren am 18.02.1976

Matrikel-Nummer: 412162



FACHHOCHSCHULE **HOCHSCHULE FÜR**  
STUTT GART **TECHNIK**

---

STUTT GART UNIVERSITY OF APPLIED SCIENCES

Studiengang Vermessung und Geoinformatik

## **Diplomarbeit**

Als PG nach der SPO-95

Ausgeführt für die Diplomprüfung am Ende des Sommersemesters 2003

## **Visualisierung von ALK-Daten mittels SVG**

**(Scalable Vector Graphics)**

Erstprüfer und Betreuer:

Prof. Dr.-Ing. Franz-Josef Behr

Zweitprüfer:

Prof. Dr.-Ing. Hardy Lehmkuhler

## Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>2</b>
<b>Abbildungsverzeichnis .....</b>	<b>6</b>
<b>Tabellenverzeichnis .....</b>	<b>8</b>
<b>Abkürzungsverzeichnis .....</b>	<b>9</b>
<b>1 Einleitung .....</b>	<b>10</b>
1.1 Zum Thema .....	10
1.2 Über die gesetzten Ziele und die benötigten Voraussetzungen .....	10
1.3 Besondere Darstellungen .....	12
1.4 Zur Struktur .....	12
1.5 Das Resultat .....	12
<b>2 Das Internet.....</b>	<b>14</b>
2.1 Entstehung .....	14
2.2 Grundlagen .....	16
2.2.1 IP-Adressen und DNS .....	16
2.2.2 TCP/IP .....	17
2.2.3 Client-Server-Struktur .....	18
2.2.4 Routing und Gateways .....	19
2.2.5 Das World Wide Web .....	19
<b>3 Die verwendeten Werkzeuge und Programmiersprachen.....</b>	<b>21</b>
3.1 Der Apache Webserver .....	21
3.2 HTML – Hypertext Markup Language .....	23
3.3 CSS - Cascading Style Sheets .....	25
3.4 JavaScript .....	26
3.5 PHP – PHP: Hypertext Preprocessor.....	32
3.6 MySQL.....	36
3.7 XML – Extensible Markup Language.....	39
3.8 SVG – Scalable Vector Graphics .....	44
3.8.1 Einführung.....	44
3.8.2 Entstehung.....	44
3.8.3 Der Adobe SVG Viewer.....	45
3.8.4 Das SVG-Dokument.....	46
3.8.5 Die ViewBox.....	48
3.8.6 Rechtecke, Kreise, Ellipsen und Linien .....	49
3.8.7 Polylinie und Polygon .....	51

3.8.8	Verwendung von Cascading Style Sheets .....	53
3.8.9	Beliebige Pfade.....	54
3.8.10	Texte.....	59
3.8.11	Gruppierungen und Transformationen .....	60
3.8.12	Symbole.....	62
3.8.13	Einbinden externer Grafiken .....	64
3.8.14	Grafische Effekte.....	65
3.8.15	Animationen und Interaktivität.....	70
<b>4</b>	<b>Die Daten der Vermessungsverwaltung .....</b>	<b>73</b>
4.1	Das Liegenschaftskataster .....	73
4.2	Das Automatisierte Liegenschaftsbuch ALB .....	74
4.3	Die Automatisierte Liegenschaftskarte ALK .....	74
4.3.1	Das Format BGRUND .....	75
<b>5</b>	<b>Das Programm BGRUNDviewer .....</b>	<b>77</b>
5.1	Die Zielgruppe.....	77
5.2	Systemanforderungen.....	77
5.3	Einlesen und Verarbeiten der BGRUND-Datei.....	79
5.3.1	Dateiupload vom Client zum Server .....	79
5.3.2	Vordaten.....	81
5.3.3	Punkte.....	81
5.3.4	Flurstücke und Grenzzüge.....	84
5.3.5	Flächen der tatsächlichen Nutzung.....	86
5.3.6	Gebäude.....	86
5.3.7	Topographische Objekte.....	88
5.3.8	Texte.....	89
5.3.9	Die Zusammenfassung der Verarbeitung .....	90
5.4	Erstellen der Grafik und der HTML-Seiten.....	91
5.4.1	Das SVG-Gerüst.....	91
5.4.2	Vordaten.....	95
5.4.3	Punkte.....	97
5.4.4	Flurstücke und Grenzzüge.....	99
5.4.5	Flächen der tatsächlichen Nutzung.....	103
5.4.6	Gebäude.....	104
5.4.7	Topographische Objekte.....	108
5.4.8	Texte.....	109
5.5	Das Menü des BGRUNDviewers.....	110
5.5.1	Aktionen beim Aufruf der Karte .....	110
5.5.2	Verschieben des Kartenausschnittes .....	110
5.5.3	Zoomen des Kartenausschnittes .....	111
5.5.4	Wiederherstellung des originalen Bildausschnittes .....	112
5.5.5	Ein- und Ausblenden von Ebenen .....	113
<b>6</b>	<b>Resümee.....</b>	<b>115</b>

6.1	Die Zukunft von SVG .....	115
6.2	Mögliche Erweiterungen des BGRUNDviewers.....	115
6.3	Zusammenfassende Schlußbetrachtung.....	119
6.4	Danksagungen .....	119
<b>Anhang: Die Dateien des Programmsystems BGRUNDviewer.....</b>		<b>121</b>
1.	Upload der BGRUND-Datei.....	121
2.	Einlesen und Verarbeiten der BGRUND-Datei.....	121
3.	Auflistung der assoziativen Arrays als Verzeichnisse für das Programm .....	121
4.	Die Klasse Vordaten .....	121
5.	Die Punkteklassen.....	121
6.	Die Klasse Punkteliste .....	121
7.	Die Klasse Linienzug.....	121
8.	Die Klasse Grenzzug .....	121
9.	Die Klasse Flurstück.....	121
10.	Die Klasse Flurstücksliste.....	122
11.	Die Klasse tatsächliche Nutzungsfläche .....	122
12.	Die Klasse Gebäude.....	122
13.	Die Klasse Gebäudeliste .....	122
14.	Die Klasse topographisches Objekt .....	122
15.	Die Klasse Liste der topographischen Objekte.....	122
16.	Die Klasse Text.....	122
17.	Die Klasse Texteliste .....	122
18.	Allgemeine Funktionen .....	122
19.	Anlegen und Löschen des benötigten Verzeichnisses und der Dateien.....	122
20.	Einbindung der Datenbankfunktionen .....	122
21.	Auflistung der durch die Verarbeitung erstellten Objekte.....	123
22.	Umschließender SVG-Bereich für die Aufnahme des Karten- und des Menüteiles.....	123
23.	CSS- Definitionen und Anlage der Filter, Verläufe und Symbole .....	123
24.	JavaScript-Funktionen für die Interaktionen der erstellten Karte.....	123
25.	SVG-Bereich mit der Kartendarstellung.....	123
26.	SVG-Bereich mit der Menüdarstellung .....	123
27.	Löschen des Verzeichnisses und der darin enthaltenen Dateien auf dem Server .....	123
<b>Glossar .....</b>		<b>124</b>
<b>Literaturverzeichnis .....</b>		<b>137</b>
<b>Webseiten .....</b>		<b>138</b>
	Die Seite überhaupt für Internet, WWW, HTML, CSS und JavaScript .....	138
	Das Internet und das World Wide Web .....	138

---

Apache Webserver.....	138
HTML – Hypertext Markup Language.....	139
CSS – Cascading Style Sheets.....	139
JavaScript.....	139
PHP – PHP: Hypertext Preprocessor.....	139
MySQL – Datenbankmanagementsystem .....	139
XML – Extensible Markup Language .....	140
SVG – Scalable Vector Graphics .....	140
Liegenschaftskataster.....	141
Online-Lexika.....	141
<b>Erklärung .....</b>	<b>142</b>
<b>Stichwortverzeichnis.....</b>	<b>143</b>

## Abbildungsverzeichnis

Abbildung 1: BGRUNDviewer .....	13
Abbildung 2: Schema der Client-Server-Struktur .....	18
Abbildung 3: Marktanteile der am meisten verbreiteten Webserver .....	22
Abbildung 4: Ein kleines HTML-Dokument.....	25
Abbildung 5: Das Document Object Model von JavaScript.....	27
Abbildung 6: Übersicht über die am meisten verwendeten Sprachen der letzten Jahre .....	30
Abbildung 7: Event-Handler und DOM, vor und nach dem Auslösen des Ereignisses .....	32
Abbildung 8: Verbreitung von PHP im World Wide Web laut Netcraft und E-Soft.....	34
Abbildung 9: phpMyAdmin .....	37
Abbildung 10: WinMySQLadmin .....	38
Abbildung 11: Aufteilung in Inhalt, Struktur und Layout .....	40
Abbildung 12: Das Konzept von XML .....	42
Abbildung 13: Kontextmenü des SVG Viewers von Adobe .....	45
Abbildung 14: Die ViewBox in SVG.....	48
Abbildung 15: Das Attribut preserveAspectRatio .....	49
Abbildung 16: Rechteck, Kreis, Ellipse und Linie .....	50
Abbildung 17: Polylinie und Polygon .....	52
Abbildung 18: moveto, (horizontal   vertical) lineto und closepath .....	56
Abbildung 19: (smooth) curveto und (smooth) quadratic Bézier curveto .....	57
Abbildung 20: Elliptical Arc .....	58
Abbildung 21: Darstellung von Texten .....	60
Abbildung 22: Transformationen mit translate, rotate, scale und skew.....	62
Abbildung 23: Verwendung von Symbolen .....	64
Abbildung 24: Einbinden von externen Grafiken.....	65
Abbildung 25: Linearer und radialer Verlauf .....	66
Abbildung 26: Füllmuster.....	68
Abbildung 27: Beschneidungsmasken.....	69
Abbildung 28: Gaußscher Weichzeichnungsfilter und Beleuchtungseffekte .....	70
Abbildung 29: interaktives SVG .....	70
Abbildung 30: Formularfeld für Dateiupload.....	79
Abbildung 31: Auswahl der Datei .....	80
Abbildung 32: Die Hierarchie der Punktartenklassen .....	83
Abbildung 33: Die Klasse Linienzug als Elternklasse.....	84
Abbildung 34: Die Zusammenfassung der Verarbeitung .....	91
Abbildung 35: Die Aufteilung in Karte und Menü.....	92
Abbildung 36: Vergleich der Darstellung mit ungekürzten und gekürzten Koordinaten .....	93
Abbildung 37: Das Menü des BGRUNDviewers.....	95
Abbildung 38: Ausgabe der Vordaten in einer HTML-Datei.....	96
Abbildung 39: Die Ausgabe der Vordaten im Menü.....	97
Abbildung 40: Auszug aus der Punkteliste.....	98
Abbildung 41: Punkte ohne und mit Punktnummer .....	99

---

Abbildung 42: Auszug aus der Flurstücksliste .....	99
Abbildung 43: Vergleich kubische Bézierkurve mit elliptischer Kurve.....	101
Abbildung 44: Die Darstellung der Flurstücksgrenzen und der Grenzzüge .....	102
Abbildung 45: Die Darstellung der Flurstücksnummern und der Zuordnungspfeile .....	103
Abbildung 46: Die Farbvorgaben der tatsächlichen Nutzungsflächen .....	103
Abbildung 47: Die Darstellung der tatsächlichen Nutzungsflächen.....	104
Abbildung 48: Auszug aus der Gebäudeliste.....	105
Abbildung 49: Der Versuch Gebäude zu schraffieren.....	106
Abbildung 50: Die Darstellung der Gebäude inklusive Ausgestaltungen .....	106
Abbildung 51: Die Darstellung von Ausgestaltungen bei Gebäuden .....	107
Abbildung 52: Auszug aus der Liste mit den topographischen Objekten.....	108
Abbildung 53: Die Darstellung der topographischen Objekte.....	108
Abbildung 54: Auszug aus der Texteliste.....	109
Abbildung 55: Die Darstellung der Lage- und Gemeindenamen sowie sonstiger Texte.....	110
Abbildung 56: Die Schaltflächen für das Verschieben des Kartenausschnittes .....	111
Abbildung 57: Die Schaltflächen für das Zoomen des Kartenausschnittes .....	111
Abbildung 58: Das Popup-Menü für die Auswahl des gewünschten Maßstabs .....	112
Abbildung 59: Die Schaltfläche für das Wiederherstellen des originalen Kartenausschnittes..	112
Abbildung 60: Die Schwarz-Weiß-Darstellung der Karte.....	113
Abbildung 61: Die Darstellung bei unsichtbarer Gebäudeebene.....	114
Abbildung 62: Die Umsetzung von Inselflächen in BGRUND.....	116
Abbildung 63: Überlappung der Nutzungsartengrenzen .....	116

## Tabellenverzeichnis

Tabelle 1: Die rasante Entwicklung des Internet .....	15
Tabelle 2: Die Netzklassen des Internet .....	16
Tabelle 3: Top-Level-Domains.....	17
Tabelle 4: Unterstützte JavaScript-Versionen der geläufigen Browser.....	28
Tabelle 5: Tastenfunktionen für den SVG Viewer von Adobe.....	46
Tabelle 6: Steuerzeichen und Parameter des path-Elementes.....	54
Tabelle 7: Verhalten der Internet-Browser bei Verwendung des SVG Viewer 3.0.....	78
Tabelle 8: Die Klasse Vordaten .....	81
Tabelle 9: Die Klasse Punkt .....	81
Tabelle 10: Die Klasse nicht numerierter Punkt.....	82
Tabelle 11: Die Klasse numerierter Punkt.....	82
Tabelle 12: Die Klasse Trigonometrischer Punkt.....	82
Tabelle 13: Die Klasse allgemeiner Punkt.....	82
Tabelle 14: Die Klasse Punkteliste.....	83
Tabelle 15: Die Klasse Linienzug.....	84
Tabelle 16: Die Klasse Flurstück.....	85
Tabelle 17: Die Klasse Grenzzug .....	85
Tabelle 18: Die Klasse Flurstücksliste.....	86
Tabelle 19: Die Klasse Nutzungsfläche.....	86
Tabelle 20: Die Klasse Gebäude.....	87
Tabelle 21: Die Klasse Gebäudeliste.....	88
Tabelle 22: Die Klasse topographisches Objekt nach altem Format .....	88
Tabelle 23: Die Klasse Liste der topographischen Objekte.....	89
Tabelle 24: Die Klasse Text .....	89
Tabelle 25: Die Klasse Lage- und Gemeindename .....	90
Tabelle 26: Die Klasse Texteliste .....	90

## Abkürzungsverzeichnis

ALK	Automatisierte Liegenschaftskarte
AP	Aufnahmepunkt, im allgemeinen ohne Schnitt- und Kleinpunkte
ARPA	Advanced Research Projects Agency
ASCII	American Standard Code for Information Interchange
DFN	Deutsches Forschungsnetz
DOM	Document Object Model
DSL	Digital Subscriber Line
GIS	Geographisches Informationssystem
GK	Gauß-Krüger
GP	Grenzpunkt
HfT	Fachhochschule Stuttgart - Hochschule für Technik
HP	Hewlett Packard
ID	Identifikation bzw. Identifizierung(snummer)
IE	Microsoft Internet Explorer
ISDN	Integrated Services Digital Network
NN	Netscape Navigator
PHP	PHP: Hypertext Preprocessor (früher: Personal Homepage)
SK	Schnitt- oder Kleinpunkt
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
TP	Trigonometrischer Punkt
VP	Versicherungspunkt
WWW	World Wide Web
XML	Extensible Markup Language

# 1 Einleitung

## 1.1 Zum Thema

In Baden-Württemberg wird die *Automatisierte Liegenschaftskarte* (ALK), die die analogen Karten FK5 (Maßstab 1:500), FK15 (im badischen Landesteil, Maßstab 1:1500) und FK25 (im württembergischen Landesteil, Maßstab 1:2500) abgelöst hat, im Format *BGRUND* geführt. In den meist sehr kostspieligen *Geoinformationssystemen* (GIS) sind dafür spezielle Konvertierungsprogramme integrierbar. In den letzten Jahren entwickelte sich, vor allem durch den Trend der Verwaltungen vermehrt Angebote über das Internet anzubieten (E-Bürgerdienste) und somit größere Bürgernähe zu demonstrieren, ein zunehmendes Interesse die Daten des Liegenschaftskatasters auch über das Internet verfügbar zu machen. Das durch diese Diplomarbeit entstandene Programm kann hinsichtlich dieses Aspektes seinen Teil dazu beitragen.

Die Aufgabe der vorliegenden Diplomarbeit bestand in der Erstellung eines Programmes, das die von der Vermessungsverwaltung beziehbaren digitalen Daten der ALK in *Vektorgrafiken* umwandeln kann. Diese Grafiken sollen in gewöhnlichen *Internet-Browsern* unter Verwendung eines speziellen *Plug-Ins*<sup>1</sup> darstellbar sein. Das Programm selbst soll auf einem *Websserver* betrieben werden und somit über *Internet* oder *Intranet* anwendbar sein.

## 1.2 Über die gesetzten Ziele und die benötigten Voraussetzungen

Diese Arbeit wurde in erster Linie hinsichtlich der nachfolgend erwähnten Aspekte durchgeführt:

- Zum einen soll das erstellte Programm eine zusätzliche Möglichkeit zur Verwendung der von der Vermessungsverwaltung erhältlichen Daten der ALK bieten. Dabei liegt das Augenmerk hauptsächlich auf dem Kreis des normalen Bürgers und kleineren Gemeinden, die sich kein kostspieliges GIS<sup>2</sup> leisten wollen bzw. können, um sich die Daten des *Liegenschaftskatasters*<sup>3</sup> in Form einer Grafik anzeigen zu lassen. Dazu sollte hier erwähnt werden, daß Behörden im Rahmen ihrer Aufgaben und Privatpersonen mit berechtigtem Interesse dazu legitimiert sind, diese Daten bei den staatlichen Vermessungsämtern, den städtischen Vermessungsämtern oder beim Landesvermessungsamt zu beziehen. Ein berechtigtes Interesse liegt zum Beispiel bei einem Grundstückseigentümer oder einem Inhaber grundstücksgleicher Rechte vor. Als Resultat wäre daraus eine stärkere Verbreitung der amtlichen Katasterdaten innerhalb der Bevölkerung und somit eine verbesserte Bürgernähe der Vermessungsverwaltung erreichbar.

---

<sup>1</sup> Erweiterung für eine Software, die zusätzliche Funktionen bereitstellt oder weitere Dateiformate unterstützt

<sup>2</sup> Geographisches Informationssystem

<sup>3</sup> Die von der Katasterbehörde geführte, aus Büchern und großmaßstäbigen Karten bestehende, nach Gemeinden und Gemarkungen gegliederte öffentliche Einrichtung, in der alle Liegenschaften so nachgewiesen und beschrieben sind, wie es die Bedürfnisse des Rechtsverkehrs, der Verwaltung, der Wirtschaft und der Statistik erfordern. Das L. dient als amtliches Verzeichnis der Grundstücke und weist die Ergebnisse der Reichsbodenschätzung nach. In digitaler Form werden ALB und ALK (in Bayern DFK) geführt.

- Als Vorlage für die Gestaltung der zu erzeugenden Karte wurde die Katasterkartenvorschrift (VwVKatKart) des Wirtschaftsministeriums Baden-Württemberg herangezogen, die auch die Grundlage der Flurkarten der Vermessungsverwaltung bildete. Allerdings wurde in einzelnen Teilen von der dort beschriebenen Darstellung abgewichen, um dem Medium des Internets besser gerecht zu werden. So spielt z.B. die Farbdarstellung eine gewichtigere Rolle, während Schraffuren, auch aus technischen Gründen, vernachlässigt wurden.
- Ziel ist es ebenfalls, in diesem Projekt Erfahrungswerte bezüglich der Verwendbarkeit des ausgewählten Grafikformates *SVG*<sup>4</sup> für die Darstellung der ALK zu sammeln. So läßt sich diese Arbeit in folgende drei Teilbereiche einteilen:
  - Einlesen der im *ASCII*<sup>5</sup>-Format vorliegenden *BGRUND*-Datei durch ein *PHP*-Skript und Bildung der enthaltenen Objekte, als da wären Punkte, Flurstücke, Grenzzüge, Nutzungsflächen, Gebäude, topographische Objekte und Texte.
  - Anschließend werden diese Objekte durch das Skript und unter Miteinbeziehung einer *MySQL*-Datenbank nacheinander in eine neu erzeugte Datei gemäß des *SVG*-Standards ausgegeben, inklusive eines Menüs durch das der Benutzer die bereitgestellten Funktionen der Karte nutzen kann.
  - Die Programmierung dieser Menüfunktionen erfolgte anhand von JavaScript, wobei jedoch beachtet werden sollte, daß hierbei nicht die bei den Browsern integrierte Script-Engine, sondern die des *SVG* Viewers benutzt wird, da nur dort das *Document Object Modell*<sup>6</sup> des *SVG*-Formates ausreichend unterstützt wird. So ist es zum Beispiel möglich, über das Menü den angezeigten Bildausschnitt zu verschieben, den abgebildeten Maßstab zu verändern oder einzelne Elementebenen ein- oder auszublenden.

Ein wichtiger Gesichtspunkt war auch die leichte Zugänglichkeit für den potentiellen Anwenderkreis. Gerade weil sich dieser nicht zwangsläufig aus Fachleuten des Umfeldes Vermessungswesen und Geoinformatik zusammensetzen soll, sollten die Voraussetzungen bezüglich Hardware, Software und Fachwissen so gering wie möglich ausfallen. Hinsichtlich der Hardware läßt sich feststellen, daß das Programm selbst für den Benutzer<sup>7</sup> transparent nutzbar ist, da es nur auf dem eingerichteten Server läuft und deshalb keine Kapazitäten vom Anwender verbraucht. Die technischen Anforderungen bezüglich der Visualisierung der erstellten Karte und der Interaktionen dürften von jedem modernen Computer erfüllt werden, die Geschwindigkeit dabei ist jedoch natürlich grundsätzlich von der Größe und dem Inhalt des abgebildeten Gebietes abhängig. Der wohl einzige kritische Punkt bei der Nutzung via Internet dürfte somit die Bandbreite des Zugangs sein, da aufgrund der zu versendenden Daten eine *ISDN*- oder *DSL*-Verbindung bzw. vergleichbares empfohlen wird.

---

<sup>4</sup> Scalable Vector Graphics

<sup>5</sup> American Standard Code for Information Interchange  
Dateien, die ausschließlich im *ASCII*-Textformat erzeugt wurden, enthalten keinerlei Gestaltung und/oder Schriftarten, aber sie können von jedem Computer gelesen und dargestellt werden.

<sup>6</sup> Vom W3C empfohlene plattform- und sprachenneutrale Schnittstelle, die Programmen und Skripten dynamischen Zugriff und die Aktualisierung des Inhalts, der Struktur und der Stilvorlagen von sowohl *HTML*- als auch *XML*-Dokumenten ermöglicht.

<sup>7</sup> Wenn hier und im Folgenden von einem *Benutzer* gesprochen wird, soll durch diese Formulierung stets auch der Fall einer *Benutzerin* mit eingeschlossen sein.

Auch die softwarespezifischen Bedingungen fallen sehr gemäßigt aus, denn prinzipiell wird nur ein aktueller Internet-Browser und ein spezielles Plug-In dafür benötigt. Zwar werden nicht alle zur Zeit im Umlauf befindlichen Browser unterstützt, die gängigsten sind jedoch, wenn auch teilweise mit Einschränkungen, einsetzbar.

Das benötigte Plug-In, z.B. den *SVG Viewer* von Adobe, der kostenlos auf der Homepage von Adobe heruntergeladen werden kann, läßt sich im Normalfall problemlos auf dem Computer installieren. Anschließend werden SVG-Dateien selbständig vom benutzten Web-Browser erkannt und mithilfe des Plug-Ins dargestellt. Bei der Anwendung des Programmes beschränkt sich das erforderliche Wissen des Benutzers somit auf die elementaren Kenntnisse im Umgang mit dem Internet, also dem Aufrufen einer bestimmten Seite im World Wide Web, der Benutzung von Links und der Auswahl der umzusetzenden Datei durch ein HTML-Formularelement. Die Benutzung des Grafikmenüs sollte durch eindeutige Symbole und Bezeichnungen intuitiv erfolgen.

### **1.3 Besondere Darstellungen**

In dieser Ausarbeitung werden besondere Begriffe und Bezeichnungen, die zum ersten Mal verwendet werden in *kursiver* Schrift angegeben, während sämtliche Abschnitte mit Auszügen oder Beispielen von Quellcode, sowie Funktions- oder Variablennamen mit einer besonderen Textart dargestellt werden. Aus persönlichen Gründen wurde bei dieser Diplomarbeit die alte deutsche Rechtschreibung verwendet.

### **1.4 Zur Struktur**

Nachdem in der Einleitung die Diplomarbeit in ihren grundsätzlichen Zügen vorgestellt wurde, folgt zuerst eine Einführung in die Entstehung und grobe Funktionsweise des Internets, gefolgt von den Erläuterungen der für dieses Projekt verwendeten Skript- und Auszeichnungssprachen. Daran anschließend wird auf die verfügbaren Daten der baden-württembergischen Vermessungsverwaltung eingegangen, hier insbesondere bezüglich des in dieser Arbeit verwendeten Formates BGRUND. Der Hauptteil dieser Ausarbeitung wird sich jedoch mit den einzelnen umgesetzten Verfahren und Techniken beschäftigen, die nötig waren, um aus den reinen Textinformationen des Liegenschaftskatasters eine interaktive Karte zu erzeugen. Den Abschluß bilden ein Ausblick auf die zukünftigen Möglichkeiten, die dieses Programm bietet, und ein persönliches Fazit.

### **1.5 Das Resultat**

Das Produkt dieser Diplomarbeit, neben dem hier vorliegenden schriftlichen Teil, soll über die Mittel des Internets öffentlich für jeden Interessierten zugänglich und benutzbar sein. So kann sich jeder im Besitz von BGRUND-Daten auf diesem Wege seine persönliche Karte zu privaten Zwecken erstellen lassen unter der Adresse: [www.BGRUNDviewer.de](http://www.BGRUNDviewer.de)

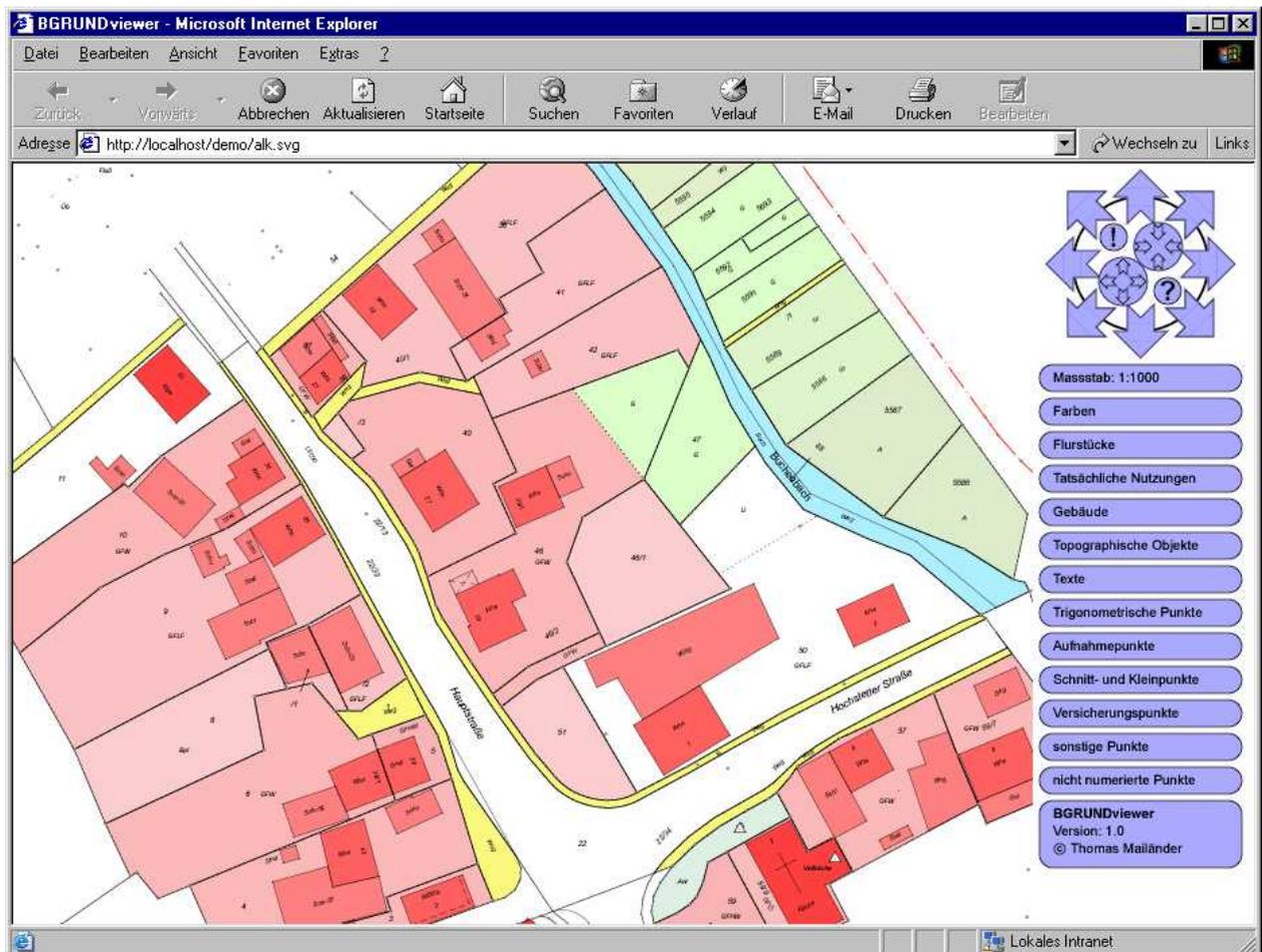


Abbildung 1: BGRUNDviewer

Hier sieht man die Darstellung der BGRUND-Testdaten des Landesvermessungsamtes Baden-Württemberg, umgesetzt durch den BGRUNDviewer und den Internet Explorer 6 mit dem Plug-In SVG Viewer von Adobe.

## 2 Das Internet

### 2.1 Entstehung

Als die Sowjetunion am 4. Oktober 1957 ihren Sputnik-Satelliten erfolgreich gestartet hatte, sahen die USA dies als bedeutende Gefährdung der nationalen Sicherheit, gerade in Anbetracht des Kalten Krieges und der damit verbundenen Bedrohung durch einen eventuellen Angriff mit Nuklearwaffen. Deswegen versuchte das amerikanische Verteidigungsministerium so schnell wie möglich den technischen Rückstand aufzuholen, und Maßnahmen zu ergreifen, die im Falle eines Erstschlages der Sowjetunion den Verlust wichtiger militärischer Daten verhindern sollten. Als Lösung ersann daher die beauftragte RAND-Corporation, diese Daten auf mehreren Rechnern in einem weitläufigen elektronischen Datennetz zu speichern. Neue und geänderte Daten sollten dabei sofort an alle anderen Rechner weitergegeben werden, um stets überall und zu jeder Zeit auf den aktuellen Datenbestand zugreifen zu können. Außerdem sollte jeder Rechner mit mehreren Verbindungen an alle anderen angeschlossen sein, um selbst bei Zerstörung eines Großteils des Netzwerkes eine zuverlässige Kommunikation gewährleisten zu können.

Des Weiteren wurde die Aufsplittung der zu verschickenden Nachrichten in kleine Datenpakete berücksichtigt, welche jeweils mit eigener Zieladresse versehen waren, und somit unabhängig von den anderen Teilnachrichten versendet werden konnten. Daraus ergab sich eine verbesserte Auslastung der vorhandenen Leitungen und eine optimierte Übertragungssicherheit durch *redundante*<sup>8</sup> Verbindungswege. Beim Empfänger wurden die einzelnen Pakete in der richtigen Reihenfolge wieder zusammengesetzt, und fehlende Teilstücke wurden neu angefordert.

Die 1958 gegründete wissenschaftliche Einrichtung *Advanced Research Projects Agency* (ARPA) besann sich 1966 auf diesen dezentralen Ansatz und erstellte innerhalb von drei Jahren eine Vernetzung von vier ihrer Großrechner<sup>9</sup>, dem sogenannten ARPA-Net. Wiederum drei Jahre später wurde dieses Netz mit jetzt schon immerhin vierzig angeschlossenen Rechnern bei der International Conference on Computer Communications vorgeführt. In der Folgezeit schlossen sich immer mehr Universitäten dem Netz an, und auch die AirForce, der Wetterdienst, die National Science Foundation (NSF) und die NASA profitierten von den gewonnenen Erfahrungswerten bezüglich der Netztechnologie.

Durch die neuen Benutzer des Netzes entstanden aber auch Probleme hinsichtlich unterschiedlicher Rechnertypen, inkompatibler Betriebssysteme und abweichenden Netzzugängen. Als Folge daraus wurde der unabhängige und somit einheitliche Protokollstandard *TCP/IP*<sup>10</sup> entwickelt, der 1982 das bis dato verwendete *NCP*<sup>11</sup> ablöste.

---

<sup>8</sup> Art der Datenhaltung, bei der bestimmte Informationen mehrfach geführt bzw. übermittelt werden

<sup>9</sup> Universität von Kalifornien in Santa Barbara, Stanford Research Institute, Universität von Utah und Universität von Kalifornien in Los Angeles

<sup>10</sup> Transmission Control Protocol/Internet Protocol, siehe Abschnitt 2.2.2

<sup>11</sup> Network Control Protocol

Dem amerikanischen Militär, immerhin ursprünglicher Auftraggeber, war allerdings trotz begrenzter Zugänge durch das Verteidigungsministerium, das vermehrte Interesse am ARPA-Net nicht geheuer, und so wurde Anfang der 80er Jahre des letzten Jahrhunderts der militärische Bereich in ein eigenes, neues Netz, dem *MILNET*<sup>12</sup> ausgelagert, um das ursprüngliche System ganz der Wissenschaft überlassen zu können. Dieses Angebot wurde in den folgenden Jahren stark genutzt, immer mehr Rechner wurden mit einbezogen und irgendwann war der Zeitpunkt gekommen, an dem das mittlerweile in die Jahre gekommene ARPA-Net nicht mehr mit der überwältigenden Entwicklung mithalten konnte. So bewilligte die US-Regierung 1986 die notwendigen Mittel für die Einrichtung einer neuen Hauptleitung unter Führung der National Science Foundation, dem sogenannten *NSFNet*, das aber auch als *Backbone*<sup>13</sup> bezeichnet wird. Dies stellte die Verbindung zwischen sämtlichen bedeutenden Rechenzentren her, über die sich wiederum kleinere Netzwerke und einzelne Rechner Zugang zum Gesamtnetz und zu anderen Teilnetzen verschaffen konnten. Diese Struktur von miteinander verbundenen Netzen führte zum Begriff des Interconnected Networks und letztendlich zum Namen *Internet*.

Das Verfahren war so erfolgreich, daß in Europa ein ebenfalls TCP/IP unterstützendes Netzsystem namens *EuropaNET* und entsprechende nationale Gegenstücke<sup>14</sup> entstanden. Auf anderen Kontinenten verliefen ähnliche Entwicklungen und nach und nach wurden diese miteinander gekoppelt, um auf Daten aus der ganzen Welt zugreifen zu können. Dies hatte zur Folge, daß das ARPA-Net nun vollständig ausgedient hatte und sein Betrieb schließlich 1990 eingestellt wurde.

Tabelle 1: Die rasante Entwicklung des Internet<sup>15</sup>

Jahr	Angeschlossene Rechner
1969	4
1971	15
1972	40
1977	111
1984	1000
1987	27000
1989	100000
1992	1000000
1994	3000000
1995	5000000
1997	16000000

<sup>12</sup> Military Network

<sup>13</sup> backbone (engl.): Rückgrat

<sup>14</sup> in Deutschland: Deutsches Forschungsnetz (DFN)

<sup>15</sup> Quelle: [http://www.dfn-expo.de/Geschichte/Geschichte\\_Internet.html](http://www.dfn-expo.de/Geschichte/Geschichte_Internet.html)

## 2.2 Grundlagen

### 2.2.1 IP-Adressen und DNS

Jeder dem Internet angeschlossene Rechner kann über eine zugewiesene *IP-Adresse* gezielt angesprochen werden. Diese Adresse setzt sich aus vier Zahlenwerten zusammen, die voneinander durch das Punkt-Zeichen getrennt sind und aus jeweils drei Ziffern bestehen. Grundsätzlich gibt es bei der Einteilung dieser IP-Nummern drei Kategorien, welche als Klasse-A-, -B- und -C-Netz definiert sind. Außerdem ist die IP-Adresse in zwei Bereiche unterteilbar, der Nummer des Netzwerkes und der Nummer des Hosts. Dabei bildet im A-Netz die erste Zahlengruppe die Netzwerknummer und die restlichen die Nummern der Hostrechner. Beim B-Netz wird das betreffende Netz durch die ersten beiden Zahlenblöcke definiert, die angeschlossenen Hosts durch die Gruppen drei und vier, und bei der Klasse C stehen sogar drei Gruppen für die Netzbezeichnung, so daß nur noch ein einziger Block für die angeschlossenen Rechner übrig bleibt. Während das Klasse-A-Netz bedeutenden Instituten, wie beispielsweise dem amerikanischen Militär vorbehalten bleibt, sind Betreiber innerhalb des B-Netzes hauptsächlich große Firmen, Universitäten und Online-Dienste. Mittlere und kleine Unternehmen, sowie kleinere Internet-Provider kommen im C-Netz unter.

Da aber, wie in der unten abgebildeten Tabelle ersichtlich ist, die Anzahl der verfügbaren IP-Adressen in den unteren Klassen stark beschränkt ist, sind einige Netzbetreiber dazu übergegangen, nicht jedem Hostrechner eine statische Nummer zu vergeben, sondern erst im Falle eines gewünschten Zugangs zum Internet wird dem entsprechenden Rechner eine zur Zeit freie Adresse, eine sogenannte dynamische IP-Adresse zugeteilt, und das auch nur für die benötigte Dauer. Bei dem zu erwartenden weiteren Wachstum des Internets wird jedoch auch dieses Verfahren irgendwann an seine Grenzen stoßen, weshalb schon Überlegungen im Gange sind, zukünftig sechs anstatt vier Bytes, also Zahlenblöcke, für eine IP-Adresse zu benutzen (Stichwort: IPv6<sup>16</sup>).

Tabelle 2: Die Netzklassen des Internet

Netzklasse	A	B	C
Beispiel IP-Adresse	<b>100.196.137.166</b>	<b>150.196.137.166</b>	<b>200.196.137.166</b>
Erste Zahlengruppe	1 – 126	128 - 191	192 – 223
Maximale Netze	126	ca. 16000	ca. 2 Mio.
Maximale Hostrechner	Ca. 16,7 Mio.	ca. 65000	254

Nun wäre es jedoch sehr mühsam, bei Aufruf einer gewünschten Internet-Seite die IP-Adresse in Form diese Zahlencodes einzugeben, und es wäre auf diese Art und Weise keine anschauliche Verknüpfung zum Seiteninhalt gegeben, wodurch sicherlich auch eine Merkfähigkeit seitens des Benutzers beeinträchtigt werden würde. Aus diesem Grund wurde eine Methode entworfen, die diese Angaben in griffigere Namen umsetzen konnte und umgekehrt. So ist es unter Verwendung dieses *Domain Name Services* (DNS) möglich, in der Adreßzeile des Browser <http://www.freemail.web.de> einzugeben und der DNS

<sup>16</sup> weiterführende Informationen unter: <http://www.heise.de/newsticker/data/jk-25.04.03-010/>

übersetzt diese Angabe so, daß die Anfrage an den Server mit der entsprechenden IP-Adresse geleitet wird.

Bei der namensbasierten Adressierungsform sind ebenfalls bestimmte Hierarchien erkennbar. So bezeichnet zum Beispiel die Deklaration `http` das verwendete Übertragungsprotokoll, `www` gibt an, daß es sich um eine Webseite handelt, `freemail` kennzeichnet in diesem Falle eine *Sub-Level-Domain*, `web` benennt die Domäne selber und `de` gibt Auskunft über die *Top-Level-Domain*.

Während die Bezeichnungen für eventuelle Sub-Level-Domains und der Domäne vom Betreiber mehr oder weniger frei gewählt werden können, gibt es bei den Top-Level-Domains nur einen bestimmten Vorrat an verfügbaren Begriffen. Diese richten sich zusätzlich entweder nach dem Land, in dem der größte Benutzerkreis für die betreffende Internetseite erwartet wird, oder auch nach der Art des Inhaltes. Natürlich gibt es auch hier Ausnahmen. So gab es in letzter Zeit die Entwicklung, daß einige Webseiten mit Bezug zu den Fernsehmedien die Top-Level-Domain `tv` verwenden, obwohl dies eigentlich für Anbieter aus dem Land Tuvalu gedacht war. Grundsätzlich können jedoch die unten aufgeführten Tabellen einen kurzen Einblick über die vorhandenen Kennungen geben.

In Deutschland werden die Domänennamen vom *DENIC*<sup>17</sup> mit Sitz in Frankfurt/Main vergeben. Die meisten Internet Provider bieten jedoch ihren Kunden an, für sie diese Arbeit zu übernehmen. Durch den großen Internet-Boom und der dadurch anschwellenden Internet-Präsenz von Unternehmen und Privatpersonen verstärkte sich selbstverständlich ebenfalls die Nachfrage nach eingängigen Internet-Adressen, und aussagekräftige Namen, wie etwa `www.auto.de`, konnten zu hohen Preisen gehandelt werden.

Tabelle 3: Top-Level-Domains

Nach Nationalität		Nach Inhalt	
<code>de</code>	Deutschland	<code>com</code>	Kommerziell
<code>at</code>	Österreich	<code>org</code>	Organisation
<code>ch</code>	Schweiz	<code>net</code>	Allgemein
<code>fr</code>	Frankreich	<code>edu</code>	Amerikanische Hochschulen
<code>it</code>	Italien	<code>gov</code>	Amerikanische Behörden
<code>nl</code>	Niederlande	<code>mil</code>	Amerikanische Militäreinrichtungen

### 2.2.2 TCP/IP

Um alle Teilnehmer des Internets mit all ihren unterschiedlichen Rechnertypen, Betriebssystemen und Netzzugängen berücksichtigen zu können, mußte ein einheitliches Format geschaffen werden, das unabhängig von den einzelnen Voraussetzungen, mit einer möglichst effizienten Ausnutzung der bestehenden Netzinfrastruktur und unter Berücksichtigung einer kontrollierten Übertragung, den Datenaustausch gewährleistet. Anfang der 70er Jahre des 20. Jahrhunderts wurde dieses Problem in der Forschung angegangen, und unter der Leitung von Robert Kahn und Vinton Cerf wurde das *Transmission Control Protocol* (TCP) entwickelt. Es sorgte für den reibungslosen Paketaustausch, indem es den Versand der Pakete ü-

<sup>17</sup> Deutsches Network Information Center, [www.denic.de](http://www.denic.de)

berwachte und diese so aufteilte, daß kein Teil des Netzes überlastet wurde. Zu diesem Zeitpunkt kümmerte es sich zusätzlich auch noch um die Adressierung sämtlicher Internet-Rechner, was aber 1980 aus Leistungsgründen von einem eigenen Protokoll, dem *Internet Protokoll* (IP), übernommen wurde. So kam es, daß Anfang 1983 im ARPA-Net das bisherige NCP von TCP/IP abgelöst wurde.

Technisch gesehen arbeitet das TCP/IP auf mehreren Ebenen. Das IP stellt die sogenannte Netzwerkschicht dar und versieht die zu verschickenden Daten gemäß der oben aufgezeigten Regeln mit den benötigten IP-Adressen. Das TCP ist ein Bestandteil der Transportschicht und seine Aufgabe ist es, die erzeugten Datenpakete zu nummerieren und beim Empfänger in der richtigen Reihenfolge wieder zusammenzusetzen. Außerdem überprüft es, ob alle Datenteile auch angekommen sind und fordert fehlende Daten beim Versender erneut an. In der Anwendungsschicht wiederum kümmern sich spezielle Protokolle um ganz bestimmte einzelne Aufgaben. So ist zum Beispiel das Hypertext Transfer Protocol (HTTP) zuständig für den Umgang mit Webseiten, das File Transfer Protocol (FTP) ermöglicht den Up- und Download zwischen Client und Server, der TELNET-Standard erlaubt den Zugriff auf einen anderen, ebenfalls dem Internet angeschlossenen Rechner, und E-Mails werden mit Hilfe der Internet Message Access Control (IMAP), dem Post Office Protocol 3 (POP3) und dem Simple Mail Transfer Protocol (SMTP) verarbeitet.

### 2.2.3 Client-Server-Struktur

Nahezu alle Vorgänge innerhalb des Internets beruhen auf der Kommunikation zwischen einem Daten anfordernden Rechner und einem Daten bereitstellenden Rechner. Dazu muß neben dem Anschluß ans Internet auch die benötigte Software installiert sein. Es gibt keine einheitliche Meinung ob jetzt der Dienste anbietende Rechner an sich, die benötigte Software, oder nur beides zusammen als Server bezeichnet werden darf. Ähnliches kann auch bezüglich des diese Dienste in Anspruch nehmenden Rechners festgestellt werden. Generell läßt sich jedoch festhalten, daß ein Server im aktiven Zustand auf Anfragen anderer Rechner wartet, um daraufhin etwa angeforderte Daten zu liefern, spezielle Programme auszuführen, darunter auch eingebundene serverseitige Skripte wie beispielsweise PHP, oder Datenbankabfragen auszuführen. Der Client hingegen bietet dem Anwender die Oberfläche, um die Anfragen zu stellen bzw. deren Ergebnisse anzuzeigen. Als spezielle Ausnahmen davon gelten dahingegen das Hochladen von Dateien auf einen Server via FTP, das Versenden von E-Mails oder von Daten mit Hilfe eines Formulars, da hier der Client den versendenden Part übernimmt.

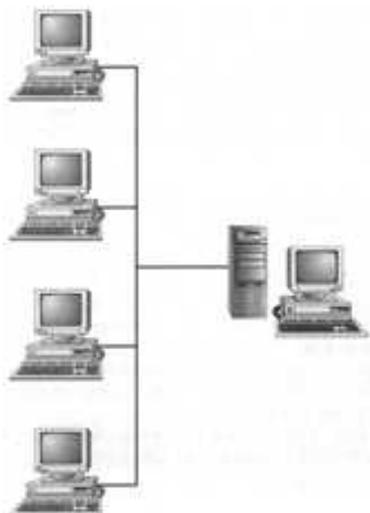


Abbildung 2: Schema der Client-Server-Struktur

## 2.2.4 Routing und Gateways

Da das Internet aus unzählbar vielen, miteinander gekoppelten Einzelnetzen besteht, und ein direkter Datentransfer nur innerhalb eines Netzwerkes möglich ist, müssen an den Übergangsstellen zwischen zwei Netzen spezielle Rechner, sogenannte *Gateways* eingerichtet werden. Wenn also jetzt ein Datentransfer zwischen zwei Rechnern in unterschiedlichen Netzwerken ablaufen soll, gehen die Daten vom Versender zum Gateway, das diese zum Gateway des Zielnetzes übermittelt. Dieses gibt die Daten dann im dortigen Netz an den eigentlich gewünschten Rechner weiter. Dieser Vorgang des Weiterleitens zwischen Sub-Netzen wird als *Routing* bezeichnet. Dabei entscheidet das liefernde Gateway jedes Mal aufs Neue, welche Leitung benutzt werden soll, abhängig von eventuellen Störungen und Auslastungen der Verbindungen.

## 2.2.5 Das World Wide Web

Das World Wide Web, das fälschlicherweise oft mit dem Internet gleichgesetzt wird, hat seine Ursprünge am europäischen Kernforschungszentrum<sup>18</sup>, mit Sitz in Genf. Der dort tätige Informatiker Tim Berners-Lee entwickelte in den 80er Jahren des vergangenen Jahrhunderts ein Programm namens *Enquire*<sup>19</sup> mit dem er die während seiner Arbeit anfallenden Daten, wie etwa Kontaktadressen, Notizen, Entwürfe und Resultate festhalten konnte. Das besondere daran war aber, daß sämtliche Einträge mit Querverweisen, den sogenannten *Links*<sup>20</sup>, zu anderen Einträgen versehen werden konnten, wobei dort gleichzeitig der Rückverweis angelegt wurde. Diese Verknüpfung zusammengehöriger Daten und die damit verbundene Technik zwischen ihnen hin- und her zu wechseln, wird als Hypertext bezeichnet. Da Berners-Lee jedoch seine Informationen auch mit anderen Kollegen austauschen wollte, machte er sich 1988 daran, dieses Verfahren auf mehrere Rechner auszuweiten. Durch seinen Kollegen Ben Segal kam er auf die Idee, dies unter Benutzung des Internets umzusetzen, und 1990 entstand der Name für dieses Projekt: Das *World Wide Web*. Dabei wurden vor allem die drei folgenden Schwerpunkte berücksichtigt:

- Nutzen des HTTP-Protokolls als Kommunikationsgrundlage zwischen Webclient und Webserver,
- das Schema der Universal Resource Identifier (URI) für eine weltweit eindeutige Adressierung sämtlicher im WWW und restlichen Internet verfügbaren Dateien,
- HTML als Auszeichnungssprache für Webdokumente.

Weiterhin programmierte Berners-Lee den nötigen Webserver und einen Browser mit integriertem Editor, den er zunächst ebenfalls als WorldWideWeb, später, um eine Verwechslung auszuschließen, als *Nexus*<sup>21</sup> bezeichnete. Nachdem er die Grundlagen dafür geschaffen hatte, stellte er Weihnachten 1990 die ersten HTML-Dokumente ins CERN-interne Netz. Im Folgejahr weitete sich seine Webtechnologie auf das Internet aus, und die Gruppe um Berners-Lee betrieb viel Öffentlichkeitsarbeit um seine Entwicklungen, vor allem bei anderen Programmierern, bekannt zu machen.

---

<sup>18</sup> Conseil Européen pour la Recherche Nucléaire, kurz: CERN

<sup>19</sup> to enquire (engl.): sich erkundigen

<sup>20</sup> link (engl.): Verweis, Verknüpfung, Verbindung

<sup>21</sup> Nexus (lat.): Zusammenhang, Verbindung

So entstanden nach und nach immer bessere Browser, davon 1992 die ersten mit einer grafischen Benutzeroberfläche (*Erwise* und *Viola*), und die Anzahl der verfügbaren Webseiten stieg stetig an.

Immer mehr Studenten versuchten sich an der Erstellung eines eigenen Webbrowsers. So auch Marc Andreessen und Eric Bina, die 1993 den am National Center for Supercomputing Application (NCSA) entwickelten MOSAIC-Browser veröffentlichten. Andreessen teilte jedoch nicht Berners-Lee idealistische Ansichten, sondern wollte mit seinem Browser das Web hauptsächlich aus kommerziellen Gründen voranbringen. So scherte er sich wenig um Konventionen und allgemeine Standards und fügte seinem Werk stetig neue Funktionen hinzu, um den Erfolg seiner Software kontinuierlich zu vergrößern. Aus demselben Grund stieg er 1994 beim Projekt der NCSA aus und gründete die Mosaic Communications Corporation, die bald darauf als *Netscape Communications Corporation* bekannt werden sollte. Im Dezember desselben Jahres erschien die erste Version des Netscape Navigators, der so erfolgreich war, daß sich Andreessen damit in den Folgejahren 90% Marktanteil sichern konnte.

Durch das nun geweckte allgemeine Interesse am World Wide Web erkannte auch der Softwareriese *Microsoft*, daß diese Technologie entgegen früherer Einschätzungen, sehr wohl ein gewichtiges Potential besitzt. 1995 wurde der erste Internet Explorer vorgestellt, jedoch mit nur wenig Erfolg. Das Unternehmen aus Redmond wollte jetzt aber unbedingt ein größeres Stück vom Kuchen abhaben, und so steckte es unglaublich viel Personal in dieses Vorhaben. Als Folge davon entstanden innerhalb von nur zwei Jahren vier Versionen des IEs, wobei der technologische Rückstand 1997 mit dem Internet Explorer 4 zum Netscape Navigator aufgeholt werden konnte. Aber erst durch die Integration des Browsers in das Betriebssystem Windows als kostenloses Zubehör gelang Microsoft der endgültige Durchbruch, und in den letzten Jahren setzte sich der IE ganz klar als dominierender Webbrowser durch.

Damit die großartige Idee von Berners-Lee nicht im Sog des Kampfes der Browser-Hersteller zugrunde ging, und das CERN weder dazu fähig war die Entwicklungen zu regeln, noch seine eigentliche Aufgabe darin sah, war es nötig geworden, eine Institution zu gründen, die sich um die technischen Grundlagen, allgemeine Standards und der Weiterentwicklung des WWW kümmern sollte. Infolgedessen wurde im Oktober 1994 am MIT Laboratory for Computer Science<sup>22</sup> (MIT/LCS) in Boston (USA) das W3C<sup>23</sup> ins Leben gerufen. W3C steht hierbei für World Wide Web Consortium, und dieses Organ setzt sich neben einem unabhängigen Kern aus Vertretern von Softwarefirmen zusammen, alles unter der Leitung des Direktors und ursprünglichen „Vaters“ des Web Tim Berners-Lee. Finanziert wird diese Einrichtung durch Beiträge der beteiligten Firmen, welche als Ausgleich aktiv an den Weiterentwicklungen mitwirken können und Zugriff auf geschützte Informationen erhalten.

---

<sup>22</sup> Homepage: <http://www.lcs.mit.edu/>

<sup>23</sup> Homepage: <http://www.w3.org/>

## 3 Die verwendeten Werkzeuge und Programmiersprachen

### 3.1 Der Apache Webserver

Wesentliche Dienste des WWW werden von HTTP-Servern, auch Webserver genannt, erbracht. Ein bedeutender Vertreter, der *Apache* Webserver, hat genau wie der Netscape-Browser, seine Ursprünge an der NCSA in Illinois. Anfang 1995 war der dort von Rob McCool entwickelte und frei verfügbare NCSA-Server, der meist benutzte Webserver des Internets. McCool hatte jedoch schon ein Jahr zuvor das NCSA verlassen, und daher wurde der HTTP-Server nicht mehr offiziell weiterentwickelt. Trotzdem ging die Entwicklung des World Wide Webs natürlich ungebrochen weiter, wodurch die bisherige Version des Programmes irgendwann den *Webmastern* nicht mehr ausreichte. So machten sie sich selber daran, die Software weiterzuentwickeln um sie an die neuen, individuellen Anforderungen anzupassen. Außerdem waren im *Quellcode* selbstverständlich noch Fehler, sogenannte *Bugs* enthalten, die ebenfalls von engagierten Webseiten-Betreibern und Programmierern angegangen wurden. Diese Veränderungen wurden durch spezielle Programme, die man als *Patch*<sup>24</sup> bezeichnet, in die Originalversion übernommen, indem fehlerhafte Abschnitte gelöscht oder durch neue Abschnitte ersetzt, und neue Bereiche für zusätzliche Funktionen eingefügt wurden. Auf diese Weise konnten die Webmaster ihre Eigenentwicklungen miteinander austauschen, um auch die Ergebnisse anderer Gleichgesinnter verwenden zu können. So kam es, daß die ehemalige Ausgangsversion des NCSA-Servers durch mehrere Patches mehr oder weniger stark modifiziert war, und es entstand aus der Bezeichnung „a patchy server“ der Begriff Apache.

Im April desselben Jahres entstand, ausgehend von Version 1.3 des NCSA-Servers, und unter Verwendung aller verfügbarer Modifikationen, die erste offizielle Apache-Version (Version 0.6.2). Bis zum Dezember wurde der Apache von der neu gegründeten *Apache Group* komplett überarbeitet und erschien unter der Versionsnummer 1.0 wiederum mit vielen hinzugekommenen Funktionen und Verbesserungen. Im Zeitrahmen nur eines Jahres gewann er so viele Anhänger, daß er den bisherigen Spitzenreiter vom NCSA in seiner führenden Stellung ablösen konnte, und wie in der folgenden Abbildung ersichtlich, im Bereich der Webserver seit nunmehr sieben Jahren unangefochten an der Spitze liegt was die Verbreitung angeht.

---

<sup>24</sup> patch (engl.): Flicker

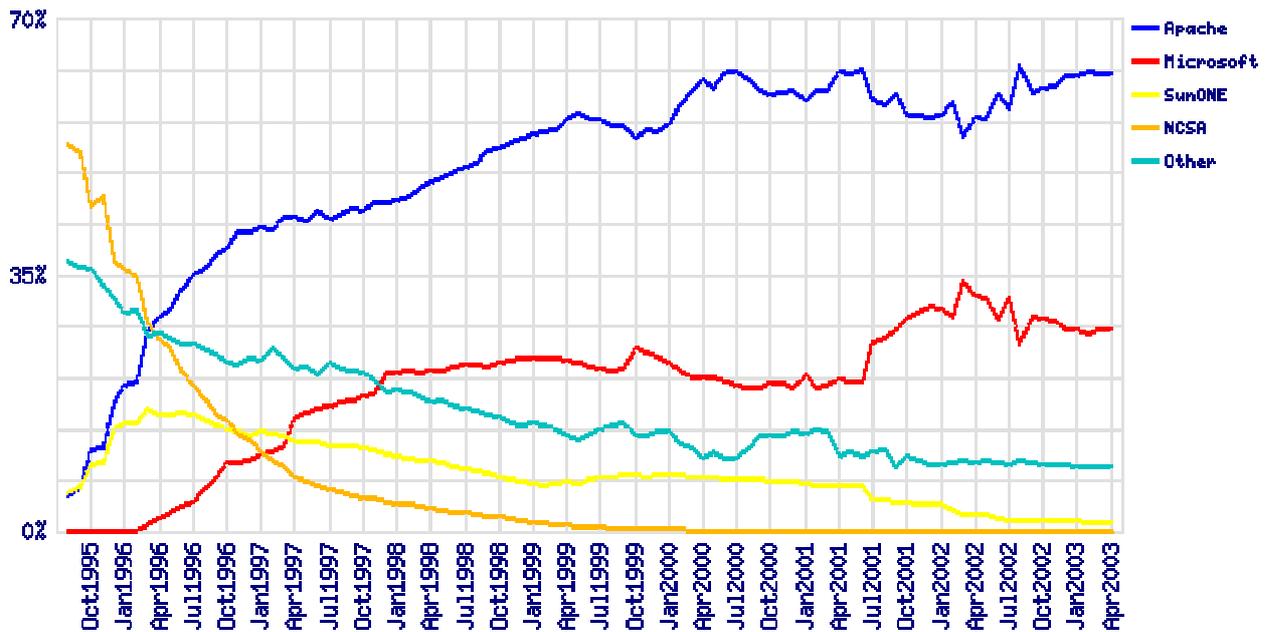


Abbildung 3: Marktanteile der am meisten verbreiteten Webserver<sup>25</sup>

Während sein Vorgänger nur für *UNIX*-Systeme vorgesehen war, gibt es den Apache Webserver ebenfalls in besonderen Varianten für die Benutzung unter dem Betriebssystem *Windows*, wodurch sich der Apache auch auf dieser Plattform, beispielsweise gegen den Microsoft-eigenen Internet-Information-Server (IIS), durchgesetzt hat. Mittlerweile liegt er in Version 2.0.45 vor und kann kostenlos bei der *Apache Software Foundation*<sup>26</sup> heruntergeladen werden.

Für die Installation gibt es zwei Möglichkeiten: entweder man wählt die Dateien als Quellcode oder schon im kompilierten Zustand. Dabei ist die kompilierte Variante, bezeichnet als Binary Version, vor allem für solche Anwender gedacht, die entweder nicht im Besitz eines eigenen Compilers sind, keine Erfahrungen mit dem Kompilieren von Quelltexten haben, oder sich einfach nicht damit befassen wollen. Der Vorteil der Sourcecode-Variante hingegen ist, daß sich der Webserver durch den modularen Aufbau der Apache-Software, entsprechend der gewünschten Anforderungen zusammenstellen und anpassen läßt.

Die großen Stärken des Apache sind zum einen seine freie Verfügbarkeit und das Prinzip des Open Source<sup>27</sup>, wodurch sich viele Interessierte an der Weiterentwicklung des Projektes beteiligen, aber auch seine Vielseitigkeit und Anpassungsfähigkeit durch das Einbinden zusätzlicher Module. Hierbei sollte jedoch erwähnt werden, daß unter dem Betriebssystem *LINUX* ein größerer Umfang an solchen Bibliotheken zur Verfügung steht als bei *Windows*-Betriebssystemen. Dies rührt von der nahen Verwandtschaft zwischen *LINUX* und *UNIX* her, welches wie oben schon erwähnt, die Plattform für den Apache-Vorgänger NCSA-Server war. Aus diesem Unterschied haben sich im Laufe der Jahre zwei feststehende Begriffe gebildet: *LAMP* und *WAMP*. Sie stehen für die Kombination aus Betriebssystem, Webserver, Datenbank und Skriptsprache, also hier für *LINUX*, Apache, MySQL und PHP, respektive *Windows*, Apache, MySQL und PHP.

<sup>25</sup> Quelle: Netcraft, Homepage: <http://news.netcraft.com/>

<sup>26</sup> Homepage: <http://www.apache.de/>

<sup>27</sup> Open Source Software zeichnet sich dadurch aus, daß ihr Quellcode veröffentlicht wird um Interessierten die Möglichkeit daran zu eröffnen sich an der Weiterentwicklung zu beteiligen.

## 3.2 HTML – Hypertext Markup Language

Es kann wohl niemand abschätzen, wie sich das World Wide Web ohne die Auszeichnungssprache *Hypertext Markup Language* entwickelt hätte, denn gerade seine Einfachheit hat ihr Tür und Tor geöffnet und machte sie zu einer der tragenden Säulen des Webs. Aufgrund der Tatsache, daß HTML ausschließlich aus ASCII-Zeichen besteht, ist es im wesentlichen schnell erlernt, und für die Erstellung und Bearbeitung genügt ein normaler Texteditor. Somit sind die ersten eigenen Dokumente schnell erstellt, und moderne Browser ignorieren auch mal den einen oder anderen kleinen Fehler. Jedoch führt gerade dies dazu, daß wahrscheinlich nur ein geringer Anteil der im Internet stehenden Seiten absolut fehlerlos und damit entsprechend der offiziellen Empfehlung des dafür zuständigen W3C sind.

Im Laufe seiner Entwicklung bis zur momentan gültigen Version 4.01<sup>28</sup>, stiegen die Fähigkeiten von HTML bemerkenswert an. War die Grundidee von Schöpfer Berners-Lee mehrere Texte miteinander durch Links zu verknüpfen, konnte man sie nach und nach in unterschiedlichen Farben, Schriftarten, Größen darstellen, zu Absätzen zusammenfassen, als Listenelemente ausgeben oder innerhalb einer Tabelle platzieren. Aber auch andere Elemente, wie Bilder oder multimediale Daten, lassen sich in HTML referenzieren und somit im Browser wiedergeben. Durch Eingabe- und Auswahlfelder innerhalb eines Formulars ergeben sich Möglichkeiten, gezielt Informationen vom Besucher der Webseite zu beziehen, die dann zum Beispiel durch *eine JavaScript-Funktion* ausgewertet werden.

Seit der Einführung der *Cascading Style Sheets*<sup>29</sup> ist es empfehlenswert, allgemein gültige Formatdefinitionen mehrerer Dokumente in einer externen Datei zu definieren, so daß diese dort zentral angelegt und bearbeitet werden können. Das Klartext-Format hat aber noch den weiteren Vorteil, daß es recht einfach durch Skripte automatisch erzeugt werden kann. Auf der Client-Seite kann dies etwa durch JavaScript und serverseitig durch CGI und Perl, PHP oder JSP<sup>30</sup> ausgeführt werden.

HTML wurde mit Hilfe von *SGML*<sup>31</sup> definiert, inzwischen gibt es aber eine Neudefinition namens *XHTML*<sup>32</sup> auf Basis des immer bedeutender werdenden Standards *XML*<sup>33</sup>.

Eine Auszeichnungssprache wie HTML läßt sich grob gesagt in zwei Bestandteile zerlegen. Zunächst gibt es die Steuerzeichen zur Strukturierung des Dokumentes. Diese werden in Form von sogenannten *Tags*<sup>34</sup> angegeben, wobei es davon zwei Arten gibt: umschließende und allein stehende. Die zweite Komponente einer Hypertext Markup Language-Datei ist der eigentliche Inhalt, also der anzuzeigende Text. Um die Unterscheidung zwischen den beiden Tag-Arten zu veranschaulichen, folgt an dieser Stelle jeweils ein kleines Beispiel:

umschließendes Tag:

```
<h1>Ich bin eine &Uuml;berschrift</h1>
```

---

<sup>28</sup> Spezifikation unter: <http://www.w3.org/TR/html4/>

<sup>29</sup> kurz: CSS, siehe Abschnitt 3.3

<sup>30</sup> Java Server Pages

<sup>31</sup> Standard Generalized Markup Language

<sup>32</sup> Spezifikation unter: <http://www.w3.org/TR/xhtml1/>

<sup>33</sup> Extensible Markup Language, siehe Abschnitt 3.7

<sup>34</sup> tag (engl.): Etikett, Kennzeichnung, Marke

alleinstehendes Tag:

Gleich folgt ein Zeilenumbruch<br>

Wie man sieht, wird im ersten Fall der auszugebende Text vom öffnenden und schließenden Tag klar eingegrenzt, wodurch der Gültigkeitsbereich der Formatangabe eindeutig festgelegt ist. Dabei heißt das Ende-Tag genauso wie das Start-Tag, zuzüglich des vorangestellten „/“-Zeichens. Was man an diesem Beispiel ebenfalls erkennen kann ist, daß Sonderzeichen, wie etwa Umlaute oder das „ß“, in HTML durch sogenannte *named Entities*<sup>35</sup> angegeben werden sollten, um unabhängig von Browser-Einstellungen die korrekten Zeichen anzuzeigen. Des weiteren gibt es bezüglich der Steuerzeichen feste Regeln wie sie zu verschachteln sind, so müssen beispielsweise einzelne Listenpunkte von dem öffnenden und schließenden Tag für die Liste selbst umgeben werden, wie folgendes Muster zeigt:

```
<ul>
  <li>Dies ist der erste Listenpunkt</li>
  <li>Dies ist der zweite Listenpunkt</li>
  <li>Dies ist der dritte Listenpunkt</li>
</ul>
```

Das komplette HTML-Dokument wird immer durch die Tags <html> und </html> umschlossen. Darin gibt es zwei Bereiche, den Kopfteil, begrenzt durch die Markierungen <head> bzw. </head>, und den eigentlichen Inhalt innerhalb der <body>-Tags. Der Kopfteil dient dazu allgemeine Informationen zum vorliegenden Dokument anzugeben, wie etwa den Titel oder Autor der Seite, aber auch bestimmte Schlagworte die von Suchmaschinen ausgewertet werden können. So ergibt sich folgender Aufbau für eine komplette Datei in HTML:

```
<html>
  <head>
    <title>Meine erste HTML-Seite</title>
  </head>
  <body>
    <h1>Ich habe es geschafft!</h1>
    Dies ist eine einfache Webseite.<br>
    War gar nicht so schwer sie zu erstellen.
  </body>
</html>
```

Das Ergebnis sieht in diesem Falle auf dem Internet Explorer bzw. Netscape Navigator folgendermaßen aus:

---

<sup>35</sup> entity (engl.): Einheit, Instanz

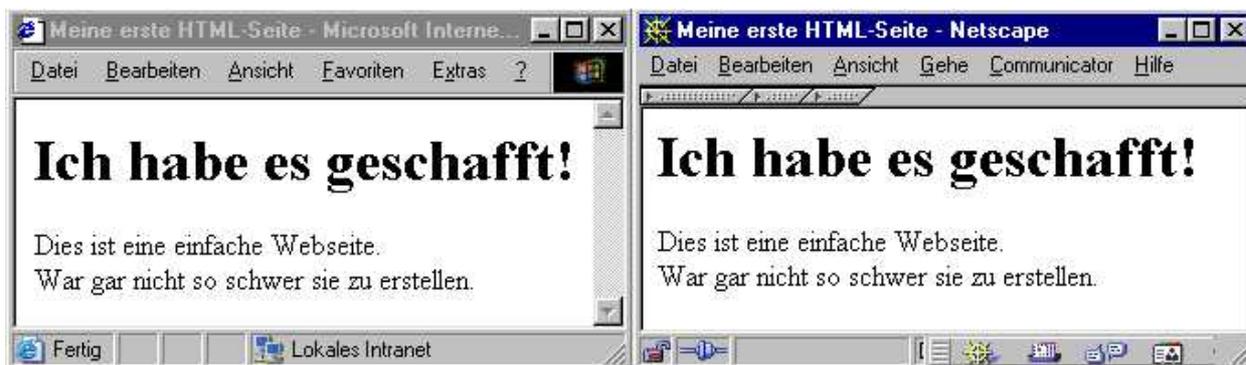


Abbildung 4: Ein kleines HTML-Dokument

Bei solch einfach gehaltenen Dateien sind kaum Unterschiede bei der Präsentation durch die Webbrowser bemerkbar. Jedoch gilt, daß bei zunehmender Komplexität und eigenen Formatdefinitionen die Eigenheiten der einzelnen Browser, dazu gehören natürlich auch die Produkte von *Opera*<sup>36</sup>, *Mozilla*<sup>37</sup> und ähnliche, einen immer größer werdenden Einfluß auf die Ausgabe des Inhaltes haben. Diese uneinheitliche Verarbeitung der Tags und der angegebenen Formate stellt immer wieder Webdesigner vor Probleme und erfordert in vielen Fällen einen zusätzlichen Aufwand bei der Erstellung.

### 3.3 CSS - Cascading Style Sheets

Die eigentliche Aufgabe von HTML war alleine die Strukturierung des Textes und der restlichen Inhalte. Dadurch daß die Browser-Hersteller jedoch versuchten, durch neue Gestaltungsmöglichkeiten der Webseiten den Marktanteil ihres Produktes zu erhöhen, bekam es aber zusätzliche Erweiterungen, die Formatierungen der einzelnen enthaltenen Elemente zuließ. Das World Wide Web Konsortium steuerte dieser Entwicklung durch die Erstellung eines Standards entgegen, im Dezember 1996 offiziell in Form der Cascading Style Sheets (CSS) veröffentlicht. Inzwischen liegt CSS 2<sup>38</sup> vor, an der Version 3 wird schon gearbeitet, und trotzdem tun sich Browser-Entwickler immer noch schwer alle Vorgaben der ersten beiden Fassungen korrekt umzusetzen. Aber auch bei den Erstellern von Webseiten hat sich die Vorgehensweise, sämtliche Formatangaben aus dem HTML-Bereich herauszulassen und über CSS zu bestimmen, noch nicht hundertprozentig durchgesetzt.

Dabei bieten die Cascading Style Sheets beachtliche Vorteile. So können CSS-Bereiche innerhalb von HTML-Dokumenten direkt angegeben werden, oder man setzt dort nur einen Verweis auf eine externe Datei, die einzig Deklarationen gemäß dieses Standards enthält. Da diese Datei dann natürlich auf allen HTML-Seiten eines Projektes angegeben werden kann, lassen sich auf diese Weise ganz einfach zentrale Formate definieren oder auch verändern.

Hier sieht man, wie Cascading Style Sheet-Bereiche im Head-Teil des HTML-Dokumentes eingebunden werden:

<sup>36</sup> Homepage: <http://www.opera.com/>

<sup>37</sup> Homepage: <http://www.mozilla.org/>

<sup>38</sup> Spezifikation unter: <http://www.w3.org/TR/REC-CSS2/>

```
<style type="text/css">
  <!--
    h1 { color:red; font-size:36pt; }
  -->
</style>
```

Hierbei werden die Steuerzeichen „<!--“ und „-->“ dazu benutzt, um Browser, die kein CSS unterstützen, davon abzuhalten, diesen Bereich als HTML umsetzen zu wollen.

Und so wird der Verweis auf eine externe CSS-Datei gesetzt, ebenfalls im Head-Bereich:

```
<link rel="stylesheet" type="text/css" href="formate.css">
```

Es wird also in diesem Beispiel die Datei `formate.css` referenziert.

Außerdem können CSS-Definitionen auch direkt bei den entsprechenden HTML-Tags gesetzt werden:

```
<h1 style="color:red; font-size:36pt">Dieser Text ist rot</h1>
```

Ebenfalls eine sehr sinnvolle Möglichkeit ist eine Formatdefinition für eine Klasse zu vergeben:

```
<style type="text/css">
  <!--
    dickUndBlau { color:blue; font-weight:bold; }
  -->
</style>
```

Im HTML-Teil kann dann diese Klasse einem Element zugewiesen werden:

```
<h1 class="dickUndBlau">Dieser Text ist fettgedruckt und blau</h1>
```

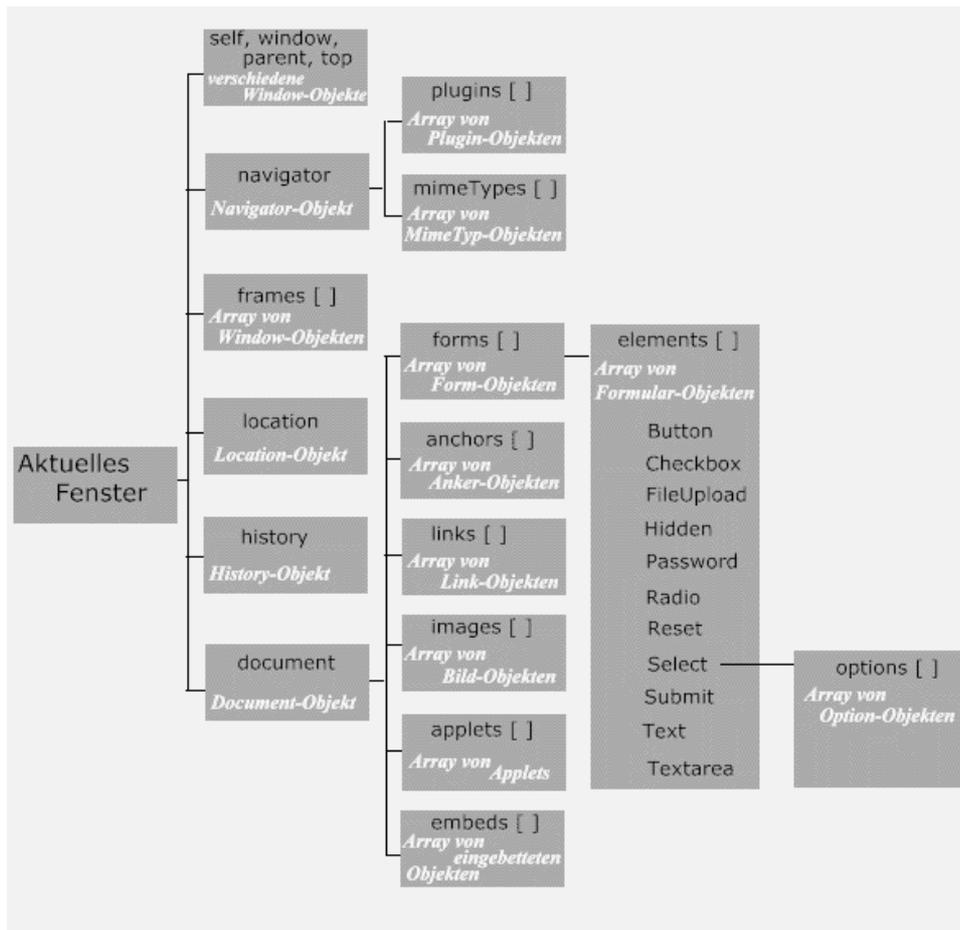
All diese Variationen lassen sich mehr oder weniger problemlos miteinander kombinieren und eröffnen damit dem Webdesigner fast grenzenlose Freiheiten bei der Gestaltung seiner Seiten.

Ferner ist es möglich, unterschiedliche Formatangaben in Abhängigkeit vom Ausgabemedium zu bestimmen, so daß etwa ein Text auf dem Bildschirm in blauer Farbe erscheint, während er bei einem Ausdruck in schwarz dargestellt wird.

### 3.4 JavaScript

Obwohl JavaScript unter bestimmten Bedingungen auch serverseitig eingesetzt werden kann, ist es meist nur als Werkzeug für den clientseitigen Umgang mit HTML-Dokumenten bekannt. Seine Anfänge lassen sich im September 1995 finden, als das Unternehmen Netscape eine Vorversion des Navigators 2.0 veröffentlichte. In dieser war die Fähigkeit enthalten, Webseiten durch eine eingebettete Skriptsprache dynamisch zu gestalten. Jene Programmiersprache, entwickelt von Brendan Eich, wurde zunächst *LiveScript* bezeichnet, doch schon im Dezember des selben Jahres benannte Netscape sie in Kooperation mit Sun, in JavaScript 1.0 um, wodurch sich die Unternehmen die Erfolgswelle der Hochsprache Java zunutze machen wollten.

Schon im darauffolgenden Jahr lieferte Netscape mit der Kombination aus dem Navigator 3.0 und JavaScript 1.1 die Möglichkeit der Kommunikation zwischen JavaScript, Java und Plug-Ins via LiveConnect. Des weiteren waren für die neue Version der Sprachkern und das Document Object Model (DOM) umfangreich geändert worden. Microsoft dagegen hinkt der Entwicklung hinterher.

Abbildung 5: Das Document Object Model von JavaScript<sup>39</sup>

Der kurz darauf erscheinende Internet Explorer 3 unterstützt zwar ebenfalls JavaScript, jedoch auf vergleichbarem Niveau wie der inzwischen abgelöste Navigator 2.0. Da Netscape der Lizenzinhaber von JavaScript war, mußte Microsoft auf einen anderen Namen ausweichen und wählte dafür *JScript*. Dafür konnte deren Browser mit einer zusätzlichen Skriptsprache aufwarten: VBScript<sup>40</sup>.

Anfang 1997 kreiert Microsoft den Begriff *Dynamic HTML* (DHTML) als die Verbindung von HTML, Skripting und Style Sheets, während sich JavaScript als Instrument für dynamische Webseiten fest etabliert hat. Unter der neuen Bezeichnung Communicator brachte Netscape im Juni den Navigator 4.0 mit integriertem JavaScript 1.2 heraus und veröffentlicht gemeinsam mit der ECMA<sup>41</sup> einen, auf JavaScript basierenden Standard namens *ECMAScript*. Vier Monate später erscheint der MSIE 4, der JavaScript 1.1 vollständig berücksichtigt, aber aufgrund eigener DOM-Erweiterungen Inkompatibilitäten zum Kontrahenten entstehen läßt. Im Februar 1998 wird der Sourcecode des Navigators freigegeben, dadurch entsteht das Mozilla-Projekt.

<sup>39</sup> Quelle: <http://www.ik.fh-hannover.de/person/becher/edvhist/sprachen/jsript.html#geschichte>

<sup>40</sup> Programmiersystem auf der Basis von Visual Basic (einer objektbasierten Programmiersprache von Microsoft), mit dem man die mit ihm erstellten Funktionen in eine HTML-Seite integrieren kann.

<sup>41</sup> European Computer Manufacturers Association, Homepage: <http://www.ecma-international.org/>

Nach Berechnungen von Netscape wird JavaScript im April auf 3,5 Millionen Webseiten betrieben und ist damit die am weitesten verbreitete Skriptsprache im World Wide Web. Die neue Version 1.3 wird im Juli zusammen mit dem Navigator 4.5 veröffentlicht, der sich laut Hersteller wie kein anderer an dem ECMAScript-Standard orientiert. Diese behält ihre Verwendung bis zum Navigator 4.7, beim Internet Explorer entspräche das der Version 5.0.

Netscapes Browser mit der Versionsnummer 6.0 und JavaScript 1.5 bekam neben ein paar kleineren Änderungen im Skript-Bereich, vor allem Neuerungen bezüglich des DOM. Microsoft hatte zwar inzwischen seinen Konkurrenten technisch überholt, bescherte sich aber selbst Probleme, indem es mit der Version 5.5 Plattformabhängigkeiten des Browsers erzeugte.

Inzwischen steht der Netscape 7.02 dem Internet Explorer 6.0 gegenüber, und nicht zuletzt aufgrund der Eingliederung in seine Betriebssysteme konnte der Konzern aus Redmond die ursprüngliche Marktsituation auf den Kopf stellen und den MSIE als eindeutig bestimmenden Browser durchsetzen. Daran änderten bisher auch die nach und nach hinzugekommenen Alternativen, wie etwa Opera, Mozilla oder Konqueror nichts. Durch den Wettstreit zwischen den beiden großen Internet-Browsern und die, aufgrund der teils eigenwilligen Umsetzungen und Erweiterungen entstandenen Inkompatibilitäten, ist es für Web-Programmierer sehr schwierig geworden, JavaScript in umfangreicherem Ausmaß zu verwenden. So müssen teilweise unterschiedliche Versionen für die verschiedenen verwendeten Browser erstellt werden, was natürlich einen beträchtlichen Mehraufwand bedeuten kann. Ansonsten sollte sich der Programmierer nur auf die wirklich gängigen Funktionalitäten beschränken, um nicht den einen oder anderen Benutzer aufgrund seines verwendeten Browsers auszuzugrenzen.

Tabelle 4: Unterstützte JavaScript-Versionen der geläufigen Browser

Netscape-Browser	JavaScript-Version	Microsoft-Browser	JavaScript-Version	Sonstige Browser	JavaScript-Version
2.0	1.0	IE 3.0	1.0	Mozilla 1.0	1.5
3.0	1.1	IE 4.0	1.1	Opera 5.x	1.4
4.0 – 4.05	1.2	IE 5.0	1.3	Opera 6.0	1.4
4.06 – 4.7x	1.3	IE 5.5	1.3	Opera 7.x	1.4
6.0	1.5	IE 6.0	1.3		
6.x	1.5				
7.0	1.5				

Die Tatsache, daß JavaScript so sehr von dem verwendeten Browser abhängt kommt daher, daß es eine interpretierte Sprache darstellt. Das heißt, daß es nicht alleine, sondern nur innerhalb einer Laufzeitumgebung ausgeführt werden kann. Diese werden von modernen Browsern als JavaScript-Interpreter<sup>42</sup> zur Verfügung gestellt. Dadurch wird auch ermöglicht, daß gewisse Sicherungsmaßnahmen getroffen werden

<sup>42</sup> interpreter (engl.): Übersetzer, Dolmetscher; Programm, das Befehle einer Programmiersprache in Maschinsprache umwandelt

können. Dies ist vor allem daher ein wichtiger Gesichtspunkt, da JavaScript schon seit jeher als kritischer Sicherheitsaspekt gilt.

In den Sprachelementen unterscheidet man von der Laufzeitumgebung unabhängige Bestandteile (die sogenannten "Core"<sup>43</sup>-Elemente der Sprache), und laufzeitumgebungsabhängige Bestandteile, die eigentlich vom Browser bzw. Server zur Verfügung gestellt werden (Browser- bzw. Serverobjekte). Die Objektivität von JavaScript ist angelehnt an Java, geht aber weniger weit. Trotz gewisser Ähnlichkeit in der Syntax<sup>44</sup> ist JavaScript aber, abgesehen von der Namensgebung, nicht von Java abgeleitet.

JavaScript verwendet für die Datentypen sogenanntes "loose typing". Variablen brauchen also bei der Deklaration nicht explizit mit einem Datentyp vereinbart werden, dieser ergibt sich bei der Verarbeitung aus dem Kontext und kann auch (für eine Variable) im Zug der Verarbeitung wechseln. Ansonsten kennt JavaScript Zahlen (number), Zeichenketten (strings), Wahrheitswerte (boolean), Arrays und Objekte.

Trotz des starken Aufkommens serverseitiger Skriptsprachen in den letzten Jahren, der bekannten Sicherheitsrisiken und der uneinheitlichen Integration in die vorhandenen Webbrowser, wird JavaScript immer noch sehr stark auf Webseiten verwendet. Ursachen dafür sind sicherlich der schnellere und unkompliziertere Einstieg im Vergleich zu serverseitigen Programmiersprachen, die Tatsache, daß bei Ausführen einer Aktion nicht jedesmal die Verbindung zum Server neu hergestellt werden muß, da sie ja lokal beim Client abläuft, und die Fähigkeit auf bestimmte Ereignisse, den sogenannten Events, innerhalb des Browsers zu reagieren.

---

<sup>43</sup> core (engl.): Kern, Herzstück, Innenteil

<sup>44</sup> Sprachregel, Satzlehre

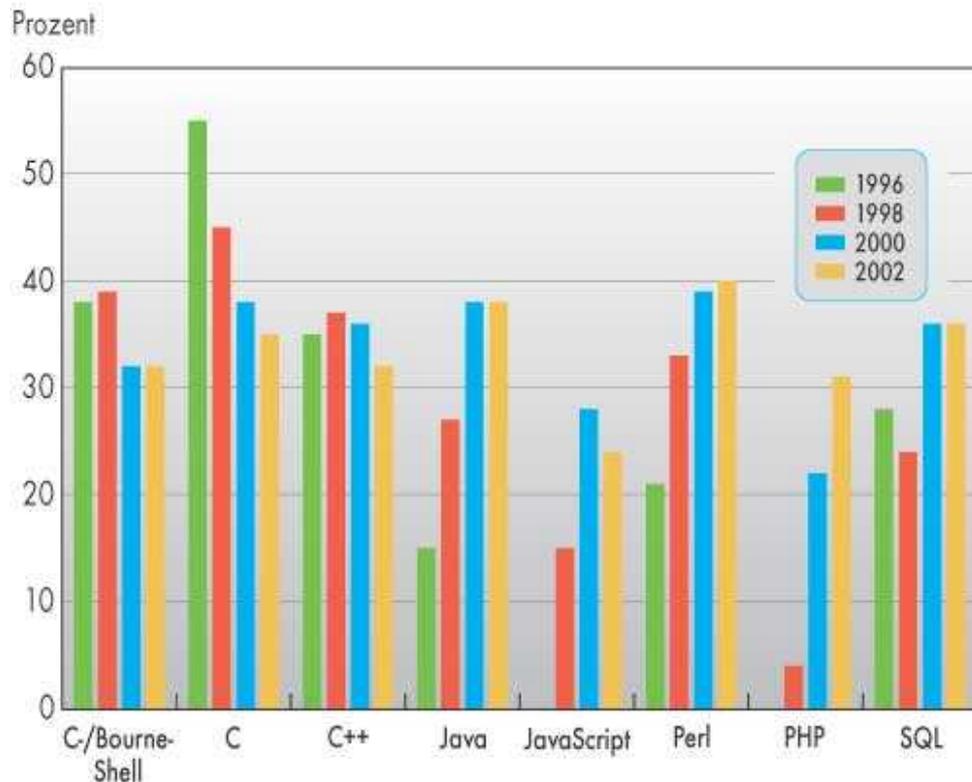


Abbildung 6: Übersicht über die am meisten verwendeten Sprachen der letzten Jahre<sup>45</sup>

Ähnlich wie Cascading Style Sheets können JavaScript-Anweisungen entweder innerhalb des HTML-Dokuments aufgeführt sein, oder sie werden dort nur mittels einer Pfad- und Dateiangabe referenziert und somit in eine externe Datei ausgelagert.

```
<html>
  <head>
    <title>Meine erste Seite mit eingebettetem JavaScript</title>
  </head>
  <body>
    <h1>HTML mit JavaScript</h1>
    <script type="text/javascript">
      <!--
        document.write("JavaScript - eine interpretierte, ");
        document.write("objektbasierte Sprache für ");
        document.write("dynamische Webseiten");
      //-->
    </script>
  </body>
</html>
```

<sup>45</sup> Quelle: <http://www.heise.de/ix/>

JavaScript-Bereiche innerhalb von HTML sollten immer zusätzlich von den Steuerzeichen für einen HTML-Kommentar ( `<!--` und `//-->` ) umgeben werden, damit Browser ohne JavaScript-Kenntnisse nicht versuchen diesen Teil als HTML zu *parsen*<sup>46</sup>.

Alternativ dazu läßt sich, wie schon erwähnt, auf eine externe Datei mit JavaScript-Inhalt und der Erweiterung `.js` verweisen.

```
<html>
  <head>
    <title>Einbeziehung einer externen JavaScript-Datei</title>
    <script src="funktionen.js" type="text/javascript"></script>
  </head>
  <body>
    <h1>HTML mit JavaScript in einer separaten Datei</h1>
  </body>
</html>
```

In der angegebenen Datei stünde dann beispielsweise:

```
alert("Herzlich willkommen!");
```

Dabei ist zu beachten, die Datei schon möglichst im Head-Bereich anzugeben, damit eventuelle Funktionen schon beim Einlesen des Body-Teiles zur Verfügung stehen.

Vor allem durch die Kombination der Event-Handler mit dem DOM kann JavaScript zu einem sehr hilfreichen Werkzeug eingesetzt werden.

```
<html>
<head>
<title>Event-Handler und DOM</title>
<script type="text/javascript">
<!--
function aendereDarstellung(element) {
  with (element.style) {
    if(fontSize == "15pt") {
      fontFamily = "Algerian";
      fontSize = "30pt";
    } else {
      fontFamily = "Arial";
      fontSize = "15pt";
    }
  }
}
//-->
</script>
```

---

<sup>46</sup> to parse (engl.): grammatikalisch bestimmen, Satzbau analysieren; Analyse und Aufbereitung eines Dokumentes entsprechend der Syntax

```

</head>
<body>
<h1>Dynamisches HTML</h1>
<span style="font-family:Arial;font-size:15pt"
onclick="aendereDarstellung(this)" >
  Klick mich!
</span>
</body>
</html>

```



Abbildung 7: Event-Handler und DOM, vor und nach dem Auslösen des Ereignisses

Zu beachten hierbei ist jedoch, daß es sich bei diesem DOM nicht um den allgemein gültigen Standard, sondern eine spezielle HTML-Variante handelt. Diese wird quasi als Überbrückungsmethode bis zur endgültigen Implementierung des offiziellen Document Object Models (ab JavaScript 1.5) verwendet.

### 3.5 PHP – PHP: Hypertext Preprocessor

PHP ist ähnlich wie HTML, ein typisches Beispiel für die vielen Erfolgsgeschichten, die das Internet mit sich brachte. Ebenfalls ursprünglich als persönliches Hilfsmittel gedacht, entwickelte es sich unterstützt durch eine tatkräftige Interessengruppe innerhalb kürzester Zeit zu einer der beliebtesten Skriptsprachen<sup>47</sup> für das World Wide Web. Mitte der 90er Jahre des letzten Jahrhunderts entwickelte der Däne Rasmus Lerdorf eine Reihe von Makros, die ihm sowohl für statistische Auswertungen seiner privaten Homepage, aber auch für die Verarbeitung von HTML-Formularen dienen sollten. Während er die erste Gruppe von Programmen als *Personal Home Page Tools* bezeichnete, nannte er die zweite *Form Interpreter*. Diese beiden Programmpakete fanden schnell im Bekanntenkreis Anklang, und Lerdorf war bereit, diese unter dem Namen *PHP/FI* weiterzugeben.

<sup>47</sup> siehe Abbildung 6: Übersicht über die am meisten verwendeten Sprachen der letzten Jahre und Abbildung 8: Verbreitung von PHP im World Wide Web laut Netcraft und E-Soft

Danach machte er sich daran, die Funktionalität seiner Programme zu erweitern und den Interpreter weiterzuentwickeln. Das Ergebnis veröffentlichte er 1997 als PHP/FI2. Sein Projekt fand immer mehr Anwender, aber auch unter den Entwicklern wurde es immer interessanter, und so ergab es sich, daß ein Entwicklerteam gegründet wurde, welches die Software noch weiter voran bringen sollte.

Darunter befanden sich unter anderem die beiden israelischen Informatikstudenten Zeev Suraski und Andi Gutmanns, die in der Folgezeit den Interpreter gründlich überarbeiteten, während Shane Caraveo sich um die Portierung auf Windows-PCs kümmerte. Weiterhin beteiligten sich vor allem Stig Sæther Bakken und Jim Winstead an dem Projekt. Das Ergebnis wurde im Juni 1998 als PHP3 vorgestellt, wobei die Bezeichnung PHP von da an offiziell für *PHP Hypertext Preprocessor* steht. Gegenüber dem Vorgänger PHP/FI2 überzeugte PHP3 vor allem durch seine Erweiterungsmöglichkeiten, die solide Infrastruktur, der Möglichkeit einer objektorientierten Syntax, sowie der Unterstützung der gängigsten Datenbanken. Somit konnte man PHP3 getrost als vollwertige Skriptsprache bezeichnen, auch wenn durch die Veränderungen nicht mehr viel von der ursprünglichen Syntax von PHP/IF übrig geblieben war.

Suraski und Gutmanns waren aber immer noch nicht mit dem Erreichten zufrieden, weswegen sie sich schon kurz nach der Fertigstellung von PHP3 mit der Version 4.0 beschäftigten. Ihr Schwerpunkt lag vor allem auf dem zentralen Kern des Systems, den sie so modifizieren wollten, daß er hauptsächlich bei komplexeren Applikationen einen Leistungsgewinn erfahren würde. Überdies sollte der modulare Aspekt von PHP intensiviert werden. Die resultierende Skript-Engine wurde, in Anlehnung an Suraskis und Gutmanns Vornamen, als *Zend-Engine* (Zeev und Andi) Mitte 1999 erstmals eingeführt, und ein Jahr später wurde PHP4, mit stark verbesserter Leistung, Unterstützung verschiedenster Webserver und vielen neuen Sprachkonstrukten, dann offiziell freigegeben. Momentan wird an der Version 5.0 gearbeitet.

Als Grund für seinen Erfolg kann PHP sicherlich die sehr gute Zusammenarbeit mit dem Webserver Apache und der Datenbank MySQL anführen. Alle drei Komponenten sind kostenlos zu beziehen, und im Laufe der Jahre entwickelte sich diese Kombination zu einem stabilen und vielseitigen Werkzeug. PHP kann auf allen gängigen Betriebssystemen verwendet werden, inkl. Linux, vielen UNIX-Unterarten (inkl. HP-UNIX, Solaris und OPENBSD), Microsoft Windows, Mac OS X und RISC OS, unterstützt aber auch einen Großteil der verbreiteten Webserver, wie den bereits erwähnten Apache, Microsoft Information Server, Personal Web Server, Netscape und iPlanet Server, Oreilly Website Pro Server, Caudium, Xitami und OmniHTTPd. Neben MySQL können ebenfalls die Datenbanken Adabas D, dBase, Empress, Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase, mSQL, Direct MS-SQL, Oracle, Ovrimos, PostgreSQL, Solid, Sybase, Velocis und Unix dbm angeschlossen werden. Durch die zusätzliche Unterstützung der ODBC<sup>48</sup> ergeben sich sogar noch mehr Möglichkeiten.

---

<sup>48</sup> Open Database Connectivity: von Microsoft entwickelter Standard für den Zugriff auf Datenbanken

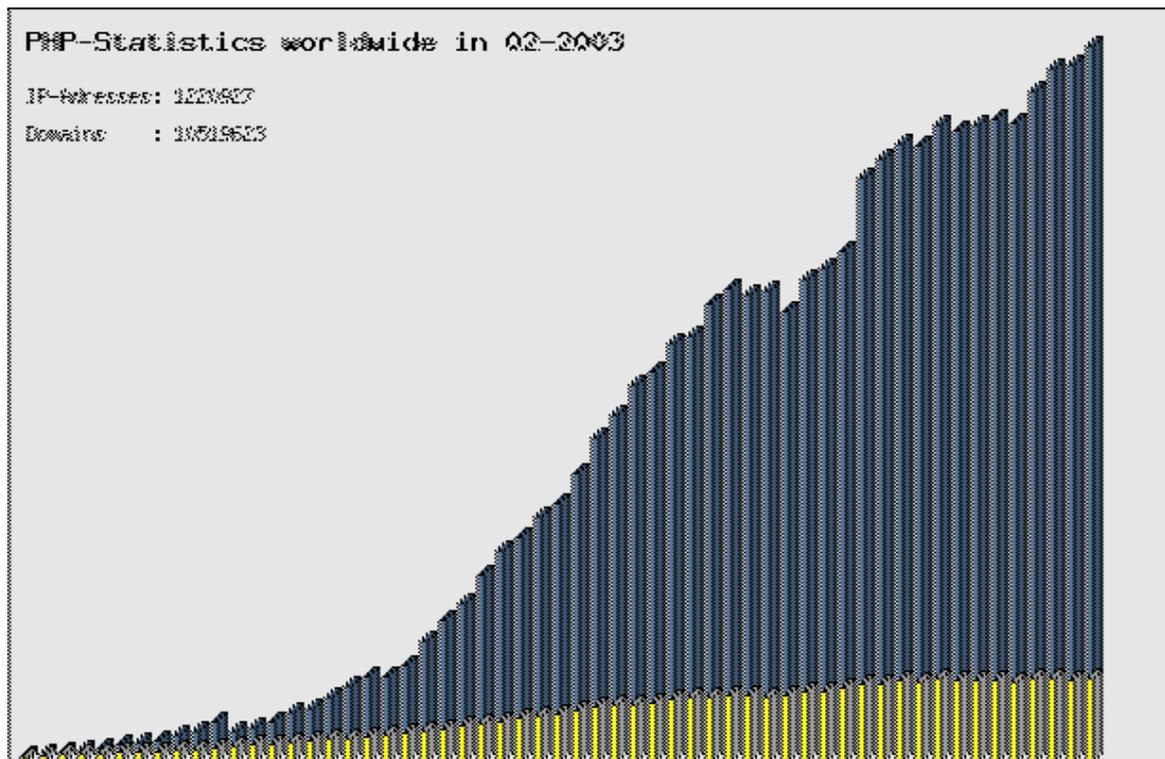


Abbildung 8: Verbreitung von PHP im World Wide Web laut Netcraft<sup>49</sup> und E-Soft<sup>50</sup>

Die Syntax von PHP erinnert an C, Perl und Java und ist im allgemeinen recht einfach zu erlernen. Dabei wird es komplett vom Server ausgeführt und ist somit, im Gegensatz zu clientseitigen Skriptsprachen wie JavaScript, absolut unabhängig von der Soft- und Hardware des Benutzers. Spätestens nachdem man selbst die Problematik von JavaScript im Zusammenhang mit unterschiedlichen Browsern erfahren hat, weiß man diese Eigenart zu schätzen.

Zurückblickend läßt sich feststellen, daß sich durch PHP sogar die ganze Art und Weise, wie Web-Publishing betrieben wird, verändert hat. Denn während bis Mitte der 90er Jahre fast nur statische HTML-Seiten im WWW anzutreffen waren, werden heutzutage viele Webseiten dynamisch, unter Verwendung von Datenbanken, zusammengesetzt, um Inhalte effektiver zu verwalten und individueller darstellen zu können. Der Einsatz von Datenbanken und den damit gekoppelten Skriptsprachen wird zugleich deshalb auch in der nächsten Zukunft zunehmen, weil die Datenmenge im Internet auf unabsehbare Zeit weiter anwachsen wird.

PHP kann in HTML eingebettet werden, es besteht also genau wie die Hypertext Markup Language nur aus reinem Text und kann mit ganz normalen Texteditoren<sup>51</sup> erstellt werden. Damit der Server zwischen HTML- und PHP-Bestandteilen des Dokumentes unterscheiden kann, werden Sektionen mit PHP-Anweisungen durch die Tags „<?php“ und „?>“ abgegrenzt. Zur Veranschaulichung folgt hier eine kleine Demonstration:

<sup>49</sup> Homepage: <http://news.netcraft.com/>

<sup>50</sup> Homepage: [http://www.securityspace.com/s\\_survey/data/index.html](http://www.securityspace.com/s_survey/data/index.html)

<sup>51</sup> Inzwischen ist es auch möglich durch eine kommerzielle Ergänzung PHP-Skripte zu kompilieren.

```
<html>
  <head>
    <title>Meine erste Seite mit PHP</title>
  </head>
  <body>
    <h1>HTML mit PHP</h1>
    <?php
      print "PHP- die Open-Source-Skriptsprache<br>\n";
      print "zur Erstellung dynamischer Webseiten";
    ?>
  </body>
</html>
```

Der Webserver verarbeitet den PHP-Block, bevor er das Dokument an den Client verschickt, beim Browser kommt somit nur reinstes HTML an. In diesem Falle würde der Quelltext beim Betrachter dementsprechend folgendermaßen aussehen:

```
<html>
  <head>
    <title>Meine erste Seite mit PHP</title>
  </head>
  <body>
    <h1>HTML mit PHP</h1>
    PHP- die Open-Source-Skriptsprache<br>
    zur Erstellung dynamischer Webseiten
  </body>
</html>
```

Wie unschwer zu erkennen ist, können also nicht nur Texte von PHP, sondern auch Tags übergeben werden, wodurch tatsächlich ein kompletter Aufbau einer HTML-Seite bewerkstelligt werden kann.

Grundsätzlich muß in PHP jede Variable mit einem vorangestellten Dollarzeichen (\$) gekennzeichnet werden, ansonsten darf der Variablenname beliebig aus Buchstaben, Zahlen und dem Unterstrich zusammengesetzt werden. PHP bietet mit den Datentypen Integer, Double, String, Array und seit der Version 4.0 Boolean und Object, fast alle gebräuchlichen Variablenarten an. Es fehlen Zeiger, dafür gibt es Referenzen, die vor allem für große Arrays und die neu eingeführten Objekte Sinn machen. Diese werden jedoch nicht fix an die jeweilige Variable vergeben, sondern PHP bestimmt, ähnlich wie JavaScript, bei einer Wertzuweisung den dafür passenden Datentyp. Prinzipiell ist dies für den Programmierer ein bequemes Verfahren, jedoch muß er dafür Sorge tragen, daß diese Toleranz seitens des Systems nicht zu unerwarteten Ergebnissen führt und sollte im Quelltext unter Umständen Kommentare bezüglich des beabsichtigten Datentyps einfügen. Alternativ dazu kann PHP jedoch auch dazu gezwungen werden einen bestimmten Datentyp zu verwenden, indem man diesen mit Hilfe der Funktion `settype()` festlegt.

```
...
<?php
  $var=3.56; // PHP weist der Variable $var den Datentyp Double zu
  print $var."<br>\n";
```

```

settype($var,integer); // Datentyp Integer wird für $var erzwungen
print $var."<br>\n";
$var=$var." Chinesen mit dem Kontrabaß";
print $var."<br>\n";?>
...

```

Dieses Skript würde daher zunächst den Double-Wert 3,56 und anschließend den Integer-Wert 3 ausgeben. Dabei ist zu beachten, daß bei der Umwandlung von Fließkomma- in Integer-Zahlen nicht gerundet wird! Durch das Anhängen einer Textkette wandelt PHP dann die Variable automatisch in einen String um.

Arrays treten in PHP in zweierlei Gestalt auf, denn neben dem normalen steht ferner der assoziative Array, auch als *Hash-Array* bezeichnet, zur Verfügung. In letzterem kann ein Wert innerhalb Arrays nicht nur über seinen Index, sondern gleichwohl über ein zugeordnetes Schlüsselwort adressiert werden.

```

...
<?php
    $person = array(name="Peter",beruf="Lehrer",alter="35");
    print $person[beruf]."<br>\n"; // gibt den Wert "Lehrer" aus
?>
...

```

Weiterhin läßt sich in PHP die Größe von Arrays dynamisch verändern. Durch Verschachtelung lassen sich mehrdimensionale Arrays erzeugen.

Durch die Integration des objektorientierten Ansatzes hat sich PHP4 verstärkt als moderne Programmiersprache hervorgetan, auch wenn dieses Konzept in der jetzigen Version sicherlich noch nicht ausgereift ist. Es ist zum Beispiel noch nicht möglich, Attribute oder Methoden von Objekten als privat oder geschützt zu deklarieren, wodurch auch andere Klassen, Funktionen und sonstige Programmbereiche jederzeit freien Zugriff auf diese Elemente haben. In der kommenden Version<sup>52</sup> wird PHP diese Funktionalität voraussichtlich beinhalten, genauso wie die technischen Voraussetzungen zum Überladen von Funktionen und eindeutig standardisierte Konstruktoren und Destruktoren. Aber auch ohne Objektorientierung bietet PHP ein großzügiges Angebot an verfügbaren Kontrollstrukturen, wie Schleifen oder Verzweigungen, sowie prozeduralen Techniken für die Definition und Verwendung eigener Funktionen.

### 3.6 MySQL

MySQL ist das derzeit populärste Open Source SQL-Datenbanksystem. Entwickelt wurde es von der schwedischen Firma T.c.X. DataKonsult<sup>53</sup>, die inzwischen als die Aktiengesellschaft MySQL AB<sup>54</sup> auftritt und von den beiden Schweden David Axmark und Allan Larsson, sowie dem Finnen Michael Widénius 1995 gegründet wurde.

<sup>52</sup> Artikel: [http://www.phpmag.de/itr/online\\_artikel/psecom,id,207,nodeid,62,ps\\_lo,60.html](http://www.phpmag.de/itr/online_artikel/psecom,id,207,nodeid,62,ps_lo,60.html)

<sup>53</sup> Homepage: <http://www.tcx.se/>

<sup>54</sup> Homepage: [www.mysql.de/](http://www.mysql.de/)

Ausgehend von der mSQL-Datenbank, mit der schon Lerdorf im Rahmen seines PHP/FI-Projektes gearbeitet hatte, entstand mit MySQL ein schnelles, zuverlässiges und vor allem leicht zugängliches Werkzeug für den Umgang mit großen Datenmengen.

MySQL ist *ein relationales Datenbank-Managementsystem*, kurz: RDBMS, das heißt, daß die Daten in separaten Tabellen gespeichert werden, um die Geschwindigkeit, aber auch die Flexibilität zu erhöhen. Durch festgelegte Beziehungen, den Relationen, können diese Tabellen miteinander verknüpft werden, wodurch Daten aus mehreren verschiedenen Tabellen auf Anfrage miteinander verknüpft werden können. Das Kürzel SQL im Namen steht für die Bezeichnung *Structured Query Language*<sup>55</sup>, was ganz frei als strukturierte Abfragesprache übersetzt werden kann, und als allgemeiner Standard für relationale Datenbanken gilt. MySQL, inzwischen in der Version 4.0.12 erhältlich, gibt es für eine große Palette, der zur Zeit verwendeten Betriebssysteme, wie Linux, Windows, Solaris, FreeBSD, Mac OS X und HP-UX.

Im Zuge der starken Verbreitung von MySQL erschienen immer mehr nützliche Zusatzprogramme zur Erleichterung der Datenbankverwaltung. Das bekannteste ist das gegenwärtig in Version 2.5.0 vorliegende Open Source Werkzeug *phpMyAdmin*. Dieses kann ebenfalls kostenlos heruntergeladen<sup>56</sup> werden und beruht, wie der Name schon andeutet, auf einer Ansammlung von PHP-Skripten, mit denen eine bequeme Verwaltung der Datenbank bewerkstelligt werden kann.

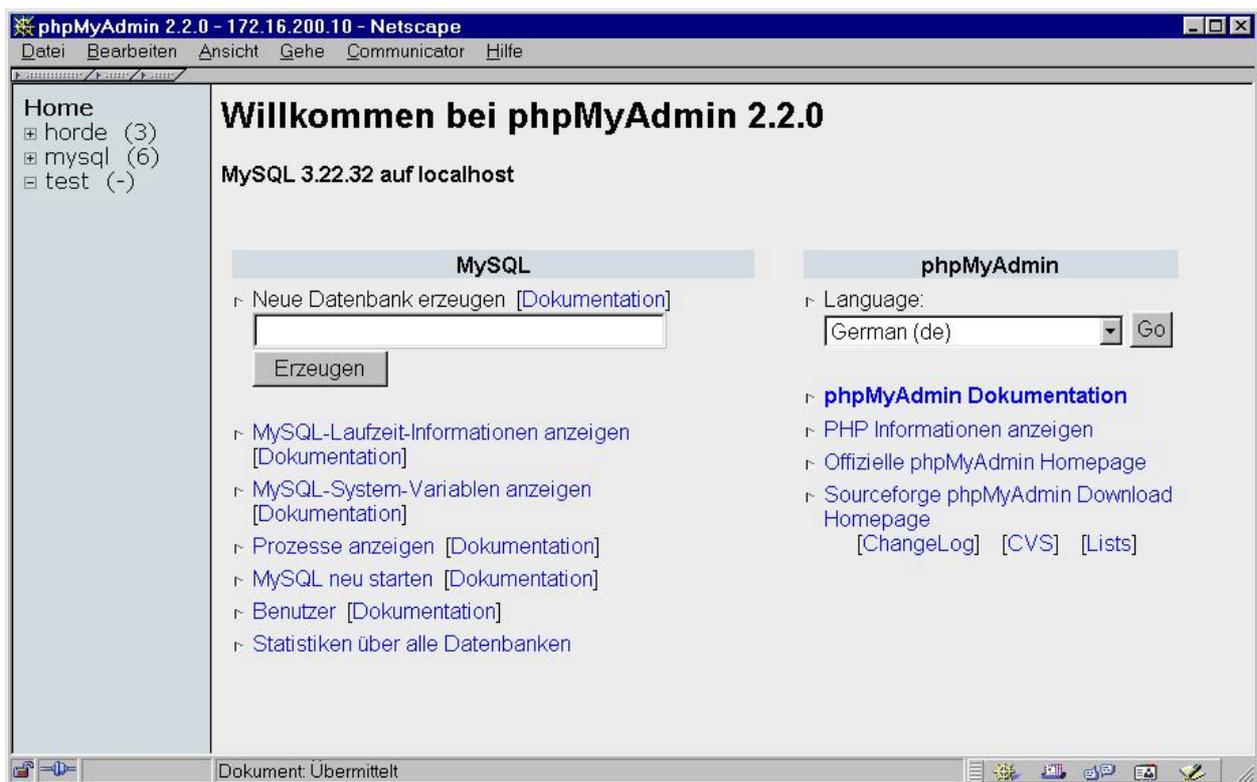


Abbildung 9: phpMyAdmin

<sup>55</sup> In den 70er Jahren von IBM ursprünglich für relationale Datenbank DB2 entwickelte Abfragesprache. Nach der Schaffung des gemeinsamen Standards ANSI-92 konnte sich SQL als die Abfragesprache im Bereich relationaler Datenbanken etablieren.

<sup>56</sup> Bezugsquelle: <http://www.phpmyadmin.net/>

Des Weiteren gibt es für Windows-Systeme das *WinMySQLAdmin-Tool*<sup>57</sup>, welches mit einer graphischen Oberfläche für die grundlegenden Einstellungen, wie etwa Speicherpfade und Benutzerrechte, sowie für das Anlegen und Löschen von Datenbanken in MySQL aufwartet.

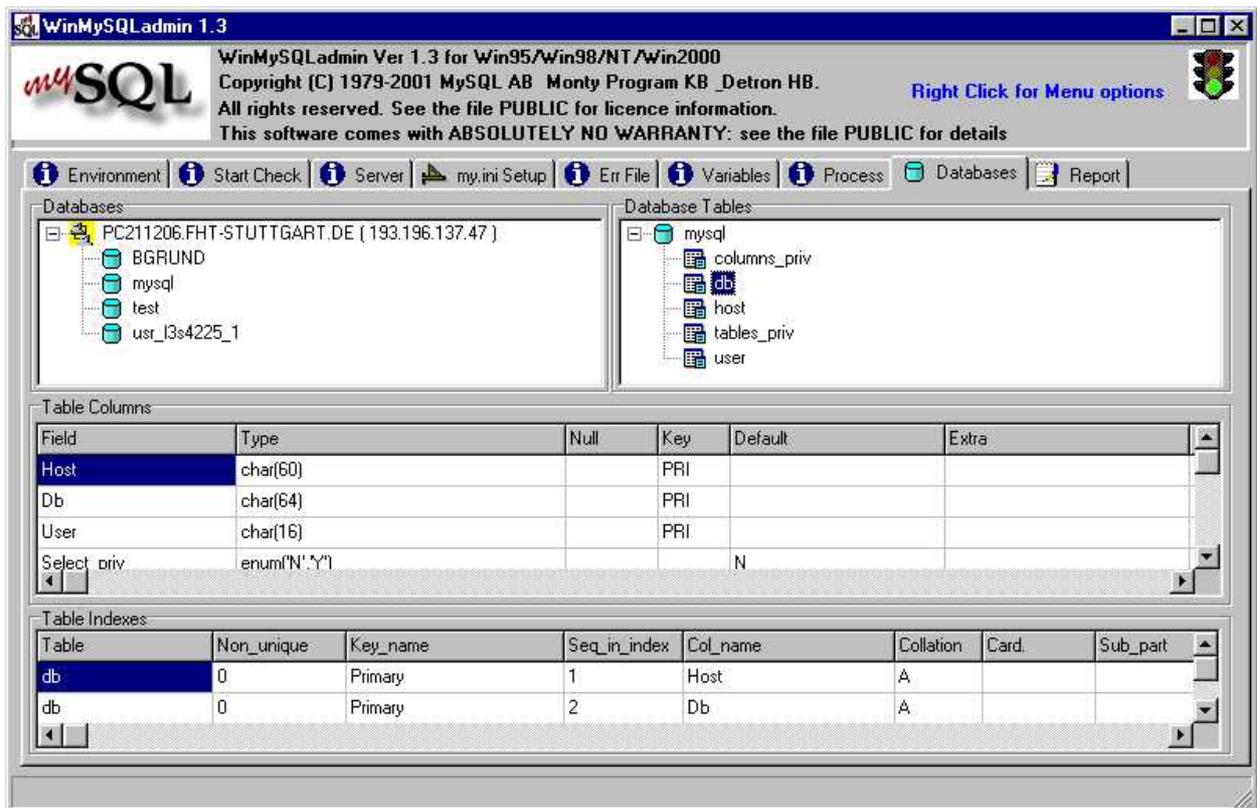


Abbildung 10: WinMySQLAdmin

PHP liefert eine Vielzahl an Funktionen für MySQL, so daß sich Abfragen damit sehr gut realisieren lassen.

...

```
<?php
```

```

$benutzer = "gast";
$password = "sagichnicht";
$db = "kundendb";
$link = mysql_connect("localhost", $benutzer, $password);
if(!link) die ("Keine Verbindung zu MySQL");
mysql_select_db($db) or die ("Konnte Datenbank nicht öffnen");
$abfrage = "SELECT strasse,hausNr FROM adressen
           WHERE(kundenID = $kundenID)";
$ergebnis = mysql_query($abfrage);
if(mysql_num_rows($ergebnis)==1) {
    $adresse = my_sql_fetch_row($ergebnis);
    $strasse = $adresse[0];

```

<sup>57</sup> Bezugsquelle: <http://www.mysql.de/downloads/os-win32.html>

```

    $hausNr = $adresse[1];
    print "Straße: $strasse, Hausnummer: $hausNr<br>\n";
} else
    print "Kein eindeutiger Datensatz gefunden!<br>\n";
mysql_close($link);
?>
...

```

So würde dieses Skript eine Verbindung zum lokal eingerichteten MySQL-Datenbankserver aufbauen, unter gleichzeitiger Anmeldung als Benutzer „gast“ mit dem Paßwort „sagichnicht“. Anschließend wird die gewünschte Datenbank, in diesem Falle „kundendb“, ausgewählt und eine Abfrage nach den Feldern „strasse“ und „hausNr“ in der Tabelle „adressen“ ausgeführt. Wenn ein Datensatz mit übereinstimmender Kunden-ID gefunden wird, gibt das Skript die entsprechenden Werte für Straße und Hausnummer aus.

### 3.7 XML – Extensible Markup Language

Die *Extensible Markup Language* ist eine *Meta*<sup>58</sup>-Auszeichnungssprache zur Beschreibung strukturierter Daten. Der besondere Vorteil des Ansatzes besteht dabei darin, daß diese Meta-Sprache auf verschiedenen Betriebssystemplattformen und in verschiedenen Anwendungen eingesetzt werden kann.

Um dies zu veranschaulichen, braucht nur das hier vorliegende Word-Dokument als Beispiel dafür betrachtet werden. Solange man die zugehörige Software - Microsoft Word – besitzt gibt es, abgesehen von einigen Versionsunterschieden, keine Probleme diese Datei zu benutzen. Soll jedoch dieses Dokument in einem anderen Programm angezeigt werden, muß dieses zu Word voll kompatibel sein oder eine entsprechende Konvertierungsfunktion besitzen, wodurch meistens Informationen verloren gehen oder verfälscht werden. Wenn dann noch zusätzlich das Betriebssystem gewechselt wird, schränkt dies die Verwendbarkeit weiter ein. Außerdem ist ein Word-Dokument hauptsächlich auf die Ausgabe in Papierform ausgerichtet. Natürlich wird auch die Darstellung mittels Bildschirm ausreichend berücksichtigt, doch schon bei einer Publikation als Webseite ist das benutzte Format überfordert. Von sonstigen Alternativen, wie der Ausgabe auf einem Handy-Display, ganz zu schweigen. Überdies verschmelzen in einem Word-Dokument Inhalt und Formatierung, so daß sie nicht ohne weiteres voneinander getrennt behandelt werden können.

All dies sollte mit der Extensible Markup Language verbessert werden, die Daten werden einmal zentral strukturiert, verständlich für sämtliche Systeme und Anwendungen und bei der Ausgabe, entsprechend des gewünschten Mediums, formatiert. Als Nebeneffekt der unabhängigen Datenstruktur und dem Aufbau aus reinem ASCII-Text (analog zu HTML), hat sich XML inzwischen jedoch auch zu einem immer wichtiger werdenden Austauschformat für elektronische Daten entwickelt.

Die Vorgänge, die später zur Entwicklung von XML führen sollten, reichen zurück bis in die 60er Jahre des 20. Jahrhunderts. Gegen Ende dieses Jahrzehntes hatten schon mehrere Firmen erhebliche Schwierigkeiten bei der Verwaltung ihrer umfangreichen elektronischen Archive. Eine dieser Firmen war IBM, und so machte sich der IBM-Mitarbeiter Charles Goldfarb daran, eine Dokumentenbeschreibung zu entwi-

---

<sup>58</sup> meta... (griech.): Wortbestandteil mit der Bedeutung „zwischen, nach, später“

ckeln, die anwendungs- und geräteunabhängig ist und weder Formatierung noch Stil des Dokumentes abbildet, sondern sich alleine und ausschließlich mit der Struktur, bestehend aus Seiten, Kapiteln, Abschnitten etc. befaßt. Aus diesen Untersuchungen entstand die *Generalized Markup Language* (GML).

Andere Unternehmen entwarfen in der Folgezeit ihre eigenen Standards, wodurch die eigentliche Hauptabsicht des uneingeschränkten Datenaustausches zunichte gemacht wurde. Um zu einem gemeinsamen Nenner zu kommen, wurde Mitte der 1980er Jahre ein ISO-Komitee mit der herstellerübergreifenden Standardisierung einer Dokumentenbeschreibungssprache beauftragt. Das Resultat war die 1986 veröffentlichte *Standard Generalized Markup Language*, oder kurz: *SGML*. In diesem Standard wurden nicht nur die Regeln zur Beschreibung der Dokumentstruktur, sondern auch Grundsätze zur Identifikation der im Dokument benutzten Zeichen und Objekte oder zur Einbindung externer Daten festgelegt.

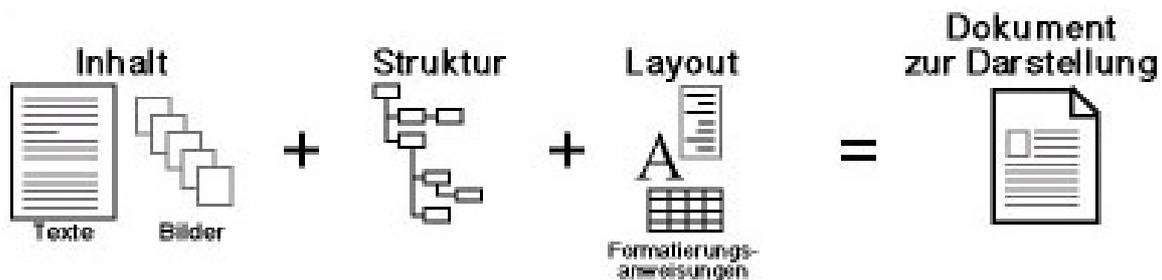


Abbildung 11: Aufteilung in Inhalt, Struktur und Layout

Da SGML allerdings für äußerst umfangreiche Dokumentensammlungen sehr großer Firmen gedacht war, war es dementsprechend umfangreich. Die Implementierungen dazugehöriger Anwendungen wurden sehr aufwendig und damit teuer. Kleineren bis mittleren Firmen stand somit immer noch kein passendes Werkzeug für die Verwaltung ihrer elektronischen Daten zur Verfügung. HTML dagegen, im übrigen selbst eine Anwendung von SGML, war durch die fest vorgegebenen Tags (ca. 70) und Attribute (ca. 50) viel zu begrenzt um diese Aufgabe zu übernehmen.

XML wurde Anfang 1998 als idealer Kompromiß zwischen SGML und HTML vom W3 Konsortium nach zwei Jahren Entwicklungszeit unter dem kanadischen Chefentwickler Tim Bray zum offiziellen Standard erhoben. Schon im Jahr zuvor hatte Microsoft mit dem Internet Explorer 4 erste Ansätze für eine Umsetzung hervorgebracht, doch erst 1999 wurde der neue Standard mit der Version 5 ausreichend unterstützt. Seit der Version 6 kann der Netscape-Browser ebenfalls damit umgehen, Opera ab Version 4, und der unter LINUX häufig eingesetzte Konqueror besitzen die benötigten Voraussetzungen gleichermaßen. Doch auch abseits der Webbrowser scheint sich XML immer mehr durchzusetzen. Die Zeichen dafür stehen recht gut, denn alle großen Unternehmen im Hard- und Softwarebereich, wie IBM, Netscape, Microsoft, SUN, SAP und andere, engagieren sich sehr stark in dem neuen Format. Aber auch vor allem im Geoinformatikbereich stehen die Voraussetzungen sehr günstig. So hat zum Beispiel das *Open-Gis-Consortium* (OGC) eine XML-Spezifikation namens *Geography Markup Language* (GML) erarbeitet, und gerade mit dem Vektorgrafikstandard SVG scheinen zukünftig noch sehr viele Möglichkeiten offen zu stehen um Geodaten im Internet zu präsentieren.

Prinzipiell sind bei XML folgende Vorteile erkennbar:

- Uneingeschränkte Erweiterbarkeit

- Trennung von Inhalt, Struktur und Layout
- Crosspublishing möglich, d.h. es existiert nur ein Dokument für verschiedene Ausgabeformen
- Überprüfung der Syntax und Semantik möglich (Validierung<sup>59</sup>)
- Offiziell verabschiedeter Standard vom W3C
- Erweiterte Linktechnologie (uni-, bi- und multidirektionale Links)
- Für Computer und Mensch aufgrund der Textbasiertheit verständliches Datenaustauschformat)
- Internationale Standardisierung durch UNICODE<sup>60</sup>-Zeichensatz
- Objektorientierter Ansatz kann interne Strukturen darstellen
- Einbinden in andere Dokumente
- Metadatenstrukturierung für bessere Suchergebnisse
- XML-DOM für Dokumentenverarbeitung
- Spezifische Komponenten für jede Anwendung (XPath, Xlink, XQL etc.)
- Benutzerspezifische Darstellungsmöglichkeiten

Da XML quasi ein reduziertes SGML darstellt und HTML durch SGML definiert wurde, ähneln sich beide Formate in ihrem Aussehen. Daß XML aber ungleich mächtiger ist, läßt sich daran erkennen, daß inzwischen auf seiner Grundlage HTML neu als XHTML beschrieben wurde. Und genau dafür steht der Begriff Meta-Sprache, also eine Sprache mit der andere Sprachen erschaffen werden können. Weitere Beispiele dafür sind *WML*<sup>61</sup>, *XSL*<sup>62</sup>, *XLink*<sup>63</sup>, *SMIL*<sup>64</sup>, *VML*<sup>65</sup>, *MathML*<sup>66</sup> und natürlich SVG.

Für diese und alle anderen, mit XML erschaffenen Sprachen, müssen ganz spezielle Regeln bezüglich der verwendbaren Elemente, deren zugehörige Attribute und Verschachtelungsmöglichkeiten angegeben werden. Dies erfolgt in der *Dokumenttyp-Definition*, kurz DTD<sup>67</sup>, die entweder innerhalb des XML-Dokumentes oder in einer referenzierten, externen Datei aufgeführt werden kann. DTDs werden jedoch nur benötigt, um ein gemeinsames Vokabular zu etablieren, um Bearbeitungswerkzeuge bei der Erzeugung gültiger XML-Dokumente zu unterstützen oder um die Validität eines solchen zu überprüfen. Für die Verarbeitung oder die Anzeige werden sie dagegen nicht gebraucht.

---

<sup>59</sup> valid (engl.): gültig

<sup>60</sup> Internationaler Standard zur Kodierung aller Schriftzeichen der Welt

<sup>61</sup> Wireless Markup Language: Beschreibungssprache für das WAP-Protokoll, welches die Darstellung von Internet-Inhalten auf dem Handy ermöglicht

<sup>62</sup> Extended Style Sheet Language: Gegenstück zu CSS für Formatierungen von XML-Dokumenten

<sup>63</sup> XML Linking Language: für die Definition von Verknüpfungen zwischen XML-Dokumenten

<sup>64</sup> Synchronized Multimedia Integration Language: zur Beschreibung von Multimedia-Effekten

<sup>65</sup> Vector Markup Language: von Microsoft definierte Beschreibung von Vektorgraphiken

<sup>66</sup> für die Darstellung mathematischer Formeln

<sup>67</sup> Document Type Definition

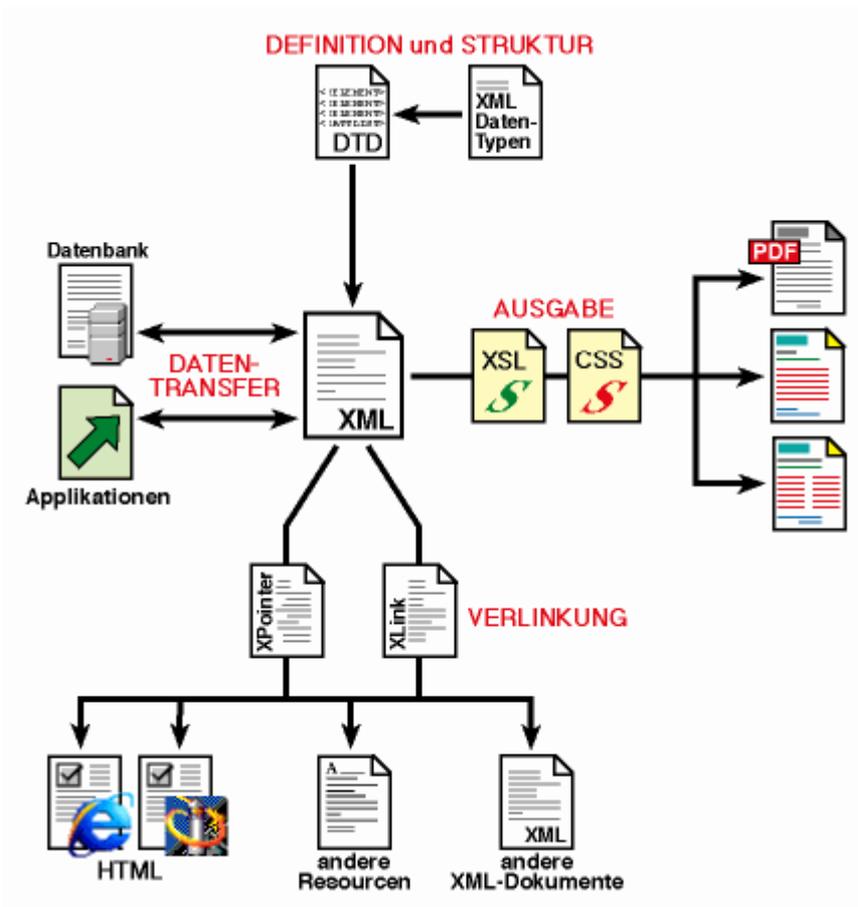


Abbildung 12: Das Konzept von XML

Ein XML-Dokument besteht immer aus zwei Teilen, dem Prolog und dem Dokumentinhalt mit den eigentlichen Daten. Der Prolog setzt sich zusammen aus der optionalen Kopfzeile, sowie einer eventuellen Document Type Definition bzw. einem Verweis auf eine solche.

```
<?xml version="1.0"?>
<!DOCTYPE test system "test.dtd">
<personen>
  <person>
    <name>Bach</name>
    <vorname>Hans</vorname>
    <alter>42</alter>
  </person>
  <person>
    <name>Meier</name>
    <vorname>Judith</vorname>
    <alter>24</alter>
  </person>
</personen>
```

Die erste Zeile enthält den Header, der die Datei als XML-Dokument kennzeichnet. Auf diese Weise erkennt der Parser, daß er diesen Teil verarbeiten soll. Zusätzlich wird die verwendete Version mit ange-

geben, Version 1.0 liegt seit Oktober 2000 in einer überarbeiteten Fassung<sup>68</sup> vor. In der nächsten Zeile wird optional die zugrundeliegende DTD angeführt, in diesem Fall liegt diese in einer eigenen Datei namens „test.dtd“. Alternativ dazu ließen sich hier die Angaben auch direkt benennen.

```
<!ELEMENT personen (person)+>
<!ELEMENT person (name,vorname,alter)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT vorname (#PCDATA)>
<!ELEMENT alter (#PCDATA)>
```

Daraus ergibt sich, daß das personen-Element ein oder mehrere Elemente „person“ besitzen kann, die wiederum die Elemente „name“, „vorname“ und „alter“ beinhalten.

Der eigentliche Dokumenteninhalte findet sich in den restlichen Zeilen, bei diesem Beispiel also zwei Personen mit ein paar persönlichen Angaben. Dabei ist vor allem auf die Reihenfolge der einzelnen Tags, auch Delimiter genannt, zu achten, um die vorgegebenen Verschachtelungsregeln einzuhalten.

Eine andere Möglichkeit Eigenschaften einer Person festzuhalten besteht über die Angabe von Attributen. Jedes Element mit Attributen benötigt dafür eine eigene, ebenfalls in der DTD anzugebende, Attributdefinition.

```
<!ELEMENT person (name,vorname,alter)>
<!ATTLIST person
  id ID #REQUIRED
  weiblich (selected|disabled) #REQUIRED "selected"
  name CDATA #IMPLIED
  vorname CDATA #IMPLIED
  alter CDATA #IMPLIED
>
```

Man sieht hier also eine Reihe von Eigenschaften, die einer Instanz des Elementes „person“ zugewiesen werden können (IMPLIED), oder sogar müssen (REQUIRED). Sie könnten auch als unveränderliche Parameter bestimmt werden (FIX). Die Angaben in der zweiten Spalte stehen für die Wertebereiche, die den einzelnen Attributen zugewiesen werden können, ID steht für einen eindeutigen Identifikator, CDATA für einen beliebigen String, und durch das Piping-Zeichen ( | ) getrennte Werte innerhalb einer Klammer kennzeichnen eine fest vorgegebene Auswahlmenge. Zusätzlich können für jedes Attribut Standardwerte (hier: weiblich = selected) angegeben werden.

Für den Anfänger in XML, der jedoch schon Erfahrungen mit HTML gesammelt hat, gilt es vor allem die Wohlgeformtheit seiner Arbeiten zu beachten. Denn im Gegensatz zu HTML und dessen Ursprung SGML ist XML kein bißchen tolerant was die Form betrifft. So kümmerte sich HTML nicht im geringsten um Groß- und Kleinschreibung. XML dagegen ist case-sensitiv, und somit sind hier die Angaben <Person> und <person> zwei unterschiedliche Informationen.

Dazu kommt, daß alle Attribute in Anführungszeichen stehen müssen. Genügte also in HTML die Anweisung color = blue, muß es in XML richtig heißen color = "blue". Ein weiterer Stolperstein für Neuanfänger mit vorhandenen HTML-Kenntnissen ist die Tatsache, daß es in XML keine alleinstehenden

<sup>68</sup> Spezifikation unter: <http://www.w3.org/TR/REC-xml>

henden Tags geben darf, sondern immer nur Paare. Dies läßt sich jedoch nicht in jedem Fall so umsetzen, weswegen zumindest anstelle eines „<br>“-Steuerzeichens in HTML das „<br/>“-Tag in XML verwendet werden muß, also ergänzt um einen nachgestellten Slash bzw. Schrägstrich.

## 3.8 SVG – Scalable Vector Graphics

### 3.8.1 Einführung

Bisher wurden im Internet Grafiken hauptsächlich als Rasterdaten, wie beispielsweise in den Formaten *GIF*<sup>69</sup>, *JPEG*<sup>70</sup>, *BMP*<sup>71</sup>, *PNG*<sup>72</sup> oder *TIFF*<sup>73</sup>, verwendet. Mit dem Aufkommen der Scalable Vector Graphics, einem zweidimensionalen Vektorgrafikstandard, könnte sich das jedoch entscheidend ändern. SVG ist ein Ableger von XML und stellt damit ebenfalls ein Klartextformat dar, was die Erstellung und Bearbeitung anhand eines herkömmlichen Texteditors erlaubt. Als weiteren Vorteil gegenüber Rastergrafiken lassen sich SVG-Grafiken ohne Qualitätsverlust skalieren und lassen sich durch das integrierte DOM animieren. Die erzeugten Dateien sind trotz dessen kleiner als vergleichbare GIF-, JPEG- oder PNG-Dateien und lassen sich sogar noch zusätzlich komprimieren. Zu HTML 4.0 ist es weitgehend und zu XHTML vollständig kompatibel, Inhalte lassen sich mit CSS oder XSL formatieren. Gegenüber dem proprietären Format *Flash* von Macromedia besticht es gerade durch die Unabhängigkeit von spezieller Software, wobei jedoch momentan, zumindest für die Betrachtung im Internet-Browser, ein spezielles Plug-In benötigt wird. Mit der zunehmenden Verbreitung von SVG kann allerdings davon ausgegangen werden, daß zukünftige Browser-Versionen selbständig mit diesen Daten umgehen können.

### 3.8.2 Entstehung

Im April 1998 legten die Unternehmen Adobe, IBM, Netscape und SUN dem W3C einen Entwurf für die *Precision Graphics Markup Language*, kurz: PGML, beim W3-Konsortium vor, welche stark an dem Post-Script-Standard angelegt war und Vorteile bei den Gestaltungsmöglichkeiten und Animationen von Grafiken vorweisen konnte. Schon einen Monat später zogen Hewlett Packard, Microsoft, Macromedia und Visio mit ihrer Lösung, der *Vector Markup Language* (VML) nach, wobei hier die Schwerpunkte auf technische Aspekte, wie beispielsweise Einbindung mathematischer Funktionen lagen. Des weiteren zeigten PRP und Orange PCSL im Juni des selben Jahres mit ihrer *Hyper Graphics Markup Language* (HGML) interessante Ansätze bezüglich Kopieren, Transformieren und Einfügen von Bildausschnitten.

Basierend auf diesen individuellen Modellen, als Kombination der jeweiligen Vorzüge und mit weiteren Verbesserungen und Ergänzungen, entstand im Oktober die Sprache SVG. Allerdings dauerte es noch eine ganze Weile, bis schließlich am 4. September 2001 die erste offizielle Empfehlung des W3C vorlag. Seit Anfang diesen Jahres liegt Version 1.1<sup>74</sup> vor, und die vom WWW-Konsortium ins Leben gerufene

---

<sup>69</sup> Graphics Interchange Format

<sup>70</sup> Joint Photographic Expert Group

<sup>71</sup> Bitmap

<sup>72</sup> Portable Network Graphic Format

<sup>73</sup> Tagged Image File Format

<sup>74</sup> Spezifikation unter: <http://www.w3.org/TR/SVG11/>

Arbeitsgruppe, mit Beteiligung von Adobe, Apple, Autodesk, Canon, Corel, HP, IBM, Kodak, Macromedia, Microsoft, Netscape/AOL, Quark, SUN und Xerox, arbeitet momentan an Version 1.2. Schon allein die Liste der beteiligten Unternehmen dürfte ein eindeutiger Hinweis für die glänzenden Zukunftsaussichten der Scalable Vector Graphics sein.

### 3.8.3 Der Adobe SVG Viewer

Von diesen Unternehmen hat sich für den normalen Benutzer von SVG-Grafiken vor allem Adobe hervorgetan, da es mit dem SVG Viewer<sup>75</sup> das wohl gängigste Plug-In für den Netscape Navigator (ab 4.0), Netscape 6.x, Internet Explorer (ab 4.0) und Opera (ab 5.x) anbietet. Ferner erweitert es den RealPlayer 8 um die SVG-Unterstützung. Unterstützt werden dabei Windows und MacOS-Plattformen, für Solaris 8 und RedHat-Linux liegen Betaversionen der Software vor. Die Windows-Variante der aktuellen Version 3.0 ist immerhin 2,5 Megabyte groß, jedoch ist hierbei eine eigene Script-Engine integriert, die bei Bedarf anstelle des eigentlichen Internet-Browsers die Laufzeitumgebung für integrierte Skriptsprachen darstellt, und somit einen besseren Zugriff auf die einzelnen SVG-Elemente erlaubt. Überdies werden dem Betrachter mittels Kontextmenü<sup>76</sup> verschiedene Funktionen zur Verfügung gestellt.

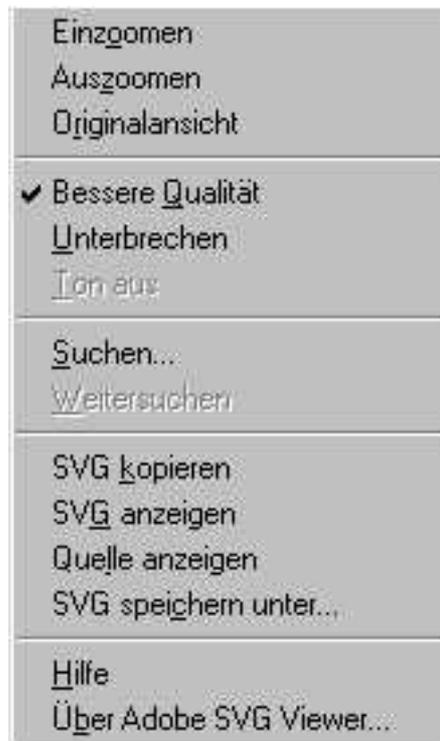


Abbildung 13: Kontextmenü des SVG Viewers von Adobe

- Das Ein- und Auszoomen der Grafik, plus der Wiederherstellung der Originalansicht.

<sup>75</sup> Bezugsquelle: <http://www.adobe.com/svg/viewer/install/main.html>

<sup>76</sup> Bei Windows und Unix aufrufbar durch Klick der rechten Maustaste, bei Mac durch Kombination von Ctrl- und linker Maustaste

- Das Aktivieren und Deaktivieren von Anti-Aliasing<sup>77</sup>

- ohne Anti-Aliasing



- mit Anti-Aliasing



- Das Unterbrechen bzw. Starten von Animationen. Bei Tonausgabe kann diese ebenfalls ein- und ausgeschaltet werden.
- Die Suche nach Texten
- Die Anzeige des Quelltextes
- Wenn davon ausgegangen wird, daß der künftige Betrachter den Viewer von Adobe benützt, kann man innerhalb des SVG-Dokumentes ein individuelles Kontextmenü erstellen, das dann ganz eigene Funktionen alternativ oder zusätzlich präsentiert.

Zudem läßt sich unter Benutzung von Tastenkombinationen ebenfalls der Bildausschnitt verändern.

Tabelle 5: Tastenfunktionen für den SVG Viewer von Adobe

Verschieben des Ausschnittes	Alt- und linke Maustaste gedrückt halten
Einzoomen	Strg <sup>78</sup> -Taste gedrückt halten und mit linker Maustaste klicken Strg- und linke Maustaste gedrückt halten und damit den gewünschten Abbildungsrahmen aufziehen
Auszoomen	Strg- und Shift-Taste gedrückt halten und mit linker Maustaste klicken

### 3.8.4 Das SVG-Dokument

Das Grundgerüst eines SVG-Dokumentes sollte immer aus den folgenden Angaben bestehen:

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no">
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg width="500" height="300" xmlns=http://www.w3.org/2000/svg
xmlns:xlink=http://www.w3.org/1999/xlink>
<!-- *** Hier folgt die eigentliche Grafik *** -->
</svg>
```

Der Anfang der ersten Zeile stellt den XML-Prolog dar und muß gleich zu Anfang einer SVG-Datei stehen, damit der Parser weiß, daß es sich hier um ein XML-Dokument handelt.

<sup>77</sup> Kantenglättung zur Vermeidung des „Treppeneffektes“ bei schrägen und gekrümmten Linien

<sup>78</sup> Bei Mac-Rechnern Befehl-Taste anstelle Strg

Die Angaben `encoding` und `standalone` sind optional, sollten aber angebracht werden, um zum einen den verwendeten Zeichensatz, in diesem Falle ISO-8859-1, mit den schriftspezifischen Zeichen für westeuropäischen und amerikanischen Sprachen anzugeben, und zum anderen um eine Aussage darüber zu tätigen, ob das vorliegende Dokument mit einem anderen XML-Dokument, in den meisten Fällen einer DTD, zusammenhängt.

Liegt wie in diesem Beispiel, eine entsprechende DTD vor, kann auf diese über `DOCTYPE` verwiesen werden. Dabei wird zusätzlich angegeben, ob die referenzierte DTD öffentlich zugänglich (`PUBLIC`) oder auf einem privaten Rechner (`SYSTEM`) abgelegt ist. Anschließend folgt der Name der Document Typ Definition (`-//W3C//DTD SVG 20010904//EN`) und ihre URL<sup>79</sup>. Es können aber auch weitere Elemente oder Entitäten angeführt werden, die dann speziell für das vorliegende Dokument Gültigkeit besitzen.

```
<!DOCTYPE ... svg10.dtd" [  
<!ELEMENT Hallo (#PCDATA) >  
<!ENTITY Author "Albert Maier">  
>
```

Das erste Element, das nach dem XML-Prolog folgen muß und sämtliche andere Inhalte einschließt, ist das SVG-Element. Aufgrund der Tatsache, daß es in der Objekthierarchie direkt nach dem Dokument selbst folgt, wird es als *Root<sup>80</sup>-Element* bezeichnet. Es kann wiederum mehrere andere SVG-Elemente beinhalten, muß aber stets als letztes wieder geschlossen werden.

Die wichtigsten Attribute innerhalb eines `svg`-Tags sind einerseits die Angaben in Form von `width` und `height` für die Anzeigegröße im Browser bzw. Ausgabegröße beim Drucken, als auch optionale *Namensraum*-Deklarationen. SVG unterstützt mehrere relative (u.a. Pixel, Prozent) und absolute (Inch, Zentimeter, Millimeter usw.) Einheiten, mit denen beispielsweise die Ausdehnung des SVG-Anzeigebereiches, dem sogenannten *Viewport*, und von graphischen Elementen angegeben werden können. Jedoch kann nicht davon ausgegangen werden, daß ein SVG-Element mit Zentimeterangaben in der exakten Größe auf dem Bildschirm ausgegeben wird. Dies scheint technisch nicht umsetzbar zu sein. Beim Ausdruck jedoch werden diese Vorgaben teilweise (abhängig vom verwendeten Browser) berücksichtigt, so daß maßstäbige Karten und Pläne möglich sind.

Durch die Verwendung von Namensräumen (namespaces) können SVG-fremde Inhalte miteingebunden werden. Diese müssen jedoch mittels einer anderen, von XML abstammenden Sprache definiert sein. Genauso wie man also somit innerhalb eines XHTML-Dokumentes eine SVG-Graphik einfügen kann, lassen sich in SVG beispielsweise mittels des `XLink`-Namensraumes Links zu bestimmten Stellen des Dokumentes oder gar zu anderen Ressourcen erzeugen. Dies entspräche ganz den gewohnten Links, die von HTML her bekannt sind, durch die Spezifikation der ungleich mächtigeren `XLink81`-Sprache ergeben sich jedoch viel mehr Möglichkeiten.

---

<sup>79</sup> Uniform Resource Locator; eindeutige Angabe für ein Dokument im Internet, enthält Art des Dienstes (`http`, `ftp`, `news` usw.), die Serveradresse inkl. der Domain (`com`, `org`, usw.) bzw. Landeskenntung (`de`, `ch`, `uk`), optional den verwendeten Port (z.B. 80), den Pfad auf dem Server und den Namen der Datei.

<sup>80</sup> root (engl.): Stamm, Ursprung, Wurzel, Haupt-

<sup>81</sup> Spezifikation unter: <http://www.w3.org/XML/Linking>

Normalerweise läßt sich bei der Verwendung des SVG Viewers von Adobe die Graphik wie oben erwähnt zoomen und verschieben. Soll dies jedoch verhindert werden, kann man dem SVG-Element das Attribut `zoomAndPan="disable"` mitgeben, wodurch die Graphik auf den ersten Blick statisch wirkt. Leider läßt sich diese Eigenschaft nicht für jedes einzelne SVG-Element festhalten, sondern es gilt immer die des Root-Elements als maßgebend.

### 3.8.5 Die ViewBox

Um gezielt Bildausschnitte einer Graphik definieren zu können, steht einem in SVG die *ViewBox* zur Verfügung. Diese wird ebenfalls als ein Attribut des SVG-Elementes angegeben und erzeugt ein Rechteck, mit dessen x- und y-Wert, sowie seiner Höhe und Breite der gewünschte Bildausschnitt plazierte werden kann. Abhängig von den Höhen- und Breitenangaben der ViewBox im Verhältnis zu denen des meist übergeordneten SVG-Elementes ergibt sich ein Abbildungsmaßstab. Dabei läßt sich steuern, ob dabei der abgebildete Ausschnitt verzerrt werden darf oder eine proportionale Darstellung eingehalten werden muß. Soll eine proportionale Abbildung erfolgen, kann weitergehend bestimmt werden, ob die Grafik auf das ViewBox-Rechteck skaliert wird oder überstehende Teile abgeschnitten werden.

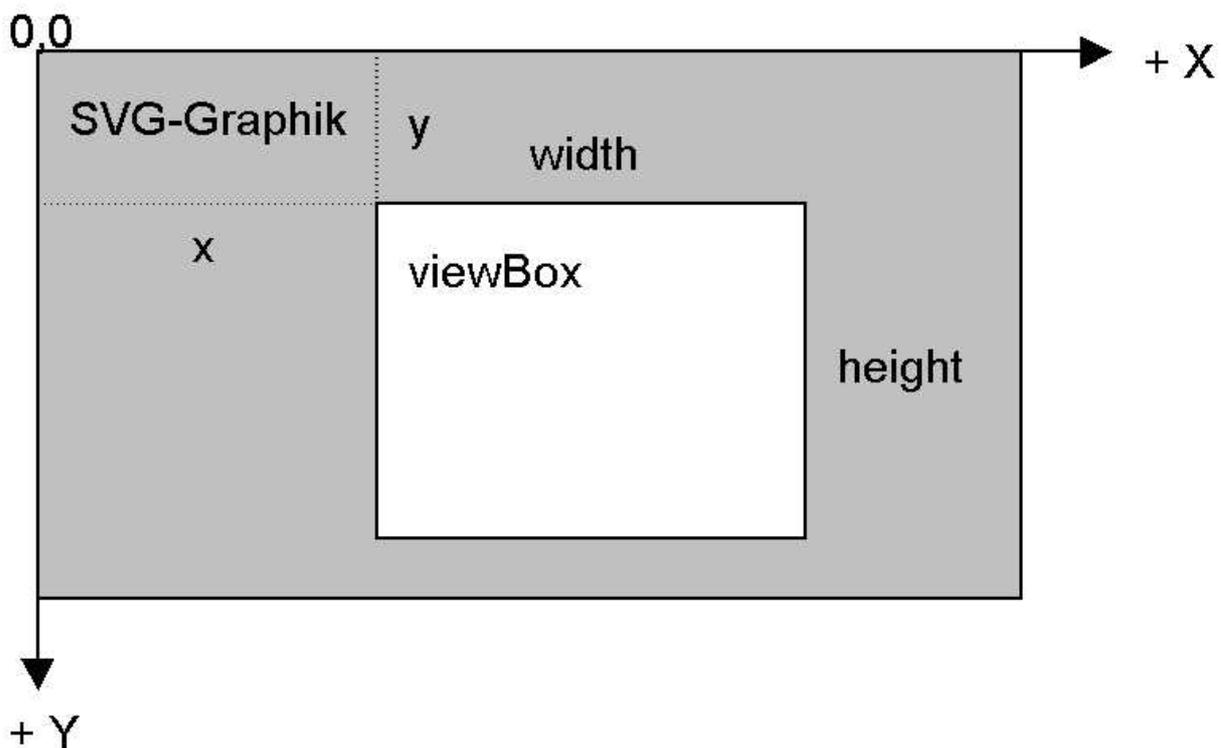


Abbildung 14: Die ViewBox in SVG

Die Parameter für die ViewBox werden folgenderweise angegeben:

```
<svg width="100" height="200" viewBox="60 20 80 60">
```

Dabei stehen die ersten beiden Werte für die x- und y-Koordinate des linken, oberen Eckpunktes der ViewBox, die dritte Angabe für die Breite und der letzte Wert für die Höhe.

Für die Festlegung der Seitenverhältnisse wird das Attribut `preserveAspectRatio` verwendet. Wird dieses auf den Wert `none` gesetzt, wird die Grafik an die vorgegebenen Ausdehnungen angepaßt, selbst wenn dadurch Verzerrungen entstehen. Bei der Angabe `meet` wird die Grafik gegebenenfalls proportional verkleinert, so daß sie komplett in der ViewBox abgebildet werden kann. In diesem Fall ist es zusätzlich vonnöten die Ausrichtung mit anzugeben. So wird die Grafik durch Angabe von `preserveAspectRatio="xMidYMin meet"` horizontal mittig und am oberen Rand des Anzeigebereiches platziert, während sie etwa durch „`xMinYMax meet`“ am linken unteren Rand erscheint. Ist hingegen der Wert für `preserveAspectRatio` gleich `slice`, wird die Grafik so proportional skaliert, daß sie den verfügbaren Raum vollständig einnimmt, auch wenn hierdurch Teile über den Viewport hinausragen und damit abgeschnitten werden. Hier läßt sich ebenfalls die Ausrichtung der Abbildung festlegen.

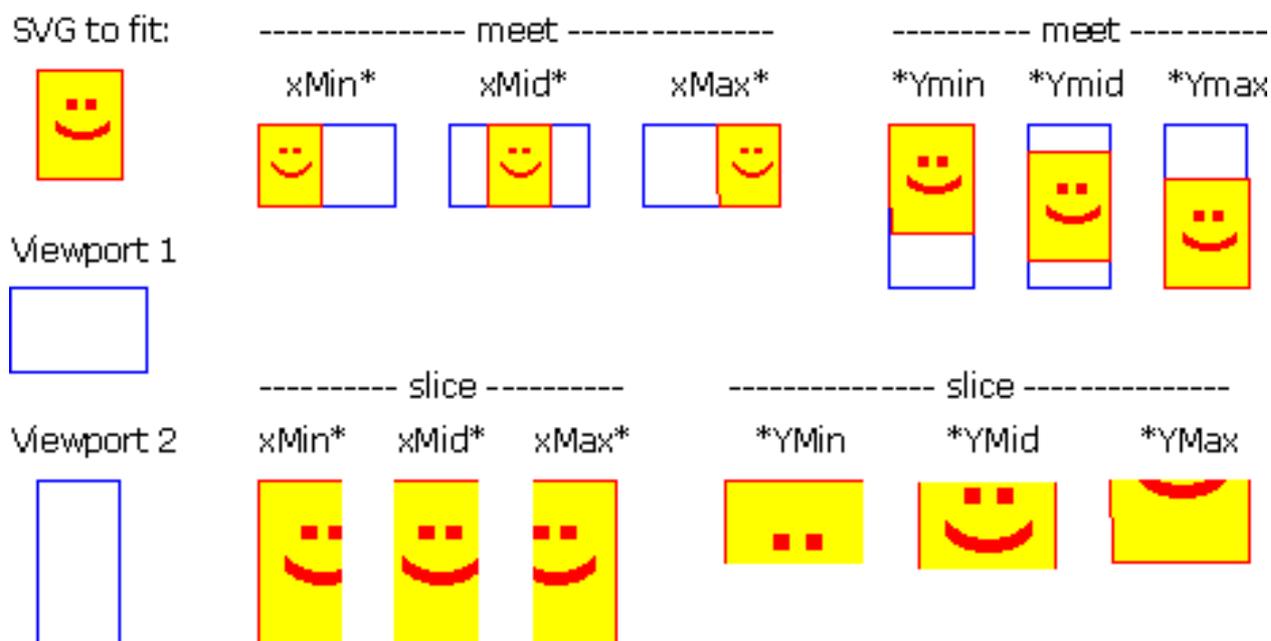


Abbildung 15: Das Attribut `preserveAspectRatio`<sup>82</sup>

### 3.8.6 Rechtecke, Kreise, Ellipsen und Linien

Einfache Formen wie Rechtecke, Kreise, Ellipsen oder Linien können in SVG recht schnell erstellt werden, da für diese Grundformen eigene Tags zur Verfügung stehen.

```
<?xml version="1.0"?>
<svg width="100%" height="100%">
<rect x="10" y="20" width="150" height="75" fill="green" stroke-
width="3" stroke="red"/>
<rect x="125" y="75" rx="30" ry="15" width="100" height="100"
```

<sup>82</sup> Quelle: <http://www.zvon.org/xxl/svgReference/Output/exd0e10952.html>; der Stern (\*) drückt hier aus, daß die konkrete andere Hälfte vor bzw. nach dem angegebenen Wert unerheblich ist. Es muß natürlich trotzdem ein konkreter vollständiger Wert, z.B. `xMinYMax` angegeben werden.

```

style="fill:yellow;stroke-width:5;stroke:orange"/>
<circle cx="100" cy="100" r="75" style="fill:none;stroke-
width:10;stroke:rgb(0,255,255)"/>
<ellipse cx="100" cy="100" rx="70" ry="25" style="fill:magenta;fill-
opacity:0.5;stroke:none"/>
<line x1="15" y1="250" x2="450" y2="150" style="stroke-
width:10;stroke:#ff0000"/>
<line x1="15" y1="265" x2="450" y2="165" style="stroke-
width:10;stroke:#0f0;stroke-dasharray:15 5"/>
<line x1="15" y1="280" x2="450" y2="180" style="stroke-
width:10;stroke:rgb(0%,0%,100%);stroke-dasharray:15 5 20"/>
</svg>

```

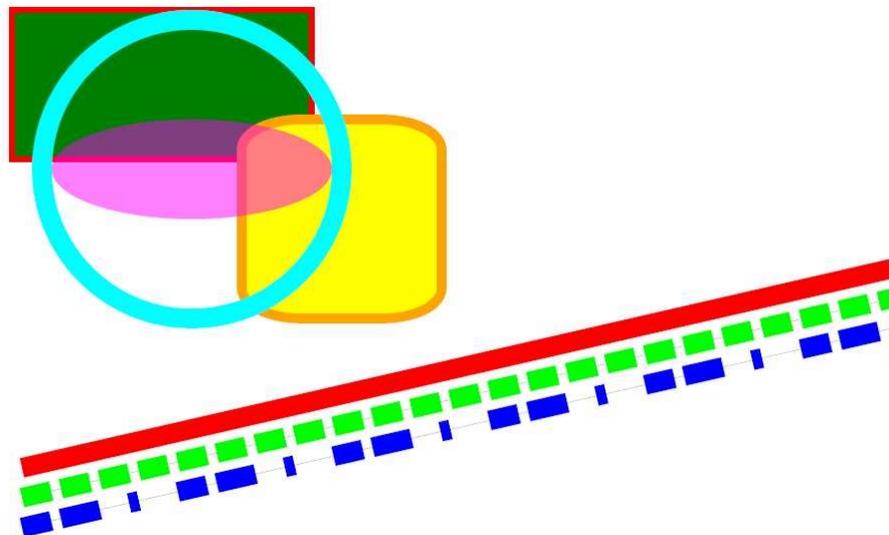


Abbildung 16: Rechteck, Kreis, Ellipse und Linie

Für ein *Rechteck* können die  $x$ - und  $y$ -Koordinate des linken oberen Eckpunktes angegeben werden. Sind diese Angaben nicht vorhanden werden sie standardmäßig mit 0 angesetzt. Dagegen müssen die Breite und die Höhe immer aufgeführt sein. Durch die Werte  $rx$  und  $ry$  lassen sich die Ecken des Rechtecks abrunden. Während beim ersten Rechteck die Füllfarbe, sowie die Strichfarbe und –dicke als Attribute des Elementes angegeben sind, werden diese Formatierungen beim zweiten über Style Sheet-Definitionen zugewiesen.

Ein *Kreis* berücksichtigt neben den Daten für den Mittelpunkt ( $cx$  und  $cy$ ) natürlich auch den Radius. Dieser ist ein Pflichtattribut, wohingegen die Mittelpunktskoordinaten bei Nichtvorhandensein vom System jeweils auf 0 gesetzt werden. Bei diesem Beispiel wurde für die Füllmethode des Kreises der Wert `none` gesetzt, wodurch genau genommen kein Kreis, sondern vielmehr ein Ring gezeichnet wird. Diese Vorgehensweise ist selbstverständlich auch bei den anderen flächenhaften Objekten möglich.

Die *Ellipse* ähnelt logischerweise in ihrer Beschreibung sehr stark derer des Kreises, mit dem Unterschied, daß es hier zwei Radien, entsprechend der  $X$ - und der  $Y$ -Achse gibt. In dem obigen Bild wurde die Umrandung der Ellipse durch die Formatangabe `style="stroke:none"` unterdrückt. Dafür wur-

de die Deckkraft der Füllfarbe (`fill-opacity`) herungesetzt, wodurch die „unter“ der Ellipse liegenden Grafikelemente durchscheinen.

Eine einfache *Linie* läßt sich über die Koordinaten des Start- (`x1` und `y1`) und des Endpunktes (`x2` und `y2`) beschreiben. Fehlt eines dieser Koordinatenpaare, werden die entsprechenden Werte standardmäßig auf 0 gesetzt. Prinzipiell lassen sich sämtliche Striche, egal ob als einfache Linie oder Umrandung einer Fläche mit Hilfe der `stroke-dasharray`-Formatierung als eine gestrichelte bzw. gepunktete Linie gestalten. Dabei geben die aufgeführten Zahlenwerte jeweils die Länge des zu zeichnenden Striches bzw. der Lücke dazwischen an. Bei einer ungeraden Anzahl an Werten folgt nach dem ersten Durchgang an Zahlenwerten quasi eine inverse Darstellung, bevor die eigentliche Darstellung wieder von Neuem beginnt.

Es gibt dreierlei Möglichkeiten den Grafikelementen eine Farbe zuzuordnen. Erstens kann man einen der *benannten Farbwerte* (z.B.: `black`, `white`, `blue`, `red`, `green`, aber auch viele ausgefallenerere<sup>83</sup>) benutzen. SVG bietet hierbei, entsprechend seines Anspruches als das kommende Grafikformat, eine um einiges größere Auswahl als HTML. Alternativ dazu lassen sich Farbwerte entsprechend des RGB-Schemas angeben, indem für den jeweiligen Rot-, Grün- oder Blaukanal entweder ein Wert von 0 bis 255 oder eine Prozentangabe von 0 bis 100% gesetzt wird. Beispiel für Rot:

```
color:rgb(255,0,0) oder
color:rgb(100%,0%,0%)
```

Die dritte Möglichkeit wäre, die RGB-Farbwerte in Hexadezimalschreibweise anzuführen. Rot entspräche auf dieser Art der Angabe:

```
color:#ff0000 oder kurz:
color:#f00
```

Wie man an dem Bild und dem zugehörigen Quelltext erkennen kann, spielt es eine gewichtige Rolle in welcher Reihenfolge die einzelnen Grafikobjekte im SVG-Dokument definiert werden, da vorherige Grafikelemente von nachfolgenden verdeckt werden. Diese Eigenart nennt sich *Painter's Algorithm* und kann selbstverständlich gut zur optischen Gestaltung eingesetzt werden. Wenn dies jedoch nicht berücksichtigt wird, können sich unter Umständen ärgerliche Überraschungen einstellen, vor allem, wenn einzelne Elemente des Bildes erst durch Interaktionen sichtbar gemacht werden sollen.

### 3.8.7 Polylinie und Polygon

Um sich von den grundlegenden Formen lösen zu können und beliebige eigene Figuren durch Linienverbindungen zu kreieren, stehen die Elemente *Polyline* und *Polygon* zur Verfügung.

```
<?xml version="1.0"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  polyline {stroke:brown;stroke-width:3;fill:red}
```

---

<sup>83</sup> Eine Übersicht über die benannten Farbwerte gibt es unter: <http://pilat.free.fr/english/routines/couleurs.php3> oder bei: <http://www.schumacher-netz.de/TR/2001/SVG/types.html#ColorKeywords>

```

    .class1 {stroke:aquamarine;stroke-width:1;fill:yellow}
    .class2 {stroke:magenta;stroke-width:1.5;fill:none}
    #ID1     {stroke:coral;stroke-width:2;fill:grey}
]]>
</style>
<polyline points="10,50 60,100 110,50 110,100 160,150 160,100"/>
<polyline class="class1" points="45,5 35,15 25,15 25,25 15,35 25,45
25,55 35,55 45,65 55,55 65,55"/>
<polyline class="class2" points="75,5 65,15 55,15 55,25 45,35 55,45
55,55 65,55 75,65 85,55 95,55"/>
<polygon id="ID1" points="105,5 95,15 85,15 85,25 75,35 85,45 85,55
95,55 105,65 115,55 125,55"/>
<polygon style="stroke:red" points="175,5 200,50 150,100 150,50 225,75
150,20"/>
</svg>

```

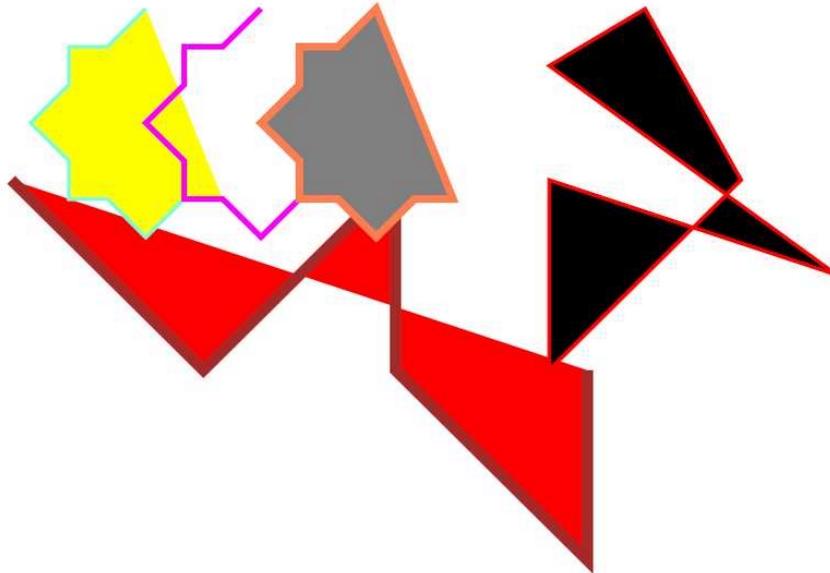


Abbildung 17: Polylinie und Polygon

Der einzige Unterschied zwischen einer Polylinie und einem Polygon besteht darin, daß bei letzterem immer automatisch eine Linie zwischen dem End- und dem Anfangspunkt gezogen und somit der Pfad geschlossen wird. Beide werden durch eine Abfolge an beliebig vielen Koordinatenpaaren beschrieben. Die x- und y-Werte sind durch Kommata zu trennen und die jeweiligen Koordinatenpaare durch ein Leerzeichen. Für die einzelnen Koordinatenwerte können auch Gleitkommazahlen (Angabe mit Dezimalpunkt) angegeben werden, jedoch sind die Angaben immer absolute Koordinatenwerte entsprechend des benutzerdefinierten Koordinatensystems, es sind also keine Einheitenkürzel wie cm, inch oder px zulässig.

### 3.8.8 Verwendung von Cascading Style Sheets

Am Anfang des vorstehenden SVG-Dokumentes, jedoch nach dem SVG-Root-Element, wurde ein CSS-Bereich eingebettet um zentrale Formatvorlagen zu definieren. Dabei sind folgende Dinge zu beachten:

- Bei Verwendung eingebetteter Cascading Style Sheets muß für das `style`-Tag der *MIME*<sup>84</sup>-Type als Attribut in der Form `type="text/css"` angegeben werden.
- Da CSS keine XML-Komponente darstellen, müssen sie vom XML-Parser unbehandelt durchgereicht werden. Dies wird dadurch erreicht, indem man den entsprechenden Bereich durch die Anweisungen `<![CDATA[ und ]]>` eingrenzt. Auf diese Weise gibt der XML-Parser den Inhalt einfach als Zeichendaten weiter und der CSS-Parser kann die Daten adäquat auswerten.

Alternativ dazu läßt sich, analog zu HTML eine externe CSS-Datei anlegen, auf die dann in beliebig vielen SVG-Dokumenten verwiesen werden kann, wie folgender Auszug zeigt.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="formate.css"?>
<svg>
...
</svg>
```

Hier wird ebenfalls die Angabe des MIME-Types benötigt, im Gegensatz zur internen CSS-Definition steht der Verweis auf die externe Datei allerdings schon vor dem Root-Element `<svg>`.

Um die vorgenommenen Formatdefinitionen den SVG-Elementen zuzuweisen, stehen bei den Cascading Style Sheets drei Varianten zur Verfügung:

- *Elementname*: Wird einem Elementnamen ein bestimmtes Format zugewiesen, so erhalten alle Elemente dieses Typs die entsprechende Gestaltung, sofern sie nicht durch eine gesonderte Formatangabe überschrieben wird.

```
polyline {stroke:brown;stroke-width:3;fill:red}
```

Hier werden somit alle Polylinien ohne eigene Formatangaben braun, in der Strichdicke von 3 Pixel gezeichnet und eine rote Füllfläche dazu angelegt, Beispiel:

```
<polyline points="10,50 60,100 110,50 110,100 160,150 160,100"/>
```

- *Klassenname*: Es können beliebig viele Klassen angelegt werden, die dann bei den einzelnen Grafikelementen angegeben werden können, um die dort vorgenommenen Formate zu übernehmen.

```
.class1 {stroke:aquamarine;stroke-width:1;fill:yellow}
```

Dabei läßt sich bei der Klassenfestlegung zusätzlich ein Elementnamen voranstellen, um diese Klasse nur den entsprechenden Elementen zugänglich zu machen.

```
polyline.class1 {stroke:aquamarine;stroke-width:1;fill:yellow}
```

Diese Definitionen werden anschließend über den Klassennamen zugewiesen, Beispiel:

---

<sup>84</sup> Multipurpose Internet Mail Extensions; Ursprünglich nur die Bezeichnung einer Technik, Binärdateien an E-Mails anzuhängen. In der Zwischenzeit handelt es sich um einen Standard, um Dateitypen im Internet voneinander zu unterscheiden. Unterschieden werden dabei zum einen so genannte MIME-Types, d. h. die Arten von im Anhang von E-Mails befindlichen Dokumenten, aber auch die in Webseiten eingebetteten Multimedia-Objekte (in der Regel Objekte, die ein Plug-in erfordern) werden häufig mithilfe dieser Typisierung definiert.

```
<polyline class="class1" points="45,5 35,15 25,15 25,25 15,35 25,45
25,55 35,55 45,65 55,55 65,55"/>
```

- *ID-Wert*: Jedem Element das eine eindeutige ID besitzt, kann ein spezielles Format über diese ID zugeteilt werden.

```
#ID1 {stroke:coral;stroke-width:2;fill:grey}
```

Analog zu den Formatklassen läßt sich hier ebenfalls ein Elementnamen voranstellen.

```
polygone#ID1 {stroke:coral;stroke-width:2;fill:grey}
```

In der Auszeichnung steht dann entsprechend:

```
<polygone id="ID1" points="105,5 95,15 85,15 85,25 75,35 85,45 85,55
95,55 105,65 115,55 125,55"/>
```

### 3.8.9 Beliebige Pfade

Mit den bisher vorgestellten Grundformen lassen sich keine Grafikelemente erstellen, die sowohl aus Geraden- wie auch aus Kurvenelementen bestehen. Mit dem Tag `<path>` und dem damit erstellbaren *Pfad* läßt sich dieses Problem in SVG jedoch lösen. Dies zeigt sich schon allein daran, daß hier der Verlauf nicht in einem Attribut `points` (wie bei der Polylinie und dem Polygon), sondern in dem Attribut `d` (für *description*<sup>85</sup>) angegeben wird. Dabei werden den jeweiligen Koordinatenwerten spezielle Steuerzeichen vorangestellt, um die gewünschte Verlaufsform auszuwählen. Die meisten dieser Befehlszeichen liegen in Groß- und Kleinschreibweise vor, wobei bei Großbuchstaben die folgenden Koordinaten als *absolute* Koordinaten, bezogen auf den Nullpunkt des SVG-Bereiches betrachtet werden, während bei Kleinbuchstaben die Koordinaten *relativ* zum vorhergehenden Punkt angenommen werden.

Tabelle 6: Steuerzeichen und Parameter des path-Elementes

Steuerzeichen und Parameter	Auswirkung
M [X] [Y]	<i>moveto</i> : damit „springt“ man zur angegebenen Stelle der Graphik ohne irgendwelche Zeichenelemente hinzuzufügen. Wird meist am Anfang eines Pfades gesetzt oder um künstliche Lücken zu erschaffen.
m [Δx] [Δy]	
Z	<i>closepath</i> : werden in sich geschlossene Pfade gewünscht, kann mit dieser Anweisung am Ende der Punktfolge eine Verbindung zwischen End- und Startpunkt erzeugt werden, ohne den Startpunkt nochmals explizit angeben zu müssen
z	
L [X] [Y]	<i>lineto</i> : erzeugt eine gerade Linie zu dem Punkt mit den angegebenen Koordinaten
l [Δx] [Δy]	
H [X]	<i>horizontal lineto</i> : erstellt ausgehend vom Ausgangspunkt eine horizontale Linie entsprechend des gegebenen Wertes
h [Δx]	

<sup>85</sup> description (engl.): Beschreibung, Bezeichnung, Darstellung, Schilderung, Bauform

V [Y]	<i>vertical lineto</i> : erstellt ausgehend vom Ausgangspunkt eine vertikale Linie
v [Δy]	entsprechend des gegebenen Wertes
C [X <sub>1</sub> ] [Y <sub>1</sub> ] [X <sub>2</sub> ] [Y <sub>2</sub> ] [X] [Y]	<i>curveto</i> : zeichnet eine kubische Bézierkurve. Dabei werden zwei Anfasserpunkte angegeben, einer am Bogenanfang und einer am –ende. Das dritte Koordinatenpaar bestimmt den Bogenendpunkt.
c [Δx <sub>1</sub> ] [Δy <sub>1</sub> ] [Δx <sub>2</sub> ] [Δy <sub>2</sub> ] [Δx] [Δy]	
S [X <sub>2</sub> ] [Y <sub>2</sub> ] [X] [Y]	<i>smooth curveto</i> : ähnelt <i>curveto</i> , mit dem Unterschied, daß im Falle einer vorangegangenen kubischen Bézierkurve deren zweiter Anfasserpunkt gespiegelt als erster Anfasserpunkt verwendet wird. Ging kein Kurvensegment voraus, wird der erste Anfasserpunkt gleich dem Bogenanfangspunkt gesetzt.
s [Δx <sub>2</sub> ] [Δy <sub>2</sub> ] [Δx] [Δy]	
Q [X <sub>1</sub> ] [Y <sub>1</sub> ] [X] [Y]	<i>quadratic Bézier curveto</i> : erzeugt eine quadratische Bézierkurve mit dem
q [Δx <sub>1</sub> ] [Δy <sub>1</sub> ] [Δx] [Δy]	einen Anfasserpunkt gemäß X <sub>1</sub> ,Y <sub>1</sub> bzw. Δx <sub>1</sub> , Δy <sub>1</sub> zum Bogenendpunkt X,Y bzw. Δx, Δy
T [X] [Y]	<i>smooth quadratic Bézier curveto</i> : ähnelt <i>quadratic Bézier curveto</i> , mit dem
t [Δx] [Δy]	Unterschied, daß im Falle einer vorangegangenen quadratischen Bézierkurve deren Anfasserpunkt gespiegelt als Anfasser verwendet wird. Ging kein Kurvensegment voraus, wird der Anfasserpunkt gleich dem Bogenanfangspunkt gesetzt.
A [RX] [RY] [X-Rotation] [Ellipsenbogen] [Richtung E.bogen] [X] [Y]	<i>elliptical arc</i> : zeichnet eine elliptische Kurve mit den Radien RX und RY. Die Ellipse kann durch die Angabe der X-Achsenrotation gedreht werden. Der Parameter für den Ellipsenbogen kann zwei unterschiedliche Werte annehmen: „0“, falls der kleine und „1“, falls der große Ellipsenbogen ausgewählt werden soll. Mit der Richtung kann bestimmt werden, ob der Bogen im Uhrzeigersinn („1“) oder der gegen den Uhrzeigersinn („0“) gewünscht ist.
a [RX] [RY] [X-Rotation] [Ellipsenbogen] [Richtung E.bogen] [Δx] [Δy]	

Zur Veranschaulichung sollen folgende Beispiele dienen:

- moveto, (horizontal | vertical) lineto & closepath

```
<?xml version="1.0"?>
<svg width="100%" height="100%">
<path stroke="#ff5050" fill="none" d="M50,50 L100,100 L150,50"/>
<path stroke="#50ff50" fill="none" d="m200,50 150,50 150,-50"/>
<path stroke="#5050ff" fill="none" d="M350,50 L400,100 L450,50 Z"/>
<path stroke="#ff50ff" fill="none" d="M50,150 L100,200 L150,150 H200
V200"/>
```

```
<path stroke="#50ffff" fill="none" d="M350,150 L400,200 M450,150 V200
H500"/>
</svg>
```

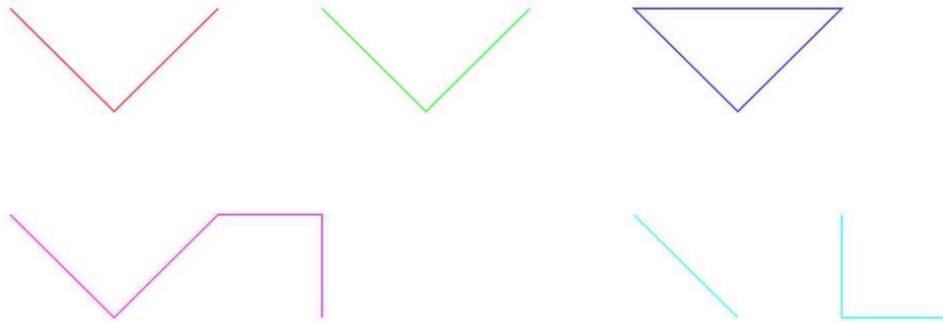


Abbildung 18: moveto, (horizontal | vertical) lineto und closepath

- (smooth) curveto & (smooth) quadratic Bézier curveto

```
<?xml version="1.0"?>
<svg width="100%" height="100%">
<path stroke="#ff5050" fill="none" d="M50,150 L100,100 C100,50 150,50
150,100 L200,50"/>
<path stroke="#50ff50" fill="none" d="M200,150 L250,100 C200,75 350,50
300,100 L350,50"/>
<path stroke="#5050ff" fill="none" d="M350,150 L400,100 C350,75 500,50
450,100 S500,150 500,50"/>
<path stroke="#ff50ff" fill="none" d="M50,300 L100,250 Q150,400
150,250 L200,200"/>
<path stroke="#ddd50" fill="none" d="M200,300 L250,250 Q300,400
300,250 T350,200"/>
</svg>
```

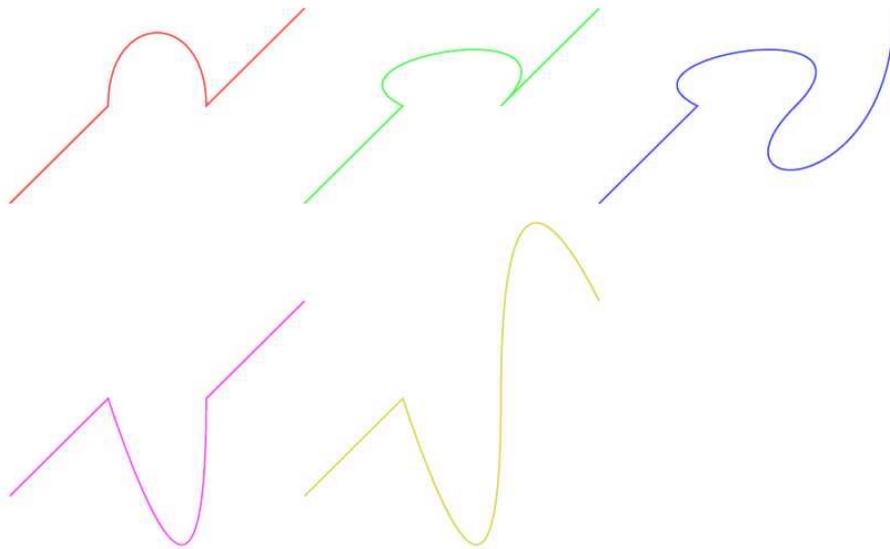


Abbildung 19: (smooth) curveto und (smooth) quadratic Bézier curveto

- elliptical arc

```
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  path {stroke-width:3;stroke:#dddddd;fill:none}
  text {font-size:20}
]]>
</style>
<g>
<path style="stroke:#ff0000" d="M200,150 A150,75 0 0 0 300,200"/>
<path d="M200,150 A150,75 0 1 0 300,200"/>
<path d="M200,150 A150,75 0 0 1 300,200"/>
<path d="M200,150 A150,75 0 1 1 300,200"/>
<text x="65" y="135">Anfangspunkt</text>
<text x="315" y="225">Endpunkt</text>
<text x="20" y="210">Ellipsenbogen=0</text>
<text x="20" y="235">Richtung des Ellipsenbogens=0</text>
</g>
<g transform="translate(500)">
<path d="M200,150 A150,75 0 0 0 300,200"/>
<path style="stroke:#ff0000" d="M200,150 A150,75 0 1 0 300,200"/>
<path d="M200,150 A150,75 0 0 1 300,200"/>
<path d="M200,150 A150,75 0 1 1 300,200"/>
<text x="65" y="135">Anfangspunkt</text>
<text x="315" y="225">Endpunkt</text>
```

```

<text x="20" y="210">Ellipsenbogen=1</text>
<text x="20" y="235">Richtung des Ellipsenbogens=0</text>
</g>
<g transform="translate(0,250)">
<path d="M200,150 A150,75 0 0 0 300,200"/>
<path d="M200,150 A150,75 0 1 0 300,200"/>
<path style="stroke:#ff0000" d="M200,150 A150,75 0 0 1 300,200"/>
<path d="M200,150 A150,75 0 1 1 300,200"/>
<text x="65" y="135">Anfangspunkt</text>
<text x="315" y="225">Endpunkt</text>
<text x="20" y="210">Ellipsenbogen=0</text>
<text x="20" y="235">Richtung des Ellipsenbogens=1</text>
</g>
<g transform="translate(500,250)">
<path d="M200,150 A150,75 0 0 0 300,200"/>
<path d="M200,150 A150,75 0 1 0 300,200"/>
<path d="M200,150 A150,75 0 0 1 300,200"/>
<path style="stroke:#ff0000" d="M200,150 A150,75 0 1 1 300,200"/>
<text x="65" y="135">Anfangspunkt</text>
<text x="315" y="225">Endpunkt</text>
<text x="20" y="210">Ellipsenbogen=1</text>
<text x="20" y="235">Richtung des Ellipsenbogens=1</text>
</g>
</svg>

```

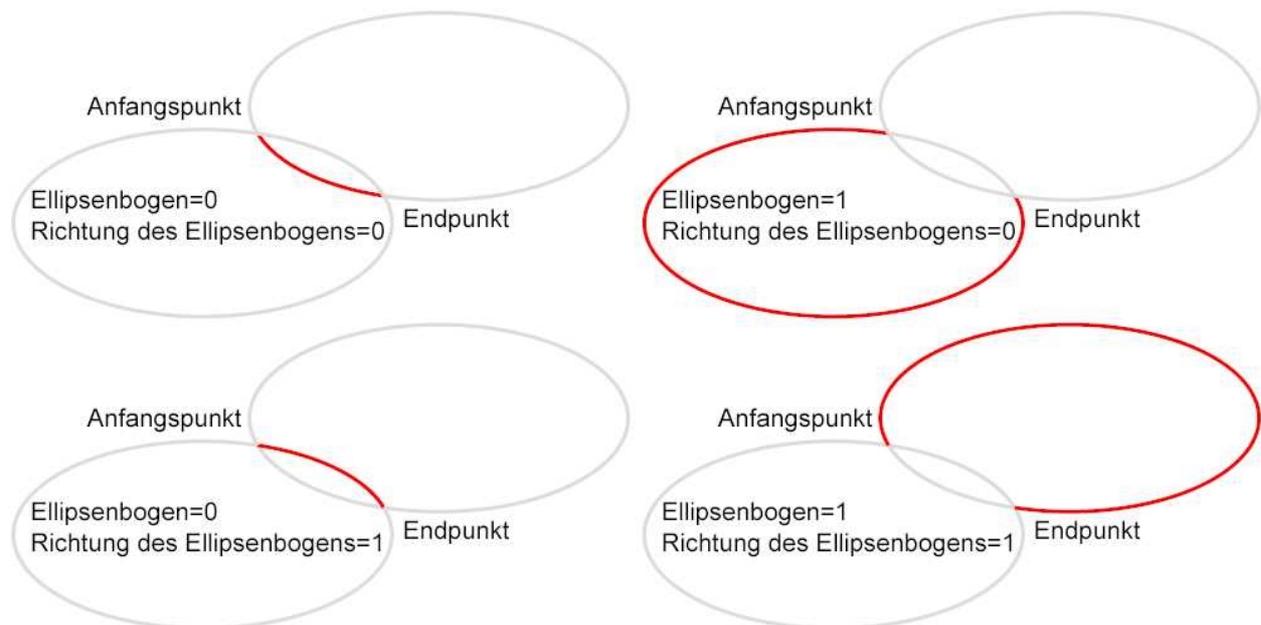


Abbildung 20: Elliptical Arc

### 3.8.10 Texte

Anhand Abbildung 20: Elliptical Arc und dem dazugehörigen Quelltext, läßt sich auch erkennen, wie einfach *Texte* in SVG plaziert werden können. Nur ist bei Texten darauf zu achten, sie mittels der x- und y-Attribute an der richtigen Stelle zu positionieren. Diese sind laut DTD zwar keine Pflichtattribute, wenn sie jedoch nicht explizit aufgeführt sind, nimmt das System dafür jeweils den Wert 0 an, und der Text wird somit normalerweise außerhalb des Anzeigebereiches gesetzt. Bei der momentanen Version von SVG gibt es allerdings den kleinen Nachteil, daß sich Texte nicht so bequem über mehrere Zeilen verteilen lassen, wie es beispielsweise bei HTML über das <br>-Tag möglich ist. Für zukünftige Versionen ist jedoch angedacht, Texte innerhalb eines oder sogar mehrerer Rechtecke bzw. Formen verlaufen und ausrichten zu lassen.

Bis es so weit ist, kann dem Problem dadurch abgeholfen werden, indem man etwa mit dem <tspan>-Tag einzelne *Textabschnitte* bildet und diese relativ zum vorhergehenden Text-Element ausrichtet. Textabschnitte werden aber auch eingesetzt, um einzelne Textbereiche besonders zu formatieren. Insgesamt stellt SVG eine Vielzahl an Formatvorgaben von Texten zur Verfügung. Der folgende Quelltext und die dazugehörige Grafik sollen einen kurzen Einblick dazu liefern.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  path {stroke-width:3;stroke:#dddddd;fill:none}
  text {font-size:20}
]]>
</style>
<text x="50" y="100">Dieses ist die erste Zeile
<tspan x="50" dy="25">Dieses ist die zweite Zeile</tspan>
<tspan x="50" dy="25">Dieses ist die dritte Zeile</tspan>
</text>
<text x="50" y="175">Das ist normale Schrift, es geht aber auch
<tspan style="font-weight:bold">dicker</tspan>.
</text>
<text x="50" y="225">Selbstverständlich sind auch unterschiedliche
<tspan style="font-size:40">Schriftgrößen</tspan>
kein Problem.
</text>
<text x="50" y="250">Selbst bei der
<tspan style="stroke:blue;fill:red">Farbgebung</tspan>
gibt es viele Möglichkeiten
</text>
<path id="textpfad" d="M50,350 C150,300 350,250 700,350 "/>
<text>
<textPath xlink:href="#textpfad" startOffset="40">
<tspan dy="-5">Man kann einen Text sogar entlang eines Pfades
```

```

erlaufen lassen</tspan>
</textPath>
</text>
<text x="50" y="375">Man kann einzelne
<tspan kerning="5"
otate="10,20,30,40,50,60,70,80,90,100">Buchstaben</tspan>
oder ganze Textbereiche
</text>
<g transform="translate(620,350),rotate(45)">
<text>drehen</text>
</g>
</svg>

```

Dieses ist die erste Zeile

Dieses ist die zweite Zeile

Dieses ist die dritte Zeile

Das ist normale Schrift, es geht aber auch **dicker**.

Selbstverständlich sind auch unterschiedliche **Schriftgrößen** kein Problem.  
Selbst bei der **Farbgebung** gibt es viele Möglichkeiten

Man kann einen Text sogar entlang eines Pfades verlaufen lassen

Man kann einzelne *B u c h s t a b e n* oder ganze Textbereiche *drehen*

Abbildung 21: Darstellung von Texten

### 3.8.11 Gruppierungen und Transformationen

Bei den letzten beiden SVG-Dokumenten trat das bislang unbehandelte `<g>`-Tag auf. Dieses kann verwendet werden, um mehrere Einzelelemente zu einer *Gruppe* zusammenzufassen, was einerseits einer besseren Strukturierung nützt, andererseits aber auch die Voraussetzungen schafft, um die enthaltenen Elemente einheitlich transformieren, also skalieren, verschieben, rotieren usw. zu können.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
rect {stroke-width:3}
text {font-size:20}
]]>
</style>
<g>
<rect style="stroke:#ff0000;fill:#00ffff" x="0" y="0" width="150"

```

```
height="75"/>
<text x="12" y="32">Original</text>
</g>
<g transform="translate(125,60)">
<rect style="stroke:#00ff00;fill:#ff00ff" x="0" y="0" width="150"
height="75"/>
<text x="12" y="32">translate(X,Y)</text>
</g>
<g transform="rotate(45,325,155),translate(250,120)">
<rect style="stroke:#0000ff;fill:#ffff00" x="0" y="0" width="150"
height="75"/>
<text x="12" y="32">rotate(D,X,Y)
<tspan x="6" dy="20">&amp; translate(X,Y)</tspan>
</text>
</g>
<g transform="translate(0,200),scale(2,0.75)">
<rect style="stroke:#ff00ff;fill:#00ff00" x="0" y="0" width="150"
height="75"/>
<text x="12" y="32">translate(X,Y)
<tspan x="12" dy="20">&amp; scale(X,Y)</tspan>
</text>
</g>
<g transform="translate(125,250),skewX(30)">
<rect style="stroke:#ffff00;fill:#8080ff" x="0" y="0" width="150"
height="75"/>
<text x="12" y="32">translate(X,Y)
<tspan x="12" dy="20">&amp; skewX(X)</tspan>
</text>
</g>
<g transform="translate(300,250),skewY(30)">
<rect style="stroke:#00ffff;fill:#ff0000" x="0" y="0" width="150"
height="75"/>
<text x="12" y="32">translate(X,Y)
<tspan x="12" dy="20">&amp; skewY(Y)</tspan>
</text>
</g>
</svg>
```

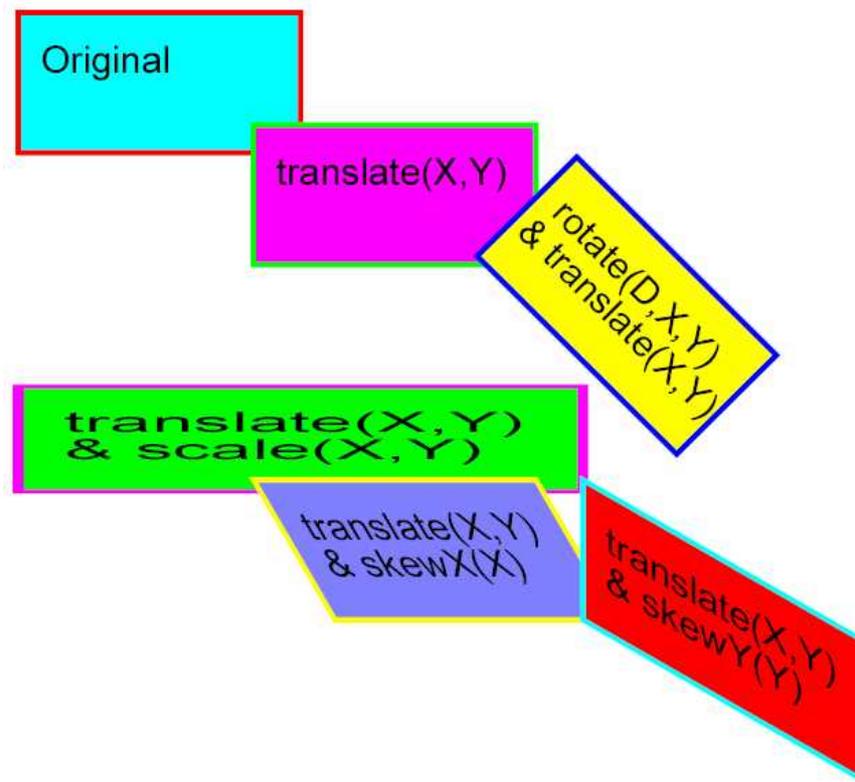


Abbildung 22: Transformationen mit translate, rotate, scale und skew

Für die *Translation* können ein oder zwei Parameter angegeben werden. Bei einem Parameter wird das Element um den entsprechenden Wert in X-Richtung verschoben, der zweite Wert wirkt sich entsprechend auf die vertikale Position aus. Bei der *Rotation* sind bis zu drei Angaben möglich. Die erste setzt den Rotationswinkel in Grad, die optionalen restlichen beiden die Koordinaten für den Rotationsmittelpunkt. Sind dafür keine Werte gegeben, wird um den Koordinatenursprung gedreht. Wird bei der *Skalierung* nur ein Parameter angegeben, wird die Grafik proportional vergrößert bzw. verkleinert, bei zwei Parametern wird für jede Koordinatenachse ein eigener Skalierungsfaktor angebracht. Mit den Anweisungen `skewX` und `skewY` kann eine *Neigung* der Grafikelemente um den übergebenen Winkel (ebenfalls in Grad) auf der jeweiligen Achse erzeugt werden.

Wie man sieht, lassen sich problemlos mehrere Transformationen miteinander kombinieren, dabei gilt es jedoch zu beachten, daß die Reihenfolge der einzelnen Schritte von Bedeutung ist. So ergibt beispielsweise eine Translation mit anschließender Rotation, trotz gleicher Parameter, ein anderes Ergebnis als die Rotation mit darauf folgender Translation.

### 3.8.12 Symbole

Soll eine Grafik innerhalb eines SVG-Dokumentes mehrmals verwendet werden, empfiehlt es sich diese als *Symbol* zu definieren, um es später mit dem Tag `<use>` referenzieren zu können. Vor allem im Bereich der Kartographie bietet sich dieses Verfahren an, um kartographische Symbole an beliebig vielen Stellen zu plazieren.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  circle {stroke-width:3;stroke:black;fill:#aaaaff}
  polygon {stroke-width:2;stroke:black;fill:#dddfff}
  text {font-size:15;text-anchor:middle}
]]>
</style>
<symbol id="BGRUNDviewer" viewBox="-5 -5 105 105"
preserveAspectRatio="none">
<circle cx="50" cy="50" r="40"/>
<polygon points="0,75 50,0 100,75"/>
<text x="50" y="35">www.
<tspan x="50" dy="12">BGRUND</tspan>
<tspan x="50" dy="12">viewer</tspan>
<tspan x="50" dy="12">.de</tspan>
</text>
</symbol>
<use xlink:href="#BGRUNDviewer" width="100" height="100"/>
<use xlink:href="#BGRUNDviewer" x="75" y="75" width="100"
height="100"/>
<use xlink:href="#BGRUNDviewer" x="150" y="150" width="200"
height="200"/>
<use xlink:href="#BGRUNDviewer" x="350" y="0" width="200"
height="100"/>
<use xlink:href="#BGRUNDviewer" x="350" y="100" width="100"
height="200"/>
<use xlink:href="#BGRUNDviewer" width="100" height="100"
transform="translate(300,300),rotate(45,300,300)"/>
</svg>
```

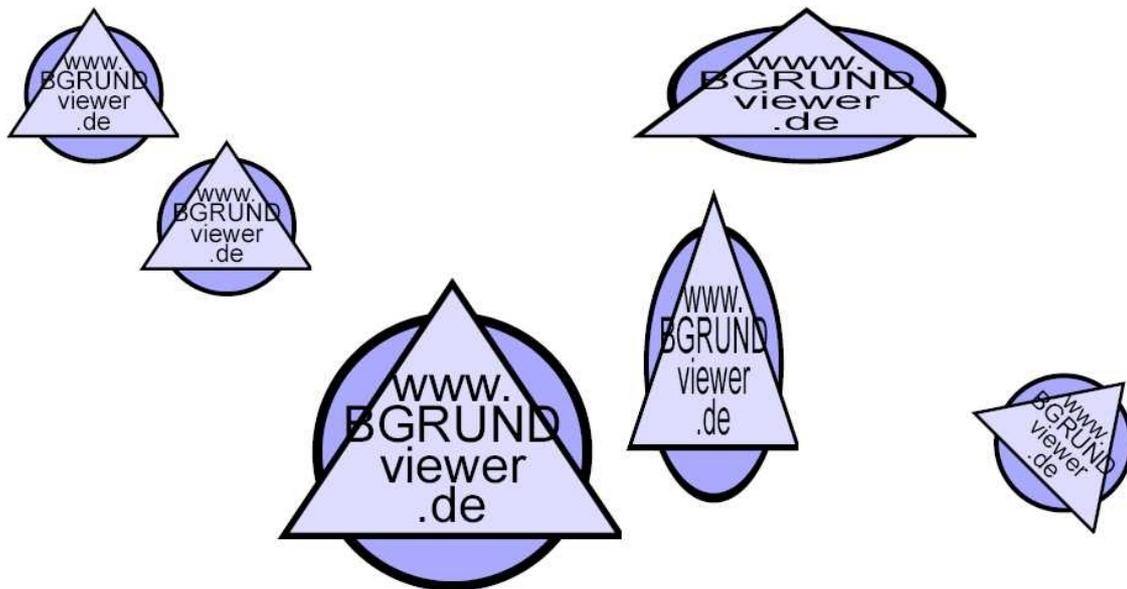


Abbildung 23: Verwendung von Symbolen

Die innerhalb des Tags `<symbol>` aufgelisteten Grafikelemente werden erst dann gerendert<sup>86</sup>, wenn dieses Symbol, referenziert über das Attribut `id`, an beliebig vielen Stellen im SVG-Dokument aufgerufen wird. Beim `symbol`-Element lassen sich, wie beim `<svg>`-Tag, die Attribute `viewBox` und `preserveAspectRatio` definieren. Dadurch und durch die Angabe der Höhe und Breite des aufrufenden Tags `<use>` lassen sich Symbole an Ort und Stelle nach Belieben skalieren, zusätzlich können alle Arten von Transformationen angewandt werden.

### 3.8.13 Einbinden externer Grafiken

Mit Referenzen können in SVG auch *externe Grafiken* eingebunden werden, seien es andere SVG-Dokumente oder Rasterbilder in den Formaten GIF, JPEG oder PNG.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  text {font-size:25;text-anchor:middle}
  path {stroke:#aaaaff;stroke-width:10;fill:none}
]]>
</style>
<image xlink:href="karte.jpg" x="50" y="50" width="750" height="70"/>
<text x="425" y="75">JPEG-Grafik</text>
<path d="M75,25 10,120 1750,0"/>
<path d="M100,25 a100 100 0 0 0 100,100 1625,0"/>
</svg>
```

<sup>86</sup> to render (engl.): erbringen, leisten, machen, wiedergeben

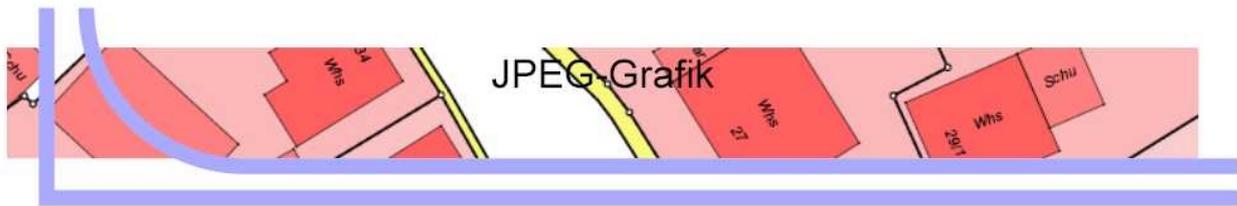


Abbildung 24: Einbinden von externen Grafiken

### 3.8.14 Grafische Effekte

Auch hinsichtlich *grafischer Effekte* bietet der SVG-Standard eine große Palette. So sind unter anderem unterschiedliche Farbverläufe, eigens definierte Füllmuster und Beschneidungspfade, Alphamasken und eindrucksvolle Filtereffekte produzierbar.

#### 3.8.14.1 Verläufe

Folgender Quelltext erzeugt drei Kreise. Der erste wird gewöhnlich mit einer vorgegebenen Farbe gefüllt. Beim zweiten wird ein *linearer Verlauf* verwendet, während beim dritten ein *radialer Verlauf* zum Einsatz kommt.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  circle {stroke:black;stroke-width:2}
  text {font-size:20;text-anchor:middle}
]]>
</style>
<defs>
<linearGradient id="linGrad" gradientUnits="userSpaceOnUse" x1="0"
y1="0" x2="100" y2="100">
<stop offset="10%" style="stop-color:#ffffff"/>
<stop offset="90%" style="stop-color:#aaaaff"/>
</linearGradient>
<radialGradient id="radGrad" gradientUnits="userSpaceOnUse" cx="50"
cy="50" r="50" fx="75" fy="25">
<stop offset="10%" style="stop-color:#ffffff"/>
<stop offset="90%" style="stop-color:#aaaaff"/>
</radialGradient>
</defs>
<g transform="translate(50,50)">
<circle cx="50" cy="50" r="50" style="fill:#aaaaff"/>
<text x="50" y="120">Normal</text>
</g>
<g transform="translate(250,50)">
```

```

<circle cx="50" cy="50" r="50" style="fill:url(#linGrad)"/>
<text x="50" y="120">Linearer Verlauf</text>
</g>
<g transform="translate(450,50)">
<circle cx="50" cy="50" r="50" style="fill:url(#radGrad)"/>
<text x="50" y="120">Radialer Verlauf</text>
</g>
</svg>

```

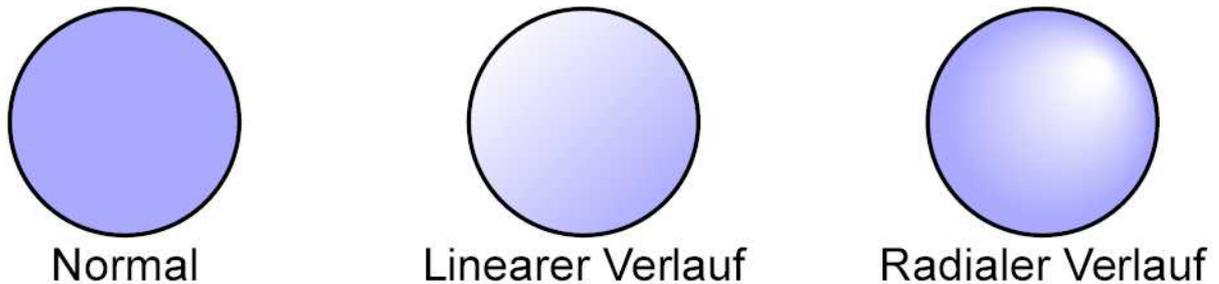


Abbildung 25: Linearer und radialer Verlauf

Verläufe, wie auch die anderen grafischen Effekte, werden am Anfang des SVG-Dokumentes innerhalb eines `<defs>`-Bereiches deklariert. Für beide Verlaufsformen steht die Auswahl zweier Koordinatensysteme bereit. `userSpaceOnUse` bezeichnet hierbei das benutzerdefinierte Koordinatensystem der Grafik, der der Effekt zugewiesen wird, wohingegen `objectBoundingBox` für ein Rechteck steht, in das die Grafik, der der Effekt zugewiesen wird, genau hineinpaßt. Beim linearen Verlauf wird über die Werte `x1,y1,x2` und `y2` eine Linie definiert, die die Richtung und die Länge des Verlaufsmusters vorgibt. Beim radialen Verlauf übernehmen diese Funktion ein Kreis mit den Angaben `cx,cy` und `r` sowie ein separater Startpunkt für den Verlauf (`fx` und `fy`). Die Anfangs- und Endzustände des Verlaufes werden bei beiden Arten über das `<stop>`-Element festgelegt. Das Attribut `offset` steuert die Grenzen der einzelnen Verlaufsabschnitte, und mit den Style-Vorgaben `stop-color` und `stop-opacity` kann die jeweilige Farbe bzw. Deckkraft angegeben werden.

### 3.8.14.2 Füllmuster

Will man spezielle Flächen mit einem *Füllmuster* versehen, können dafür mit Hilfe des Tags `<pattern>` Musterkacheln angelegt werden, die später für die Füllung herangezogen werden.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  line {stroke:red;stroke-width:3}
  text {font-size:20;text-anchor:middle}
]]>
</style>
<defs>

```

```
<pattern id="schraffur" patternUnits="userSpaceOnUse" x="0" y="0"
width="10" height="100" viewBox="0 0 10 100">
<line x1="5" y1="0" x2="5" y2="100"/>
</pattern>
<pattern id="patText" patternUnits="userSpaceOnUse" x="0" y="0"
width="50" height="20" viewBox="0 0 50 20">
<text x="25" y="20">Hallo</text>
</pattern>
</defs>
<g transform="translate(50,50)">
<circle cx="50" cy="50" r="50" style="fill:url(#schraffur)"/>
<text x="50" y="120">kreisförmige Schraffur</text>
</g>
<g transform="translate(250,50)">
<rect x="0" y="0" width="150" height="100"
style="fill:url(#schraffur)"/>
<text x="75" y="120">rechteckige Schraffur</text>
</g>
<g transform="translate(500,50)">
<polygon style="fill:url(#schraffur)" points="50,0 100,100 0,100"/>
<text x="50" y="120">Schraffur in Polygon</text>
</g>
<g transform="translate(50,250)">
<circle cx="50" cy="50" r="50" style="fill:url(#patText)"/>
<text x="50" y="120">kreisförmige Schraffur</text>
</g>
<g transform="translate(250,250)">
<rect x="0" y="0" width="150" height="100"
style="fill:url(#patText)"/>
<text x="75" y="120">rechteckige Schraffur</text>
</g>
<g transform="translate(500,250)">
<polygon style="fill:url(#patText)" points="50,0 100,100 0,100"/>
<text x="50" y="120">Schraffur in Polygon</text>
</g>
</svg>
```

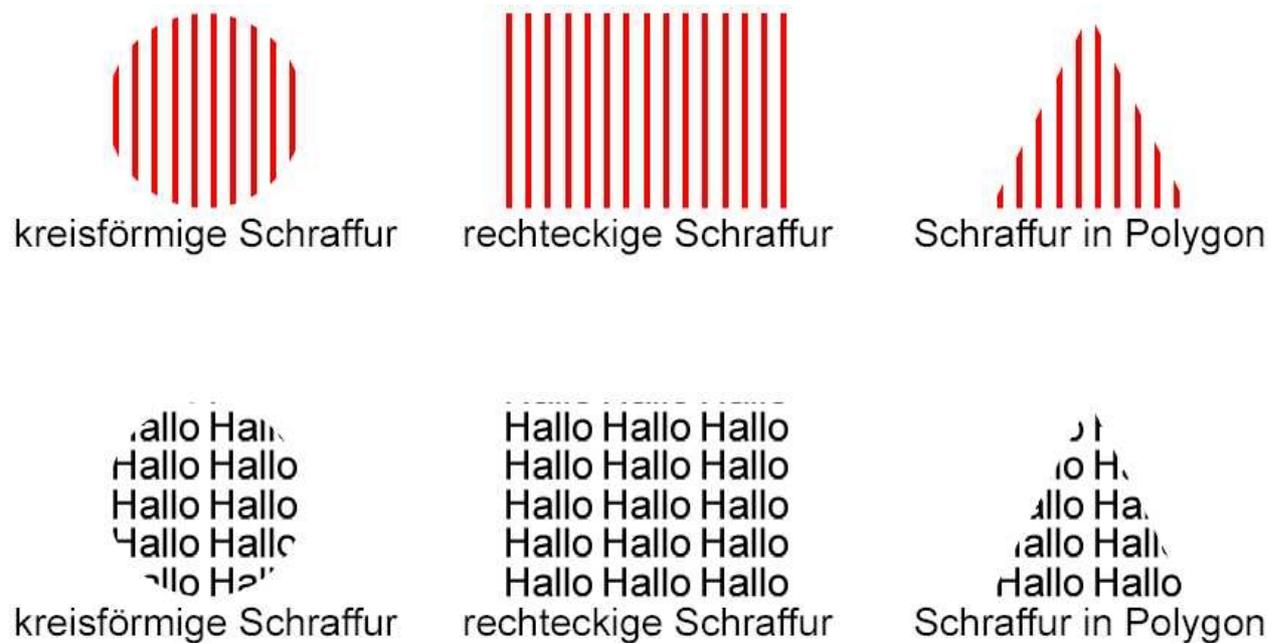


Abbildung 26: Füllmuster

Die `patternUnits` entsprechen den `gradientUnits` der Verläufe und mit `x` und `y` werden die Koordinaten für den Startpunkt der ersten Musterkachel im Füllmuster festgelegt. Zusätzlich wäre es machbar, das Koordinatensystem des Musters über das Attribut `patternTransform`, entsprechend den allgemeinen Transformationsvorgaben umzuformen. Leider gelten die damit angegebenen Werte immer für das gesamte SVG-Dokument und können nicht für jedes Objekt, dem dieses Füllmuster zugewiesen wird, angepaßt werden. Typisches Beispiel dafür wäre eine Gebäudeschraffur, bei der der Winkel der Schraffurlinien von der Richtung der Gebäudekontur abhängig ist (parallel bzw. senkrecht oder im 45°-Winkel zu einer bestimmten Gebäudeseite).

### 3.8.14.3 Masken

Mit *Beschneidungsmasken* bzw. -pfaden lassen sich Grafiken oder Bilder in einer bestimmten Form beschneiden. Diese Form richtet sich nach der Kontur eines beliebigen Grafikelements, das innerhalb des Tags `<clipPath>` aufgeführt wird.

```
<?xml version="1.0"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
text {font-size:90;font-weight:bold;fill:none;stroke:black;stroke-
width:1;font-family:Impact}
]]>
</style>
<clipPath id="BGRUNDviewer" clipPathUnits="userSpaceOnUse">
<text x="0" y="80">BGRUNDviewer</text>
</clipPath>
<image xlink:href="karte.jpg" width="700" height="85" style="clip-
```

```
path:url(#BGRUNDviewer)"/>
<text x="0" y="80">BGRUNDviewer</text>
</svg>
```



Abbildung 27: Beschneidungsmasken

#### 3.8.14.4 Filter

Tags die *Filtereffekte* definieren, beginnen grundsätzlich mit `fe`. So gibt es in SVG etwa einen Gauß'schen Weichzeichnungsfiler (`<feGaussianBlur>`) und spezielle Beleuchtungseffekte (`<feDiffuseLighting>` und `<feSpecularLighting>`), die die Grafiken unter Annahme einer bestimmten Lichtquelle plastisch hervortreten lassen. Ferner lassen sich zwei Bilder über die Tags `<feComposite>` und `<feBlend>` miteinander verrechnen, Farbwerte gezielt verändern (`<feColorMatrix>`) oder Alphamasken (`<mask>`) anlegen. Im Rahmen dieser Diplomarbeit wurden diese Effekte jedoch nur sehr begrenzt eingesetzt, weshalb hier neben der nachfolgenden Kostprobe nur auf das Buch „SVG – Scalable Vector Graphics“<sup>87</sup> von Iris Fibinger verwiesen wird.

```
<?xml version="1.0"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
text {font-size:90;font-weight:bold;fill:none;stroke:black;stroke-
width:5;font-family:Impact}
]]>
</style>
<filter id="IDdiffuse" x="0" y="0" width="700" height="85"
filterUnits="userSpaceOnUse">
<feGaussianBlur in="SourceAlpha" stdDeviation="3" result="schatten"/>
<feDiffuseLighting in="schatten" surfaceScale="10" diffuseConstant="1"
lighting-color="white" result="DiffLight">
<fePointLight x="-5000" y="-5000" z="10000"/>
</feDiffuseLighting>
<feComposite in="DiffLight" in2="SourceAlpha" operator="in"
result="DiffLight"/>
<feComposite in="SourceGraphic" in2="DiffLight" operator="arithmetic"
k1="0" k2=".5" k3=".5" k4="0"/>
</filter>
```

<sup>87</sup> Fibinger, Iris (2002): SVG – Scalable Vector Graphics, Praxiswegweiser und Referenz für den neuen Vektorgrafikstandard. Markt + Technik Verlag, ISBN 3-8272-6239-9

```
<text x="0" y="80" style="filter:url(#IDdiffuse)">BGRUNDviewer</text>
</svg>
```

# BGRUNDviewer

Abbildung 28: Gaußscher Weichzeichnungsfilter und Beleuchtungseffekte

### 3.8.15 Animationen und Interaktivität

Um SVG-Grafiken dynamisch zu gestalten ergeben sich zwei Möglichkeiten. Entweder benutzt man dafür SVG-eigene Tags und Attribute, die dem XML-Multimediastandard SMIL<sup>88</sup> entstammen, oder die Attribute der Grafikelemente werden verändert, indem man mittels einer Skriptsprache, wie etwa JavaScript, und dem Document Object Model direkt auf sie zugreift und deren Werte neu setzt. Hierzu stelle man sich einen Text vor, der beim Überfahren mit dem Mauszeiger ein bisher unsichtbares Rechteck sichtbar macht. Sobald der Mauszeiger den Textbereich verläßt soll dieses Rechteck wieder verschwinden.

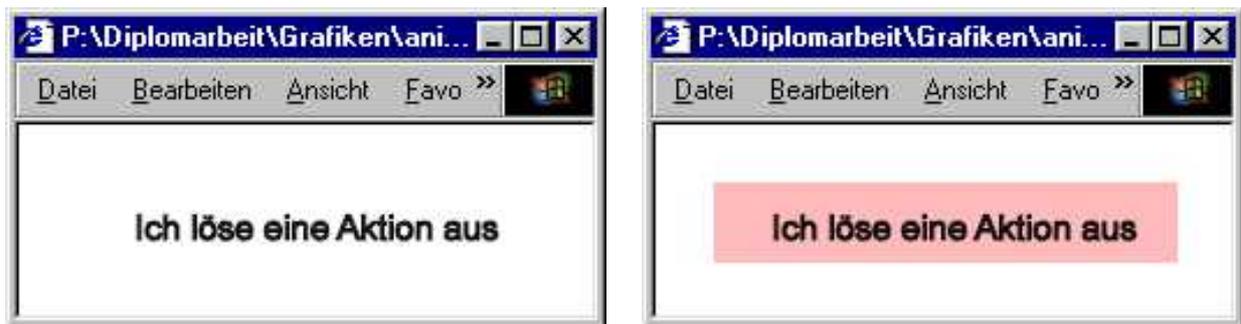


Abbildung 29: interaktives SVG

Unter Verwendung der aus dem Modul SMIL-Animation herrührenden Tags kann dies mit folgendem Quelltext realisiert werden:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg width="100%" height="100%">
<style type="text/css">
<![CDATA[
  rect {stroke:none;fill:#ffbbbb;visibility:hidden}
  text {font-size:15;stroke:black;font-family:Arial}
]]>
</style>
<rect id="rechteck" x="25" y="25" width="200" height="35"/>
```

<sup>88</sup> Synchronized Multimedia Integration Language, Spezifikation unter: <http://www.w3.org/TR/REC-smil/>

```
<text x="50" y="50">Ich löse eine Aktion aus</text>
<set xlink:href="#rechteck" attributeName="visibility" to="visible"
begin="mouseover" end="mouseout"/>
</svg>
```

Hier reicht also ein einziges Tag, um die gewünschte Interaktion ablaufen zu lassen. Bei stufenlosen Veränderungen findet dabei das Element `<set>` Gebrauch, bei interpolierbaren Veränderungen, wie etwa Farbwechseln, Deckkraft- oder Positionsänderungen, ist dagegen das Tag `<animate>` vielseitiger einsetzbar, da hier noch Zeitangaben bezüglich Dauer oder ähnliches angegeben werden können. Das Bezugselement dafür ist stets das vorhergehende, in diesem Fall also das `<text>`-Objekt. Über die Angabe `xlink:href="#rechteck"` wird das betroffene Grafikelement referenziert, `attributeName` definiert die gewünschte Eigenschaft und `to` den verlangten Endwert. Die Attribute `begin` und `end` legen fest, wann die Veränderung gestartet bzw. beendet werden soll. Die Kriterien können entweder Zeitwerte oder bestimmte Ereignisse sein.

Um das gleiche Ergebnis durch ein Skript und den Zugriff über das DOM zu erreichen, kann das SVG-Dokument alternativ in dieser Weise gestaltet werden:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg id="svgDoc" width="100%" height="100%">
<script type="text/javascript">
<![CDATA[
function zeige(evt,elementID) {
    var element =
document.getElementById("svgDoc").getElementById(elementID);
    if(evt.type == "mouseover")
        element.setAttribute("visibility","visible");
    else
        element.setAttribute("visibility","hidden");
}
]]>
</script>
<style type="text/css">
<![CDATA[
    rect {stroke:none;fill:#ffb3bb}
    text {font-size:15;stroke:black;font-family:Arial}
]]>
</style>
<rect id="rechteck" x="25" y="25" width="200" height="35"
visibility="hidden"/>
<text x="50" y="50" onmouseover="zeige(evt,'rechteck')"
onmouseout="zeige(evt,'rechteck')">Ich löse eine Aktion aus</text>
</svg>
```

Wie unschwer zu erkennen ist, liegt bei dieser Variante der Aufwand etwas höher als bei der Verwendung von Animations-Tags, wodurch diese Lösung nur dann interessant erscheint, sobald gleich mehrere Elemente oder Attribute verändert werden sollen. Zu beachten ist jedoch, daß bei SVG, im Gegensatz zu HTML, kein unmittelbarer Zugriff auf die einzelnen Elemente möglich ist, sondern der Weg über das SVG-Dokument eingeschlagen werden muß. Dazu wird das Rechteck in diesem Fall über `document.getElementById("svgDoc").getElementById(elementID)` angesprochen und dann über die Methode `setAttribute` modifiziert. Auslesen könnte man den Wert mit der Methode `getAttribute`, eine ganz wichtige Stolperstelle gibt es hierbei allerdings: wurde die Eigenschaft des Grafikelementes durch CSS angegeben, muß dementsprechend auch die Anweisung angepaßt werden.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<svg id="svgDoc" width="100%" height="100%">
<script type="text/javascript">
<![CDATA[
function zeige(evt,elementID) {
    var element =
document.getElementById("svgDoc").getElementById(elementID);
    if(evt.type == "mouseover")
        element.style.setProperty("visibility","visible");
    else
        element.style.setProperty("visibility","hidden");
}
]]>
</script>
<style type="text/css">
<![CDATA[
    rect {stroke:none;fill:#ffb3bb}
    text {font-size:15;stroke:black;font-family:Arial}
]]>
</style>
<rect id="rechteck" x="25" y="25" width="200" height="35"
style="visibility:hidden"/>
<text x="50" y="50" onmouseover="zeige(evt,'rechteck')"
onmouseout="zeige(evt,'rechteck')">Ich löse eine Aktion aus</text>
</svg>
```

Auch das Bestimmen einer CSS-Angabe müßte in diesem Falle in `style.getPropertyValue` (man beachte das zusätzliche „Value“) abgeändert werden.

Während der Arbeit an dieser Diplomarbeit fiel außerdem auf, daß das Verändern von Attributen über JavaScript und das Document Object Model nicht hundertprozentig zuverlässig verläuft. Das heißt im konkreten Fall, daß die Sichtbarkeit von Grafikelementengruppen (festgehalten durch das `<g>`-Tag) nicht absolut zuverlässig durch Event-Handler (onmouseover, onmouseout) gesteuert werden konnte, weshalb auf die Animations-Tags ausgewichen wurde.

## 4 Die Daten der Vermessungsverwaltung

### 4.1 Das Liegenschaftskataster

Das *Liegenschaftskataster* ist der einzige flächendeckende und aktuell gehaltene Nachweis aller Flurstücke und Gebäude in Baden-Württemberg. Es wird von den staatlichen Vermessungsämtern<sup>89</sup> und verschiedenen städtischen Vermessungsdienststellen geführt. Wesentliche Bestandteile des Liegenschaftskatasters sind das *Liegenschaftsbuch* als beschreibender Nachweis und die *Liegenschaftskarte* als geometrischer Nachweis der Liegenschaften.

Seit 1994 wird das Liegenschaftsbuch in digitaler Form als *Automatisiertes Liegenschaftsbuch (ALB)* geführt. Die Liegenschaftskarte wird derzeit digital als *Automatisierte Liegenschaftskarte (ALK)* in Form der Punkt- und Grundrissdatei *BGRUND* eingerichtet und soll künftig die Katasterkarten ersetzen. Die Abgabe von Auszügen aus ALB und ALK<sup>90</sup> setzt ein berechtigtes Interesse voraus. Die Datenbereitstellung erfolgt über die Angabe der Flurstücksnummer oder über einen beliebig definierten rechteckigen bzw. polygonförmigen Rahmen, oder gemarkungsweise.

Nach dem Nachweis eines berechtigten Interesses können bei den einzelnen Vertretungen der Vermessungsverwaltung die Daten vor Ort erhoben werden, entweder auf Papierform oder auf digitalen Datenträgern. Man kann aber auch auf der Internetseite<sup>91</sup> des Landesvermessungsamtes über das Informationssystem *GEODIS*<sup>92</sup> digitale Daten der Landesvermessung und des Liegenschaftskatasters, wie etwa auch Topografische Kartenwerke, Landschafts- und Höhenmodelle, Luftbilder und Orthophots und Festpunkt-daten beantragen. Je nach Datenmenge wird die Anfrage sofort oder über Nacht verarbeitet und kann je nach Wunsch als Download angeboten, per E-Mail oder auf Diskette bzw. CD-ROM verschickt werden.

Spätestens ab 2005 sollen die *Automatisierte Liegenschaftskarte* und das *Automatisierte Liegenschaftsbuch* zum neuen Datenmodell *des Amtlichen Liegenschaftskataster Informationssystems (ALKIS)* zusammengefaßt werden<sup>93</sup>. Dieses soll dann, zusammen mit *dem Amtlichen Topographisch-kartographischen Informationssystem (ATKIS)* und *dem Amtlichen Festpunktinformationssystem (AFIS)*, ein bundesweit einheitliches Gesamtkonzept darstellen.

---

<sup>89</sup> Übersicht unter: <http://www.lv-bw.de/lvshop2/ProduktInfo/wir-ueber-uns/vermbw.pdf>

<sup>90</sup> Beispiel unter: [http://www.lv-bw.de/lvshop2/ProduktInfo/geodaten/g\\_lik\\_a\\_lb.htm](http://www.lv-bw.de/lvshop2/ProduktInfo/geodaten/g_lik_a_lb.htm) bzw. [http://www.lv-bw.de/lvshop2/ProduktInfo/geodaten/g\\_lik\\_a\\_alk.htm](http://www.lv-bw.de/lvshop2/ProduktInfo/geodaten/g_lik_a_alk.htm)

<sup>91</sup> Homepage: <http://www.lv-bw.de/>

<sup>92</sup> Weitere Informationen unter: <http://www.lv-bw.de/lvshop2/hilfe/hilfe.html>

<sup>93</sup> Quelle: Reinhold Hummel (Wirtschaftsministerium Baden-Württemberg): Entwicklungstendenzen im zukunftsorientierten Liegenschaftskataster. Dieser Vortrag wurde im Rahmen des 4. Vermessungsingenieurtages 2002 an der HfT Stuttgart gehalten. Nachzulesen ist er unter: [http://www.fht-stuttgart.de/fbv/fbvweb/veranstaltungen/ingtag/Berichte/Bericht\\_Hummel.pdf](http://www.fht-stuttgart.de/fbv/fbvweb/veranstaltungen/ingtag/Berichte/Bericht_Hummel.pdf)

## 4.2 Das Automatisierte Liegenschaftsbuch ALB

Das ALB enthält die Angaben über Flurstücke und Gebäude, sowie die Ergebnisse der amtlichen Bodenschätzung bei landwirtschaftlichen Flächen. Die Eigentümer werden in Übereinstimmung mit dem Grundbuch nachrichtlich geführt. Das ALB liegt flächendeckend für ganz Baden-Württemberg vor.

Auszüge mit verschiedenen vordefinierten Inhaltszusammenstellungen auf Papier oder in den Datenformaten PDF<sup>94</sup>, CSV<sup>95</sup>, MS Excel97-Tabelle oder WLDGE<sup>96</sup> können online oder bei den zuständigen Ämtern bestellt werden<sup>97</sup>.

Weitere Auswertungen, z.B. eine Auswerteliste Jagdkataster, können auf Anfrage erstellt werden.

Dateien mit Zuordnungen von Gemarkungen zu Gemeinden, sowie weiteren Amtszugehörigkeiten (je mit Schlüssel des Statistischen Landesamts) und mit Straßen- und Gewannamen von Gemarkungen, werden ebenfalls auf Anfrage beim Landesvermessungsamt abgegeben (Regionaldaten).

Zur Aktualisierung früher bezogener ALB-Daten im Datenformat WLDGE werden auf Wunsch nur die seit dem letzten Datenbezug neu hinzugekommenen und geänderten Daten oder erneut der komplette Datenbestand abgegeben.

## 4.3 Die Automatisierte Liegenschaftskarte ALK

Die ALK wird landesweit als Punkt- und Grundrißdatei eingerichtet.

In der ALK-Punktdatei werden für die numerierten Punkte des Liegenschaftskatasters neben den Koordinaten im Gauß-Krüger-Koordinatensystem - und gegebenenfalls zusätzlich auch im Soldnersystem - Angaben über die Art, Qualität, Herkunft sowie die Ab- oder Vermarkung der Grenzpunkte geführt. Diese werden durch verschiedene Bemerkungen zum Punkt ergänzt.

In der ALK-Grundrißdatei werden Flurstücke, politische Grenzen und Katasterbezirksgrenzen, Gebäude, tatsächliche Nutzungen, topographische Objekte, Texte und Punktnummern zur Beschriftung, sowie nicht numerierte Punkte in digitaler Form gespeichert. Bei Texten wird unterschieden, ob es sich um Lagebezeichnungen bzw. Gemeindenamen oder um sonstige Texte mit oder ohne Fachbedeutung handelt.

Grundrißdaten werden als flächen-, linien- oder punktförmige Objekte geführt. Den Daten liegt das Gauß-Krüger-Meridianstreifensystem zugrunde. Die geometrische Definition der Objekte ergibt sich durch eine Folge von numerierten und nicht numerierten Punkten. Ein in der Punktdatei gespeicherter Punkt wird dabei stets mit seiner Punktnummer übergeben. Nicht numerierte Punkte aus der Grundrißdatei werden mit ihren Gauß-Krüger-Koordinaten übergeben.

Daneben werden auch attributive Daten, wie z.B. Entstehungsdatum und Qualitätsangaben der ALK-Grundrißobjekte, geführt. Informationen über die Genauigkeit der Objekte werden entweder als Sachda-

---

<sup>94</sup> Portable Document Format

<sup>95</sup> Comma-Separated Variables-File

<sup>96</sup> Workdatei Liegenschaftsbuch Daten-Gewinnung mit Entschlüsselungen

<sup>97</sup> Weitere Informationen unter:

[http://www.lv-bw.de/lvshop2/ProduktInfo/geodaten/g\\_lika\\_alb\\_auszuege.htm](http://www.lv-bw.de/lvshop2/ProduktInfo/geodaten/g_lika_alb_auszuege.htm)

tum zum Objekt übergeben oder lassen sich aus dem Lagestatus bzw. der Genauigkeitsstufe der verwendeten Grenzpunkte ableiten.

In vielen Gemarkungen liegen die ALK-Grundrißdaten bereits flächendeckend vor. Die Daten sind zum großen Teil durch vermessungstechnische Berechnungen entstanden und bieten damit bestmögliche Genauigkeit und Zuverlässigkeit. Die Datengewinnung wird durch die Digitalisierung großmaßstäbiger Flurkarten ergänzt. Abhängig von der Qualität der verwendeten Erfassungsgrundlage und dem gewählten Gebiet (Ortslage oder Feldlage) ergeben sich nachbarschaftsbezogen Lagegenauigkeiten von einigen Zentimetern bis zu einigen Dezimetern.

Punktdateien werden im Datenformat BGRUND oder druckaufbereitet als Punkteliste abgegeben.

Grundrißdaten werden als analoge Karte oder digital in den Vektorformaten *BGRUND*, *DXF*<sup>98</sup>, *SQD*<sup>99</sup> und *Shape* oder in den Rasterformaten TIFF, BMP und PNG abgegeben, wobei digitale Auszüge den vollständigen Inhalt der Punktdatei mit einschließt. Sonderauswertungen, wie eine Datei nur mit politischen Grenzen (Gemeinde- und Gemarkungsgrenzen) im Format BGRUND oder DXF, oder eine Flurnamenliste, sind auf Anfrage beim Landesvermessungsamt möglich.

Zur Aktualisierung eines früher bezogenen ALK-Datenbestandes werden auf Wunsch entweder nur die seit dem letzten Datenbezug neu hinzugekommenen und geänderten Objekte oder erneut der komplette Datenbestand abgegeben.

Die regelmäßige Laufendhaltung der ALK-Daten erfolgt durch die staatlichen Vermessungsämter bzw. städtischen Vermessungsdienststellen.

### 4.3.1 Das Format BGRUND

In Baden-Württemberg findet das BGRUND-Schnittstellenformat bei der Abgabe von digitalen Daten aus der Punkt- und Grundrißdatei, sowie beim Punktdatenaustausch zwischen den staatlichen Vermessungsämtern und Dritten (Öffentlich bestellte Vermessungsingenieure) Anwendung. Dabei werden die Daten antragsweise als sequentieller ASCII-Datenbestand in Sätzen mit der festen Länge von jeweils 50 Zeichen übergeben. Sofern die Information in einem Datensatz nicht abbildbar ist, wird sie bei Punktdaten in weiteren Folgedatensätzen übergeben. Jedes Grundrißobjekt wird in einer eigenen Datengruppe, die aus mehreren Datensätzen bestehen kann und der das zugehörige Steuerzeichen für den Objekttyp (vergleichbar mit der Ebene in kommerziellen grafischen Informationssystemen) vorangestellt ist, dargestellt. Innerhalb einer Datengruppe können leere Datenfelder auftreten und werden mit dem Endezeichen „E“ abgeschlossen.

Seit Ende Februar 2002 wird bei den Texten unterschieden, ob es sich bei diesen um eine Lagebezeichnung bzw. einen Gemeinidenamen oder um sonstige Texte mit oder ohne Fachbedeutung handelt. Im ersten Fall können zusätzlich der Gemeinideschlüssel bzw. Lageschlüssel der ALB-Regionaldatei als Sachdaten mit übergeben werden.

Seit dem gleichen Zeitpunkt können für Flurstücke optional die Sachdaten Lageschlüssel, Lagebezeichnung und Hinweise zum Flurstück aus dem ALB übergeben werden. Diese sind jedoch nicht im Standard-

---

<sup>98</sup> Drawing Exchange Format

<sup>99</sup> Sequentiell Data Format

auszug enthalten, sondern werden erst auf besonderen Antrag hinzugefügt. Analog dazu gilt eine optionale Erfassung der Sachdaten Hausnummer, Lageschlüssel und Lagebezeichnung aus dem ALB zu Gebäuden, die ebenfalls nur auf spezielle Anfrage den Daten beigegeben werden.

Bei den Sachdaten zu den Punkten wird zukünftig damit zu rechnen sein, daß Höhenangaben und UTM-Koordinaten eine gewichtigere Rolle einnehmen werden, die entsprechenden Folgedatensätze wurden jedenfalls inzwischen dementsprechend angelegt bzw. angepaßt. Für den Datenaustausch zwischen der Vermessungsverwaltung und den Öffentlich bestellten Vermessungsingenieuren wurde ein Format für Punkte mit vorübergehender Bedeutung festgelegt.

Momentan werden topographische Objekte noch als Linien in der Grundrißdatei vorgehalten, ab Juli 2003 werden sie hinsichtlich einer Migration nach ALKIS in punkt-, linien- oder flächenförmige Objekte überführt.

## 5 Das Programm BGRUNDviewer

### 5.1 Die Zielgruppe

Der BGRUNDviewer stellt in erster Linie ein Angebot für Privatpersonen und kleinere Gemeinden und Betriebe dar, die sich die örtliche Situation anhand der Daten des Liegenschaftskatasters anzeigen wollen, ohne sich dafür eine kostenintensive Spezialsoftware anschaffen zu müssen. Momentan ist dieser Personenkreis, was die Nutzungsmöglichkeiten der ALK-Daten angeht, noch sehr stark an eine analoge Form, sprich einem Ausdruck auf Papier, gebunden. Laut der Verordnung der Landesregierung über die Festsetzung der Gebührensätze für Amtshandlungen der staatlichen Behörden (Gebührenverordnung) fallen dabei bei einer Abgabe in Größe eines DIN A4-Blattes immerhin Kosten um die 20 € an. Demgegenüber steht ein Mindestpreis von etwa 50 € (prinzipiell abhängig von der Anzahl der beinhalteten Gebäude und Flurstücke) bei der Datenabgabe in digitaler Form, jedoch können diese Daten immer wieder verwendet werden und erfahren bei sachgemäßer Behandlung keine materielle Alterung.

Wenn man sich nun also digitale Daten im BGRUND-Format besorgt hat, kann man mit dem über die Webseite [www.BGRUNDviewer.de](http://www.BGRUNDviewer.de) erreichbaren Programm eine mittels Plug-In im Internet-Browser darstellbare Karte im SVG-Format erstellen, inklusive den zugehörigen HTML-Dateien mit zusätzlichen Angaben zu den Vordaten, den enthaltenen Punkten, Flurstücken und Grenzzügen, Gebäuden, topographischen Objekten und Texten. Diese Dateien können anschließend auf den Client heruntergeladen werden, um immer wieder verwendet zu werden ohne die Internetseite erneut aufrufen zu müssen.

Denkbar wäre auch die Kombination aus dem Geodaten-Informationssystem GEODIS des Landesvermessungsamtes Baden-Württemberg und dem BGRUNDviewer, wodurch sich der Anwender einen visuellen Überblick über die gewünschten Daten verschaffen kann, bevor er sie dann schließlich endgültig beantragt.

Eine weitere Einsatzmöglichkeit besteht in der Anwendung, auch über Intranet denkbar, bei den staatlichen Vermessungsämtern, um einerseits für den Dienstgebrauch BGRUND-Informationen zu visualisieren und andererseits das Personal bei der Beratung von Kunden zu unterstützen.

### 5.2 Systemanforderungen

Da PHP-Skripte zur Laufzeit interpretiert werden, laufen damit erstellte Programme bei weitem nicht so schnell ab wie vergleichbare Lösungen mit Java oder gar C bzw. C++. Um die Wartezeiten auf ein erträgliches Maß zu halten und den Server nicht übermäßig zu beanspruchen, wurde daher die Größe der umzuwandelnden BGRUND-Datei auf 650 Kilobyte begrenzt. Als Entwicklungsumgebung wurde ein Windows NT-Rechner mit 450 MHz und 128 Megabyte eingesetzt, der für die Umsetzung einer Ausgangsdatei mit ca. 620 KB etwa 45 Sekunden benötigt. Dabei wird das erzeugte SVG-Dokument etwa 860 KB groß. Der moderne Server der Internetseite [www.BGRUNDviewer.de](http://www.BGRUNDviewer.de) brauchte für die Verarbeitung, inklusive des Datentransfers, ungefähr 20 Sekunden. Dafür gibt es drei Erklärungen: erstens läuft der Webserver der Internetseite auf einem LINUX-Betriebssystem, mit dem der Apache-Server prinzipiell eine

bessere Leistung erzielt als unter Windows. Zweitens wird dort PHP über ein Apache-Modul integriert, während bei der Entwicklungsumgebung PHP als CGI-Version eingebunden wurde. Als letzter und entscheidender Grund kann angeführt werden, daß bei dem Internet-Server mit Sicherheit ein leistungsfähigerer Prozessor eingesetzt wird.

Dieser spielt auch auf Seiten des Clients eine bedeutende Rolle. So vergehen auf einem Rechner mit 450 MHz etwa 10 Sekunden bis die produzierte SVG-Grafik (mit 860 KB) durch den Internet-Browser und das Plug-In vollständig geparkt und angezeigt wird. Dem gegenüber stehen 3 Sekunden bei einem Client mit einem Prozessortakt von 1,7 GHz.

PHP läßt sich zwar auch mit anderer Server-Software kombinieren, jedoch wird es empfohlen weiterhin den Apache zu benutzen, da ausschließlich mit ihm das Programm entwickelt und getestet wurde. Die verwendete Version war 2.0.x, jedoch sollten Versionen ab 1.3.x ausreichen. Bei PHP selbst muß mindestens Version 4.0 benutzt werden, da erst von da an die objektorientierten Ansätze eingebettet sind, auf die beim BGRUNDviewer zugegriffen wird.

Trotz der Absicht, dem Benutzer bei der Wahl des Internet-Browsers möglichst freie Wahl zu lassen, ergaben sich im Laufe der Diplomarbeit einige mehr oder weniger große Probleme in Abhängigkeit von der verwendeten Software. Unter Windows ließen sich für folgende Browser die aufgeführten Verhaltensweisen feststellen:

Tabelle 7: Verhalten der Internet-Browser bei Verwendung des SVG Viewer 3.0

Netscape Communicator ab Version 4.7	Keine Probleme bekannt
Netscape ab Version 6.0	Die automatische Installation verläuft nicht fehlerfrei. Es müssen nachträglich die Dateien NPSVG3.dll und NPSVG3.zip auf dem Rechner gesucht und in das Plugins-Verzeichnis (z.B. C:\Program Files\Netscape\Netscape 6\Plugins) des Browsers verschoben werden.
Internet Explorer ab Version 4.0	Keine Probleme bekannt
Internet Explorer ab Version 5.5	Beim Ausdruck hält sich der Browser nicht an die vorgegebenen Dimensionen des SVG-Bereiches, sondern paßt ihn nach eigenem Ermessen an, wodurch eine Verzerrung entsteht.
Opera ab Version 6	JavaScript wird nicht unterstützt, somit kein Zoomen, Verschieben des Ausschnittes und Ein- und Ausblenden von Ebenen möglich.
Opera ab Version 7	SVG kann nicht ausgedruckt werden, der Browser gibt immer nur ein leeres Blatt aus.

Der SVG Viewer 3.0 von Adobe stellt wohl das zur Zeit beste Plug-In für SVG-Grafiken dar. Er ist als kostenloser Download verfügbar und beinhaltet seit der Version 3.0 eine eigene Script-Engine, die anstelle des Browsers die Laufzeitumgebung für JavaScript übernimmt.

Wie in der vorhergehenden Tabelle zu erkennen ist, arbeitet der SVG Viewer noch nicht mit allen Browsern zusammen und selbst bei denen die unterstützt werden gibt es immer wieder Probleme. Wenn sich

der SVG-Standard allerdings weiter so entwickelt und verbreitet wie er es seit seiner offiziellen Veröffentlichung getan hat, kann mit Sicherheit davon ausgegangen werden, daß die allgemein gebräuchlichen Browser nach und nach eine eigene Unterstützung für SVG-Grafiken beinhalten werden. Erste Anzeichen dafür zeigen sich beim Mozilla-Browser<sup>100</sup>.

## 5.3 Einlesen und Verarbeiten der BGRUND-Datei

### 5.3.1 Dateiupload vom Client zum Server

Da der BGRUNDviewer selbst keinen Zugriff auf Daten der Vermessungsverwaltung anbietet, muß jeder Benutzer seine eigenen Dateien zur Verfügung stellen. Um diese von seinem Client auf den Server übertragen zu können, steht auf der Webseite ein HTML-Formular zur Verfügung mit dem der Upload durchgeführt werden kann.

```
...  
<form name="fileForm" enctype="multipart/form-data"  
action="dateiauslesen.php" method="POST">  
<input type="hidden" name="MAX_FILE_SIZE" value="650000">  
<input type="file" name="fupload">  
<input type="submit" value="Verschicke Datei!">  
</form>  
...
```

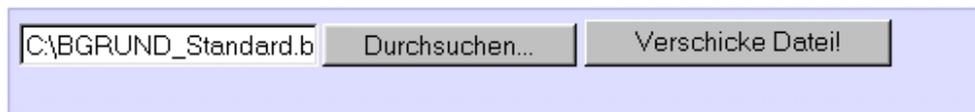


Abbildung 30: Formularfeld für Dateiupload

Durch das Feld input mit dem Typ file wird auf der Webseite ein Eingabefeld erstellt, in dem die hochzuladende Datei und ihr Pfad auf dem Rechner angegeben werden kann. Zusätzlich wird zur Erleichterung ein Durchsuchen-Knopf angeboten der eine bequeme Auswahl der gewünschten Datei ermöglicht.

---

<sup>100</sup> Das Mozilla SVG Projekt: <http://www.mozilla.org/projects/svg/>



Abbildung 31: Auswahl der Datei

PHP verlangt, daß vor dem Feld das die Datei hochlädt, ein verborgenes Feld<sup>101</sup> eingefügt wird, das die Bezeichnung `MAX_FILE_SIZE` trägt und als Wert die maximal zulässige Dateigröße in Byte<sup>102</sup> angibt. Diese Dateigröße kann allerdings die im Feld `upload_max_filesize` in der Konfigurationsdatei `php.ini` angegebene Dateigröße nicht überschreiten.

Wird nun das Formular über den `submit`-Knopf abgeschickt, wird die Datei hochgeladen und die Seite `dateiauslesen.php` aufgerufen. Dort kann PHP auf diese Datei als globale Variable unter dem im Formular angegebenen Namen<sup>103</sup> zugreifen und steht somit zur weiteren Verarbeitung zur Verfügung.

Da nie sicher ist, ob die hochgeladene Datei nicht doch zu groß war, eine gar nicht vorhandene Datei angegeben wurde oder die Datei überhaupt nicht dem BGRUND-Format entspricht, werden gleich nach der Übertragung Kontrollen dafür durchgeführt. Erkennt das Programm Fehler, wird der Benutzer mit dem `header`-Befehl auf die Seite mit dem Dateiapload zurück geleitet und eine entsprechende Meldung ausgegeben.

Um den Test auf BGRUND-Konformität durchzuführen und überhaupt die Datei zum Lesen zu öffnen, wird die Anweisung `$fp=fopen($fupload, "r")` ausgeführt, wodurch das Programm einen Zeiger (`$fp`) auf den Textanfang erzeugt. Der Parameter `r` gibt dabei an, daß die entsprechende Datei nur gelesen werden kann.

Anschließend wird auf dem Server ein Verzeichnis für den Benutzer angelegt und die benötigten Dateien dort hinein kopiert bzw. als neues Dokument erstellt. Des weiteren wird in der Datenbank eine Tabelle erstellt, die Felder für den Punktnumerierungsbezirk, die Leitpunktnummer und die Folgepunktnummer eines numerierten Punktes enthält.

<sup>101</sup> `<input type="hidden" ... >`

<sup>102</sup> hier: 650000

<sup>103</sup> hier: `fupload`, durch die Variablenkennzeichnung in PHP wird daraus jedoch `$fupload`

### 5.3.2 Vordaten

In den ersten Zeilen einer BGRUND-Datei werden die sogenannten Vordaten angegeben. In den Vordaten werden die Antragsnummer, interne Verwaltungsangaben, das Abrufdatum, der Rahmen des Bearbeitungs- /Abrufgebiets, das Koordinatensystem GK und die Nummern der betroffenen Punktnummerierungsbezirke einschließlich ihrer Verschlüsselung übergeben.

Um diese Vordaten berücksichtigen zu können, wurde in PHP eine entsprechende Objektklasse definiert. Somit läßt sich bei jedem Umsetzungsvorgang eine Instanz der Klasse Vordaten bilden und deren Eigenschaften mit den zeilenweise eingelesenen Angaben aus der BGRUND-Datei belegen.

Tabelle 8: Die Klasse Vordaten

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Antragsnummer</li> <li>• Verwaltungsangaben</li> <li>• Schnittstellenversion</li> <li>• Abrufdatum</li> <li>• Abrufrahmen (Array)</li> <li>• Punktnummernbereiche (Array)</li> <li>• Datenabgebende Stelle</li> <li>• Umfang der Datenbereitstellung</li> <li>• Datum der letzten Datenabgabe</li> <li>• Auszugstyp</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> <li>• Ausgabe in SVG-Datei</li> </ul>

### 5.3.3 Punkte

Den Vordaten folgend, eingerahmt von den Textmarken PUNKTDATEN und PUNKTDATENENDE erfolgt eine Auflistung aller abgerufenen bzw. zum Abrufgebiet gehörenden Punktdaten. Dabei sind die Punkte nach steigendem Rechtswert sortiert, d.h. *Trigonometrische Punkte (TP)*, *Aufnahmepunkte (AP)*, *Schnitt- und Kleinpunkte (SK)*, *Versicherungspunkte (VP)*, sonstige numerierte Punkte und nicht numerierte Punkte sind wahllos miteinander vermischt. Da die unterschiedlichen Punktarten jedoch teils gleiche, teils unterschiedliche Eigenschaften besitzen, mußte ein Schema erdacht werden, das sowohl einen eingelesenen Punkt eindeutig einer Punktart zuweist, als auch die Beziehungen der Punktarten untereinander umsetzt. Daraus entstanden folgende Punktklassen:

Tabelle 9: Die Klasse Punkt

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Punktart</li> <li>• Rechtswert</li> <li>• Hochwert</li> <li>• Angabe, ob Grenz- oder Kreismittelpunkt</li> <li>• Angabe, ob Gebäudepunkt</li> <li>• Angabe, ob Bestandteil eines topographischen Objektes</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> <li>• Zeichnen in SVG-Datei</li> <li>• Ausgabe der Punktnummer in SVG-Datei</li> </ul>

- |   |
|---|
| • Lagestatus bzw. Genauigkeitsstufe bei nicht numerierten Punkten |
|---|

Tabelle 10: Die Klasse nicht numerierter Punkt

Abgeleitet von Klasse: Punkt	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Abmarkung</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> </ul>

Tabelle 11: Die Klasse numerierter Punkt

Abgeleitet von Klasse: Punkt	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Punktstatus</li> <li>• Quellenhinweis</li> <li>• Hinweis auf Gerade</li> <li>• Angabe, ob Kreismittelpunkt</li> <li>• Gemarkungsnummer</li> <li>• Art der Abmarkung</li> <li>• Höhe bzw. Tiefe der Ab-/Vermarkung</li> <li>• Behörde bei zeitweilig ausgesetzter Abmarkung</li> <li>• Gemarkungsnummer des Aktenhinweises</li> <li>• Aktenhinweis</li> <li>• Zuständige Stelle</li> <li>• Datum der Speicherung</li> <li>• Verweis auf zugehörige Versicherungspunkte (Array)</li> <li>• Verweis auf zugehörige Kreismittelpunkte (Array)</li> <li>• Verweis auf besondere Nummer als Landesgrenzpunkt</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> </ul>

Tabelle 12: Die Klasse Trigonometrischer Punkt

Abgeleitet von Klasse: numerierter Punkt	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Bezeichnung der TK25</li> <li>• TP-Nummer</li> <li>• Unternummer</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> </ul>

Tabelle 13: Die Klasse allgemeiner Punkt

Abgeleitet von Klasse: numerierter Punkt	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Punktnumerierungsbezirk</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> </ul>

<ul style="list-style-type: none"> <li>• Leitpunktnummer</li> <li>• Folgepunktnummer</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgabe in HTML-Datei</li> </ul>
---	---

Dabei spielt die Klasse Punkt eine dominierende Rolle, denn alle anderen Klassen leiten sich davon ab. Die erste Aufteilung erfolgt in die Gruppen der numerierten und der nicht numerierten Punkte, wobei die Klasse der numerierten Punkte wiederum in Trigonometrische Punkte und allgemeine Punkte aufgegliedert wurde. Der Kategorie der allgemeinen Punkte wurden alle Aufnahmepunkte, Schnitt- und Kleinpunkte, Versicherungspunkte, Grenzpunkte, Gebäudepunkte und Punkte mit topographischer Bedeutung zugewiesen. Auf diese Weise können Eigenschaften und Methoden der übergeordneten Klassen verwendet werden, wie etwa die Ausgabe in das SVG-Dokument, die nur einmal zentral in der obersten Punktklasse realisiert wurde. Aber auch bei den Konstruktoren und der Ausgabe in der HTML-Datei werden die jeweils entsprechenden Methoden der Elternklasse aufgerufen.

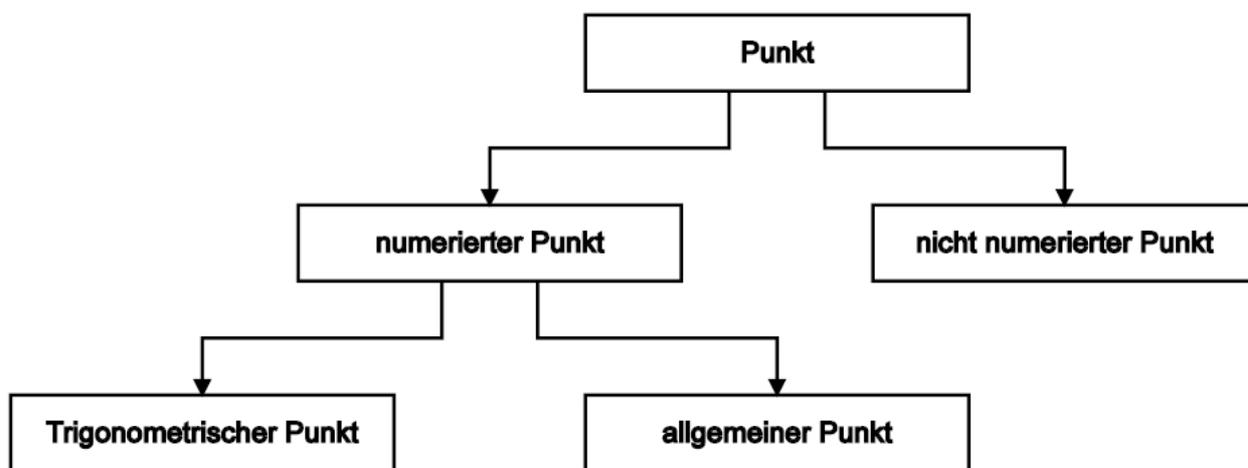


Abbildung 32: Die Hierarchie der Punktartenklassen

Um die Punkte verwalten zu können wurde eine weitere Klasse für die Punkteliste angelegt, die Angaben zu den minimalen und maximalen Koordinatenwerten erfaßt und für jede Punktart eine separate Liste (Array) zur Punktaufnahme anbietet.

Tabelle 14: Die Klasse Punkteliste

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Minimaler Rechtswert</li> <li>• Maximaler Rechtswert</li> <li>• Minimaler Hochwert</li> <li>• Maximaler Hochwert</li> <li>• Liste der Trigonometrischen Punkte (Array)</li> <li>• Liste der Aufnahmepunkte (Array)</li> <li>• Liste der Schnitt- und Kleinpunkte (Array)</li> <li>• Liste der Versicherungspunkte (Array)</li> <li>• Liste der allgemeinen Punkte (Array)</li> <li>• Liste der nicht numerierten Punkte (Array)</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgabe aller Punkte in HTML-Datei</li> <li>• Zeichnen aller Punkte in SVG-Datei, inklusive Punkt-nummern</li> <li>• Suche nach einem bestimmten Punkt in den Listen</li> <li>• Das Ermitteln der Koordinaten eines bestimmten Punktes</li> </ul>

### 5.3.4 Flurstücke und Grenzzüge

Nach den Angaben aus der Punktdatei folgen die Grundrißdaten, wobei die Flurstücke den Anfang bilden. Allen Grundrißdaten sind Angaben zur Gemarkung und Flur vorangestellt. Innerhalb eines Antrags können Daten mehrerer Flurstücksnumerierungsbezirke vorkommen, die dann jeweils blockweise abgehandelt werden.

Flurstücke werden als flächenförmige Objekte geführt und übergeben. Dies zeigt sich daran, daß der Linienzug des Umrings als eine Folge von Punktnummern, Koordinatenpaaren<sup>104</sup> und Kreisbogenparametern definiert wird. Da auch bei den Grenzzügen, Nutzungsflächen, Gebäuden und topographischen Objekten immer mindestens zwei Punkte und ein eventuell dazwischen liegender Kreisbogen vorhanden ist, wurde eine Klasse Linienzug angelegt, mit der diese Werte aufgenommen und weiterverarbeitet werden können.

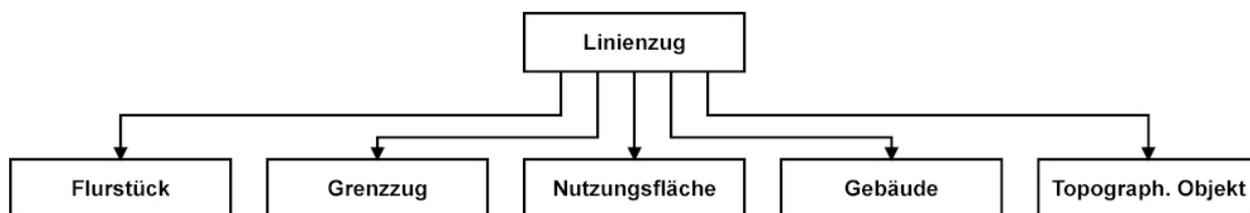


Abbildung 33: Die Klasse Linienzug als Elternklasse

Tabelle 15: Die Klasse Linienzug

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>Liste der beschreibenden Punkte bzw. Kreisbögen (Array)</li> </ul>	<ul style="list-style-type: none"> <li>Konstruktor</li> <li>Ausgabe in HTML-Datei</li> <li>Zeichnen in SVG-Datei</li> </ul>

Die einzelnen Elemente der Punkteliste können entweder

- Kreisbögen mit den Angaben: Radius und Lage des Kreismittelpunktes bezüglich der Sehne,
- Trigonometrische Punkte mit den Angaben: Blattbezeichnung der TK25, TP-Nummer und Unternummer,
- Numerierte Punkte mit den Angaben: Schlüssel des Punktnumerierungsbezirkes, Leitpunktnummer und Folgepunktnummer oder
- Nicht numerierte Punkte mit den Angaben: Rechtswert und Hochwert, sein.

Bei der Ermittlung der Kreisbögenparameter ist eine wichtige Tatsache hervorzuheben: in der BGRUND-Datei wird bei kreisförmigen Grenzen mit einem Kreismittelpunkt nach dem Bogenanfang das Steuerzeichen „K+“ bzw. „K-“ übergeben, wenn der Kreisbogen links bzw. rechts von der Sehne liegt. Bei Kreisbögen die durch den Radius anstelle des Kreismittelpunktes definiert werden, gilt hingegen entsprechend der Schnittstellenbeschreibung eigentlich der Parameter „L“ bzw. „R“ als Hinweis darauf, ob der fiktive

<sup>104</sup> bei nicht numerierten Punkten

Kreismittelpunkt links oder rechts der Sehne liegt. Allerdings wird in allen Fällen, in denen der fiktive Kreismittelpunkt genau auf der Sehne liegt, stets „L“ ausgegeben, was teilweise dazu führt, daß der Kreisbogen auf der falschen Seite gezeichnet wird. Dies machte es nötig, daß beim Einlesen die Angaben „L“ bzw. „R“ durch die jeweiligen Werte „K+“ bzw. „K-“ kontrolliert und gegebenenfalls korrigiert werden.

Durch die Elternklasse Linienzug wird der Klasse Flurstück ein Großteil der Zuständigkeit abgenommen. Mit den Angaben zum Flurstücksnumerierungsbezirk, der Darstellung und Position der zugehörigen Nummer und den Sachdaten zum Objekt ist diese Klassenbeschreibung trotzdem recht umfangreich.

Tabelle 16: Die Klasse Flurstück

Abgeleitet von Klasse: Linienzug	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Gemarkungsnummer</li> <li>• Flurnummer</li> <li>• Gemarkungsname</li> <li>• Stammnummer</li> <li>• Unternummer</li> <li>• Datum der Objektbildung</li> <li>• Angaben zur Darstellung und Position der Flurstücksnummer<sup>105</sup> für Maßstäbe 1:1 bis 1:1000 (Array)</li> <li>• Angaben zur Darstellung und Position der Flurstücksnummer für Maßstäbe 1:1000 bis 1:2500 (Array)</li> <li>• Lageschlüssel und Lagenname nach Regionaldatei<sup>106</sup> (Array)</li> <li>• Hinweise zum Flurstück aus dem ALB<sup>107</sup> (Array)</li> <li>• Enthaltene Nutzungsflächen (Array)</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> <li>• Zeichnen des Umringes in SVG-Datei</li> <li>• Ausgabe der Flurstücksnummer in SVG-Datei</li> <li>• Zeichnen der Umringe der enthaltenen Nutzungsflächen</li> <li>• Enthaltene Nutzungsflächen in SVG-Datei farblich anlegen</li> <li>• Ausgabe der Abkürzungen für die enthaltenen Nutzungsflächen in SVG-Datei</li> </ul>

In BGRUND werden auch unvollständig gespeicherte Flurstücke, sowie politische Grenzen und Katasterbezirksgrenzen übergeben. Diese werden als Grenzzüge bezeichnet, und die dafür zuständige Klasse gleichen Namens ist wie die der Flurstücke, ebenfalls von der Elternklasse Linienzug abgeleitet.

Tabelle 17: Die Klasse Grenzzug

Abgeleitet von Klasse: Linienzug	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Objektschlüssel für die Art des Grenzzuges</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> </ul>

<sup>105</sup> enthält die Angabe, ob die Nummer gewöhnlich, mit Bruchstrich, nur als Unternummer oder überhaupt nicht dargestellt werden soll. Ferner können optional die absoluten Koordinaten für die Platzierung und einen Drehwinkel angegeben werden, sowie die GK-Koordinaten für einen Zuordnungspfeil.

<sup>106</sup> Wird nur bei der Ausgabe auf besondere Anforderung aus dem ALB bereitgestellt. Enthalten sind der Gemeindegeschlüssel, der Lageschlüssel, der Lagenname und eventuell die Hausnummer.

<sup>107</sup> Wird nur bei der Ausgabe auf besondere Anforderung aus dem ALB bereitgestellt. Der Eintrag kann ein Hinweis auf eine Umlegung, eine Flurneuordnung oder eine Zugehörigkeit zu einer Jagdgenossenschaft sein, gegebenenfalls schließt es auch die Verfahrensnummer mit ein.

- |                         |
|-------------------------|
| • Zeichnen in SVG-Datei |
|-------------------------|

Sofern nichts anderes angegeben ist, wird für die Art des Grenzzuges eine Flurstücksgrenze angenommen. Der Grenzzug könnte aber genauso gut Teil einer Bundes-, Landes-, Regierungsbezirks-, Regionen-, Kreis-, Gemeinde-, Gemarkungs- oder Flurgrenze sein.

Flurstücke und Grenzzüge eines Antrags werden über eine Instanz der Klasse Flurstücksliste verwaltet.

Tabelle 18: Die Klasse Flurstücksliste

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Liste der Flurstücke (Array)</li> <li>• Liste der Grenzzüge (Array)</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgabe aller Flurstücke und Grenzzüge in HTML-Datei</li> </ul>

### 5.3.5 Flächen der tatsächlichen Nutzung

Tatsächliche Nutzungen werden als flächenförmige Objekte flurstücksbezogen geführt, d.h. sie dürfen die Flurstücksgrenzen nicht überschreiten. Bei der Ausgabe wird ihre Geometrie zusammen mit der Nummer des Bezugsflurstücks und der tatsächlichen Nutzung übergeben. Alle weiteren attributiven Daten zur tatsächlichen Nutzung werden in einem oder mehreren Sachdatensätzen ausgegeben.

Tabelle 19: Die Klasse Nutzungsfläche

Abgeleitet von Klasse: Linienzug	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Schlüssel für die Art der tatsächlichen Nutzung</li> <li>• Datum der Objektbildung</li> <li>• Darstellung der Nutzungsart<sup>108</sup></li> <li>• Rechtswert für Positionierung der tat. Nutzung</li> <li>• Hochwert für Positionierung der tat. Nutzung</li> <li>• Drehwinkel für Positionierung der tat. Nutzung</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> <li>• Zeichnen des Umrings in SVG-Datei</li> <li>• Fläche in SVG-Datei farblich anlegen</li> <li>• Ausgabe der Abkürzung in SVG-Datei</li> </ul>

Liegt nur eine Nutzungsfläche für das entsprechende Flurstück vor, gilt natürlich der Umring des Flurstücks ebenfalls als Begrenzung der Nutzungsfläche und wird nicht separat aufgeführt, wie es bei mehreren Nutzungen auf einem Flurstück der Fall ist. Da ferner die Nutzungsflächen nicht flächendeckend vorliegen, kann es auch vorkommen, daß Flurstücke ohne Angaben zu ihren Nutzungen ausgegeben werden.

### 5.3.6 Gebäude

Gebäude werden ebenfalls als flächenförmige Objekte geführt. Bei der Ausgabe wird deren Geometrie zusammen mit der Nummer des Bezugsflurstücks und der Gebäudenutzung übergeben. Alle weiteren attributiven Daten zum Gebäude werden in einem oder mehreren Sachdatensätzen ausgegeben.

<sup>108</sup> Keine Darstellung oder Ausgabe der Abkürzung für die Art der tatsächlichen Nutzung

Tabelle 20: Die Klasse Gebäude

Abgeleitet von Klasse: Linienzug	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Gemarkungsnummer</li> <li>• Flurnummer</li> <li>• Gemarkungsname</li> <li>• Stammnummer des Bezugflurstücks</li> <li>• Unternummer des Bezugflurstücks</li> <li>• Objektschlüssel der Gebäudenutzung</li> <li>• Datum der Objektbildung</li> <li>• Genauigkeitsstufe</li> <li>• Position<sup>109</sup> der Gebäudenutzung (Array)</li> <li>• Art der Schraffur<sup>110</sup></li> <li>• Eigenname<sup>111</sup> des Gebäudes für Maßstäbe 1:1 bis 1:1000 (Array)</li> <li>• Eigenname des Gebäudes für Maßstäbe 1:1001 bis 1:2500 (Array)</li> <li>• Hausnummer<sup>112</sup> des Gebäudes für Maßstäbe 1:1 bis 1:1000 (Array)</li> <li>• Hausnummer des Gebäudes für Maßstäbe 1:1001 bis 1:2500 (Array)</li> <li>• Liste der Ausgestaltungen (Array)</li> <li>• Lageschlüssel und Lagenname<sup>113</sup> nach Regionaldatei (Array)</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> <li>• Zeichnen des Umringes in SVG-Datei</li> <li>• Zeichnen der Ausgestaltungen in SVG-Datei</li> <li>• Gebäudefläche in SVG-Datei farbig anlegen</li> <li>• Ausgabe der Gebäudenutzung in SVG-Datei</li> <li>• Ausgabe der Hausnummer in SVG-Datei</li> <li>• Ausgabe des Eigennamens in SVG-Datei</li> </ul>

Die graphische Ausgestaltung einzelner Gebäudeseiten bzw. des gesamten Gebäudes bei der Präsentation kann eine der folgenden Bedeutungen besitzen:

- Hochhausbegrenzungslinie
- Offene Begrenzungslinie
- Begrenzung eines unterirdischen Gebäudes
- Durchfahrt im Gebäude
- Durchfahrt an überbauter Verkehrsstraße
- Arkade
- Überdachung, offene Halle
- Auskragende Geschosse

<sup>109</sup> Rechtswert, Hochwert und eventuell Drehwinkel oder Anweisung zur Unterdrückung der Ausgabe

<sup>110</sup> Parallel bzw. senkrecht zur ersten Gebäudeseite, im 45° Winkel dazu mit dünnen oder dicken Schraffurlinien oder gar keine Schraffur

<sup>111</sup> Inklusive der absoluten Position (Rechtswert, Hochwert und gegebenenfalls Drehwinkel)

<sup>112</sup> Enthält neben Hausnummer, Rechtswert, Hochwert und eventuell Drehwinkel Angaben zur Darstellung. Möglich sind eine gewöhnliche Darstellung, Unterdrückung der Ausgabe, Darstellung der Unternummer mit Bruchstrich oder Unterdrückung der Stammnummer

<sup>113</sup> Enthalten sind der Gemeindeschlüssel und der Lageschlüssel nach der Regionaldatei sowie der Lagenname. Dabei gehen diese Angaben immer unmittelbar dem Sachdatensatz mit der Hausnummer voran.

- Schornstein im Gebäude (sofern weit sichtbar)

Die graphische Ausgestaltung bezieht sich immer auf die nachfolgend übergebene Gebäudeseite oder Linie innerhalb des Gebäudes bzw. den nachfolgend übergebenen Punkt<sup>114</sup>.

Für die Aufnahme der Gebäude in einem Array wurde die Klasse Gebäudeliste eingerichtet, die auch eine komplette Ausgabe in eine HTML-Datei einleitet.

Tabelle 21: Die Klasse Gebäudeliste

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Liste der Gebäude (Array)</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgabe aller Gebäude in HTML-Datei</li> </ul>

### 5.3.7 Topographische Objekte

Ab Juli 2003 sollen die topographischen Objekte in einem neuen Format geführt werden. Bislang werden sie ausschließlich als linienförmige Objekte in der Grundrißdatei gehalten, nach dem neuen Format wird eine Unterscheidung in punktförmige, linienförmige und flächenförmige topographische Objekte ermöglicht werden. Zusätzlich können von da an zu jedem topographischen Objekt Sachdaten erfaßt und ausgegeben werden. Da jedoch während der Erstellung des BGRUNDviewers nicht genügend Testdaten mit dem kommenden Format zur Verfügung standen, beschränkt sich der momentane Umfang des Programmes bisher nur auf das alte Format.

Tabelle 22: Die Klasse topographisches Objekt nach altem Format

Abgeleitet von Klasse: Linienzug	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Objektschlüssel für die Art des topographischen Objekts</li> <li>• Datum der Objektbildung</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> <li>• Zeichnen in SVG-Datei</li> </ul>

Dabei können Objektschlüssel für folgende Arten der Topographie angegeben werden:

- Grenze der tatsächlichen Nutzung, wenn noch keine flächenförmigen Objekte tatsächliche Nutzung gebildet sind
- (überdeckte) topographische Linie
- (un)deutliche Böschungsoberkante
- (un)deutliche Böschungsunterkante
- offene Gebäudebegrenzungslinie, solange noch nicht als Sachdatum zum Gebäude geführt
- ober-/unterirdische Fernwasserleitung
- ober-/unterirdische Erdgas-/Heizgasleitung
- ober-/unterirdische Rohölleitung
- ober-/unterirdische Fernheizleitung
- oberirdische Hochspannungsleitung außerhalb/innerhalb der Ortslage

<sup>114</sup> Im Falle eines Schornsteines

- unterirdische Hochspannungsleitung
- (überdeckter) Fahrbahnrand

Ein großer Nachteil des alten BGRUND-Formates bei den topographischen Objekten ist, daß diese sortiert nach aufsteigendem Rechtswert des Anfangspunktes ausgegeben werden und somit der Zusammenhang mehrerer Objekte, beispielsweise bei Böschungsflächen verloren geht, wodurch sich nur unter erheblichem Aufwand eine Verknüpfung wiederherstellen ließe.

Tabelle 23: Die Klasse Liste der topographischen Objekte

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Liste der topographischen Objekte nach altem Format (Array)</li> <li>• Liste der punktförmigen topographischen Objekte (Array)</li> <li>• Liste der linienförmigen topographischen Objekte (Array)</li> <li>• Liste der flächenförmigen topographischen Objekte (Array)</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgabe aller topographischer Objekte in HTML-Datei</li> </ul>

Es wurden bei der Liste der topographischen Objekte bereits die Arrays für die Aufnahme der Topographie nach dem neuen Format angelegt, da diese in einem späteren Zeitpunkt ebenfalls in dem Programmsystem BGRUNDviewer berücksichtigt werden sollen.

### 5.3.8 Texte

Seit Ende Februar 2002 werden Texte in BGRUND in zwei Gruppen eingeteilt. Zum einen liegen Steuerzeichen für Lagebezeichnungen und Gemeindenamen vor, und zum anderen können sonstige Texte mit oder ohne Fachbedeutung geführt werden.

Unter die Kategorie der sonstigen Texte fallen zum Beispiel Flurstücksnummern von nicht vollständig gebildeten Flurstücken, sowie die Abkürzung von tatsächlichen Nutzungen, falls die Nutzungsfläche nicht komplett enthalten ist. Aber auch Klassifizierungen von Straßen und Gewässern können auf diese Weise angegeben werden.

Tabelle 24: Die Klasse Text

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Objektschlüssel für die Schriftformatierung bei festgelegten Maßstabsbereichen</li> <li>• Der eigentliche Text</li> <li>• Rechtswert für die Positionierung</li> <li>• Hochwert für die Positionierung</li> <li>• Drehwinkel für die Positionierung</li> <li>• Speicherungsdatum</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> <li>• Zeichnen in SVG-Datei</li> </ul>

Lage- und Gemeindenamen werden eingesetzt um etwa Gemeinde-, Ortsteil oder Gewannamen, aber auch Straßen- oder Gewässernamen in der Karte anzuzeigen.

Tabelle 25: Die Klasse Lage- und Gemeindegname

Abgeleitet von Klasse: Text	
Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Gemeindegchlüssel nach Regionaldatei</li> <li>• Lageschlüssel nach Regionaldatei</li> <li>• Lage- bzw. Gemeindegname nach Regionaldatei</li> </ul>	<ul style="list-style-type: none"> <li>• Konstruktor</li> <li>• Ausgabe in HTML-Datei</li> </ul>

Für die Speicherung der Instanzen aus den Klassen Text und Lage- und Gemeindegname stehen in der Klasse Texteliste Arrays zur Verfügung. Diese Klasse übernimmt auch die Ausgabe in die dafür vorgesehene HTML-Datei.

Tabelle 26: Die Klasse Texteliste

Eigenschaften	Methoden
<ul style="list-style-type: none"> <li>• Liste der Lage- und Gemeindegnamen (Array)</li> <li>• Liste der sonstigen Texte (Array)</li> </ul>	<ul style="list-style-type: none"> <li>• Ausgabe aller Lage- und Gemeindegtexte und der sonstigen Texte in HTML-Datei</li> </ul>

### 5.3.9 Die Zusammenfassung der Verarbeitung

Nachdem das Programm die komplette BGRUND-Datei eingelesen und verarbeitet sowie alle benötigten Dateien erstellt hat, erfolgt eine Auflistung aller gebildeten Elemente.

# BGRUNDviewer

Startseite	2	Trigonometrische Punkte
Voraussetzungen	2	Aufnahmepunkte
Demonstration	0	Schnitt- und Kleinpunkte
zum Programm	4	Versicherungspunkte
E-Mail	419	sonstige Punkte
	269	nicht numerierte Punkte
	53	Flurstücke
	51	Grenzzüge
	56	Gebäude
	42	topographische Objekte nach altem Format
	0	punktförmige topographische Objekte nach neuem Format
	0	linienförmige topographische Objekte nach neuem Format
	0	flächenförmige topographische Objekte nach neuem Format
	11	Lage- und Gemeindennamen
	12	sonstige Texte

Abbildung 34: Die Zusammenfassung der Verarbeitung

Von dieser Seite aus kann man ebenfalls über Hyperlinks sämtliche HTML-Seiten mit den Informationen zu den erschaffenen Elementen wie auch die erzeugte Karte aufrufen.

## 5.4 Erstellen der Grafik und der HTML-Seiten

### 5.4.1 Das SVG-Gerüst

Scalable Vector Graphics erlaubt es innerhalb des zentralen Root-Elements, also dem äußersten SVG-Element, weitere SVG-Objekte aufzuführen. Dies wurde bei diesem Projekt dazu benutzt um einen SVG-Bereich für die Kartendarstellung und einen für das Interaktionsmenü zu bilden.

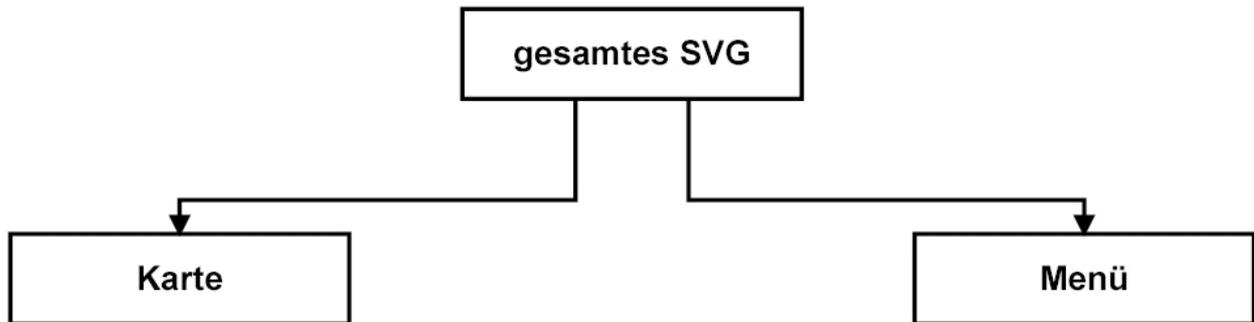


Abbildung 35: Die Aufteilung in Karte und Menü

Der zentrale SVG-Bereich nimmt den ganzen Bildschirmbereich ein und enthält Angaben, die sowohl das Zoomen und Verschieben des Kartenausschnittes über das Adobe Plug-In unterbinden, sowie die Anweisung Skripte in JavaScript nicht vom Browser, sondern vom Plug-In selbst interpretieren zu lassen. Des Weiteren erfolgen hier schon sämtliche Formatdefinitionen für alle Elemente der Karte und des Menüs mittels Cascading Style Sheets. Zusätzlich erfolgen die Festlegungen für die Filter- und Verlaufseffekte, aber auch die Symbole, etwa für die Darstellung der Punkte in der Karte, entsprechend ihrer Punktart, oder für die Schaltflächen des Menüs, werden hier bestimmt.

Die Dimensionen, also Breite und Höhe des Kartenbereiches werden in Zentimeter angegeben, damit später stets der aktuelle Maßstab aus gerade diesen Abmessungen und den Entsprechungen der zugehörigen ViewBox ermittelt werden kann. Die Anfangsparameter der ViewBox werden durch die minimalen und maximalen Koordinatenwerte der Punkteliste bestimmt. Hier ist jedoch zu berücksichtigen, daß bei einem geodätischen Koordinatensystem wie dem hier verwendeten Gauß-Krüger-Koordinatensystem, die vertikale Achse von Süd nach Nord verläuft, während bei den meisten Grafikstandards, und somit auch bei SVG, die Y-Achse von oben nach unten verläuft. Aus diesem Grund mußten diese Werte, und natürlich auch alle späteren Koordinaten, mit einem negativen Vorzeichen versehen werden. Außerdem wurden alle Rechts- und Hochwerte um die jeweiligen Minimalwerte gekürzt. Dies erzeugte zum einen kürzere Angaben innerhalb der SVG-Auszeichnung und somit eine kleinere Dateigröße, was sich insbesondere bei der Datenübertragung im Internet, aber auch bei der Aufbereitungszeit des SVG Viewers positiv auswirkt. Zum anderen schien SVG bei den absoluten, unveränderten GK-Koordinaten an die Grenzen des Auflösungsvermögens zu kommen, so daß es teilweise zu unschönen Grafikfehlern kam. Nachfolgender Vergleich soll dies verdeutlichen:

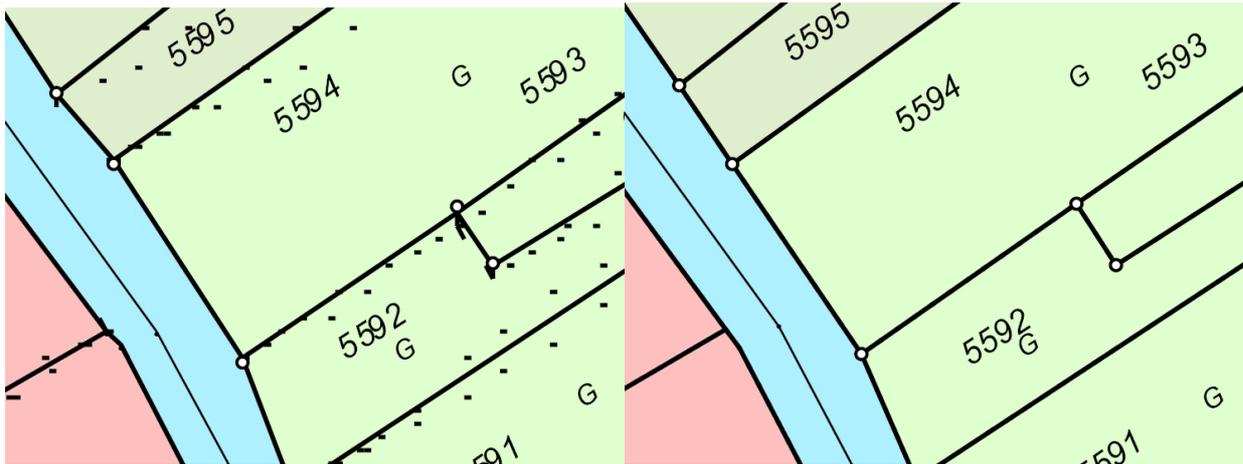


Abbildung 36: Vergleich der Darstellung mit ungekürzten und gekürzten Koordinaten

Unverkennbar werden bei den Texten, in diesem Fall die Flurstücksnummern, die Zeichenabstände nicht gleichmäßig eingehalten, wodurch eine unsaubere Schriftweise entsteht. Bei genauerer Betrachtung kann man auch feststellen, daß die Symbole für die Punkte nicht exakt auf den Knick- und Endpunkten der Grenz- und Topographielinien platziert wurden. Extrem ist auch die Abweichung der gestrichelten Nutzungsartengrenzen von den Flurstücksgrenzen. Sollten diese eigentlich deckungsgleich sein, weichen sie bei ungekürzten Koordinaten so stark voneinander ab, daß die Nutzungsartengrenze als unregelmäßiger „Klötzchenhaufen“ entlang der Flurstücksgrenze abgebildet wird.

In SVG stehen zwar nicht direkt Ebenen zur Verfügung, jedoch lassen sich Elemente innerhalb eines `<g>`-Tags gruppieren, wodurch sich beispielsweise wie in einem GIS, einzelne Elementarten komplett aus- und wieder einblenden lassen. Dadurch, daß sich die Grafikelemente entsprechend ihrer Reihenfolge im SVG-Dokument überlappen und somit verdecken, mußte bei der ebenenweise Ausgabe der Grafik auf eine sinnvolle Reihenfolge geachtet werden:

- Farbflächen für die tatsächlichen Nutzungen
- Farbflächen für die Gebäudenutzungen
- Umringe für die Flächen der tatsächlichen Nutzung
- Umringe für die Flurstücksflächen und die Grenzzüge
- Abkürzungen für die tatsächlichen Nutzungen
- Umringe für die Gebäudeflächen und die Ausgestaltungen
- Topographische Objekte (nach altem Format)
- Lage- und Gemeindepnamen
- Sonstige Texte
- Flurstücksnummern und Bezugspfeile
- Gebäudenutzungen
- Eigennamen der Gebäude
- Hausnummern
- Symbole der trigonometrischen Punkte
- Symbole der Aufnahmepunkte, ohne Schnitt- und Kleinpunkte
- Symbole der Schnitt- und Kleinpunkte
- Symbole der Versicherungspunkte

- Symbole der sonstigen numerierten Punkte
- Symbole der nicht numerierten Punkte
- Nummern der Versicherungspunkte
- Nummern der sonstigen numerierten Punkte
- Nummern der Schnitt- und Kleinpunkte
- Nummern der Aufnahmepunkte, ohne Schnitt- und Kleinpunkte
- Nummern der trigonometrischen Punkte

Das Menü enthält neben den Schaltflächen für das Ein- und Ausblenden der einzelnen Elementgruppen bzw. -ebenen Knöpfe für das Verschieben, sowie das Ein- und Auszoomen (jeweils in drei Stufen) des Bildausschnittes. Außerdem wird hier der aktuelle Maßstab angezeigt, und wenn der Mauszeiger über diesen Bereich geführt wird, klappt ein Untermenü auf mit dem ein neuer Maßstab gewählt werden kann. Ferner sind Schaltflächen für die Wiederherstellung des originalen Bildausschnitts und das Ein- und Ausblenden der Angaben aus den Vordaten enthalten. Ergänzt wird das Menü durch die Angabe des Programmnamens, des Autors und der benutzten Programmversion.



Abbildung 37: Das Menü des BGRUNDviewers

### 5.4.2 Vordaten

Neben der Ausgabe in eine HTML-Datei werden zusätzlich die wichtigsten Angaben aus den Vordaten auch innerhalb des SVG-Dokumentes angegeben. Für beide Aufgaben steht jeweils eine eigene Methode der Klasse Vordaten bereit.

```

Dateiname: BGRUND_Standard.bgr
Antragsnummer: 020054711
Antragsjahr: 2002
Amtsnummer: 5 Testamt
laufende Nummer: 4711
Verwaltungsangaben: 4D 005
Version der Schnittstelle: DIGV050
Abrufdatum: 28.05.02
2 Rahmenpunkte
1. Punkt: Rechtswert: 340399000, Hochwert: 544524000
2. Punkt: Rechtswert: 340419000, Hochwert: 544544000
1 Punktnummernbereich
Nr.: 0: Koordinatensystem: GK, Bezirk: Neustetten, von: 1/000, bis: 19999/699
Zuständige Stelle: Landesvermessungsamt
Datenabgebende Stelle: staatliches Vermessungsamt
Umfang der Datenbereitstellung: Standard: Flurstück wird übergeben, wenn mindestens ein Punkt der Bezugsseite in den Rahmen fällt

```

Abbildung 38: Ausgabe der Vordaten in einer HTML-Datei

Für die Ausgabe in SVG wurden dabei zwei nahezu identische Ebenen angelegt, die beide Male die Angaben zur Antragsnummer, dem Abrufdatum, der datenabgebenden Stelle<sup>115</sup>, dem beteiligten Amt und der verwendeten Schnittstellenversion enthält. Außerdem werden alle Punktnummernbereiche aufgezählt, aus denen für den Auftrag Punktdaten ausgegeben wurden. Beide sind zunächst nicht auf dem Bildschirm sichtbar.

Die erste Version kann über den Knopf mit dem Fragezeichen sichtbar gemacht werden, damit sich der Benutzer die Informationen aus den Vordaten am Bildschirm anzeigen lassen kann. Durch einen abermaligen Klick auf das Feld mit dem Fragezeichen oder auf das Feld mit dem Ausrufezeichen innerhalb des Vordatenrahmens wird diese Ebene wieder unsichtbar gemacht und ermöglicht somit erneut den Zugang zu den Schaltflächen für das Ein- und Ausblenden der einzelnen Kartenebenen.

Um bei einem Ausdruck stets die Vordaten mit auszugeben wurde die zweite Ebene mit den Vordaten angelegt. Durch eine spezielle Klassendefinition in CSS wird diese Ebene nie am Bildschirm ausgegeben, jedoch immer bei der Ausgabe auf einem Drucker oder ähnlichem.

Ursprünglich war geplant, die erste Variante durch JavaScript vor einem Druckauftrag sichtbar zu machen, aber zur Zeit gibt es nur beim Internet Explorer ab Version 5 einen Event-Handler<sup>116</sup>, der aufgerufen wird bevor das Dokument ausgedruckt wird. Alternativ dazu wäre auch eine neue Schaltfläche vorstellbar gewesen, über die der Druck gestartet wird nachdem zuvor weitere Anweisungen ausgeführt wurden. Jedoch scheint die Script-Engine des Plug-Ins noch nicht die print-Anweisung in JavaScript verarbeiten zu können, wodurch auch dieser Lösungsansatz nicht durchgeführt werden konnte.

<sup>115</sup> Landesvermessungsamt oder staatliches Vermessungsamt

<sup>116</sup> onbeforeprint; das Gegenstück dazu wäre onafterprint

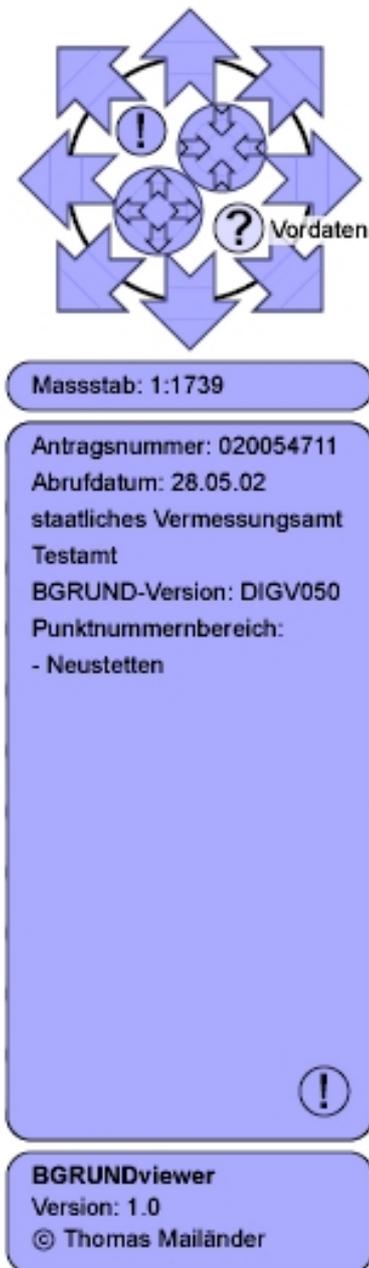


Abbildung 39: Die Ausgabe der Vordaten im Menü

### 5.4.3 Punkte

Bevor die Punkte in die HTML-Datei geschrieben werden, werden alle Punktlisten (TP, AP, SK, VP, sonstige numerierte Punkte und nicht numerierte Punkte) sortiert, damit sich die Reihenfolge nicht mehr aus dem aufsteigenden Rechtswert, sondern aus der Punktnummer ergibt.

```

Schlüssel des Punktnummerierungsbezirkes: 0
Leitpunktnummer: 39
Folgepunktnummer: 72
Punktart: keine Punktart gespeichert oder Grenz-/Kreismittelpunkt
Rechtswert (in Zentimeter): 340406252
Hochwert (in Zentimeter): 544527132
Punktart Grenzpunkt oder Kreismittelpunkt: weder Grenz- noch Kreismittelpunkt
Punktkenennung aus gegebenen Grundrißdaten: Gebäudepunkt
Punktkenennung aus gegebenen Grundrißdaten: kein Punkt eines topographischen Objekts
Lagestatus: endgültige Gauß-Krüger-Koordinaten
Punktstatus: gültiger Punkt
Quellenhinweis: Dauerriß ist zu beachten
Hinweis bei nummerierten Grenzpunkten: Punkt liegt nicht in Gerade
Kennung für Kreismittelpunkt: kein Kreismittelpunkt
Gemarkung: Neustetten
Art der Abmarkung und Vermarkung: unabgemarkt oder unvermarkt
Höhe oder Tiefe der Ab-/Vermarkung auf dm: Höhe oder Tiefe nicht bekannt
Aktenhinweis (ohne Gemarkungsnummer): 1988
Zuständige Stelle: Landesvermessungsamt
Datum der Speicherung bzw. der letzten Fortführung: 29.01.1996

```

Abbildung 40: Auszug aus der Punkteliste

Um in der SVG-Grafik den Punkten das richtige Symbol zuzuordnen, wird der jeweilige Punktstatus herangezogen. In Anlehnung an die Katasterkartenvorschrift wurde nämlich folgende Einteilung der Punkte erforderlich:

- Trigonometrische Punkte
- Aufnahmepunkte, sowie Schnitt- und Kleinpunkte
- Versicherungspunkte und sonstige numerierte Punkte
- Nicht numerierte Punkte

Da TPs in der Regel vermarkt sein müssen, kann diesen gleich das entsprechende Symbol in Form eines Dreiecks zugewiesen werden.

Bei den anderen Punkte muß allerdings zuerst überprüft werden, ob diese in der Örtlichkeit vermarkt bzw. abgemarkt sind, da abhängig davon das entsprechende Symbol benutzt werden muß, um dem Kartenbetrachter einen detaillierteren Überblick über die Situation vor Ort zu verschaffen.

Beispiel für die Beschreibung eines TPs in der erzeugten SVG-Datei:

```

<a xlink:href="punkte.htm#8002/114/0" target="pListe"><use
id="8002/114/0" xlink:href="#symbolTP" x="17661" y="-5767"/></a>

```

Das ermittelte Symbol wird über das `<use>`-Tag und die darin enthaltene Angabe `xlink:href="#symbolTP"` in der Grafik referenziert. Die um die Minimalwerte der Punkteliste gekürzten Koordinaten plazieren das Symbol an der gewünschten Position.

Jedes Punktsymbol wird mit einem Link versehen, damit sich bei einem Klick auf den Punkt in der Karte ein neues Fenster des Internet-Browsers öffnet. In dem neuen Fenster wird die HTML-Datei mit den Punktinformationen angezeigt, und über einen gesetzten Anker scrollt der Browser automatisch zum angeklickten Punkt. Die Schlüsselbezeichnung für die Verknüpfung zwischen Link und Browser besteht aus:

- Bezeichnung der TK25 / TP-Nummer / Unternummer bei Trigonometrischen Punkten
- Rechtswert / Hochwert bei nicht numerierten Punkten

- Schlüssel des Punktnummerungsbezirks / Leitpunktnummer / Folgepunktnummer bei allen anderen Punkten

Im späteren Verlauf des Programmes werden die Punktnummern aller numerierter Punkte in die SVG-Datei ausgegeben. Zwar sind die Ebenen, in denen die dafür benötigten Textelemente aufbewahrt werden zunächst unsichtbar, über die Schaltflächen des Menüs lassen sie sich jedoch nach Bedarf sichtbar machen, und die Punktnummern werden somit angezeigt.



Abbildung 41: Punkte ohne und mit Punktnummer

Die Positionen der einzelnen Punktnummern ergeben sich aus den Koordinaten des zugehörigen Punktes und einem kleinen Koordinatenzuschlag, wodurch die Nummern stets etwas rechts von dem Punkt platziert werden.

### 5.4.4 Flurstücke und Grenzzüge

In der HTML-Datei werden für die Flurstücke neben Angaben wie der Nummer, der Gemarkung und der Flur auch die Punkte und Kreisbögen des Umringes beschrieben. Ferner werden die enthaltenen Flächen der tatsächlichen Nutzung aufgezählt, sofern Informationen dazu vorliegen.

<p>Flurstücksnummer: <b>42/000</b>                  Gemarkung/Flur: 9994/000, Neustetten                  Punkt: 0/39/10 Punkt: 0/430/1 Punkt: 0/430/5 Punkt: 0/1838/55                  Punkt: 0/430/6 Punkt: 0/430/0 Punkt: 0/430/2 Punkt: 0/39/25                  Punkt: 0/39/24 Punkt: 0/39/10                  Tatsächliche Nutzung: GFLF: <b>Gebäude- und Freifläche Land- und Forstwirtschaft</b></p>
<p>Flurstücksnummer: <b>46/000</b>                  Gemarkung/Flur: 9994/000, Neustetten                  Punkt: 0/39/12 Punkt: 0/39/17 Punkt: 0/39/18 Punkt: 0/39/11                  Punkt: 0/39/25 Punkt: 0/430/2 Punkt: 0/430/0 Punkt: 0/430/3                  Punkt: 340410092, 544534901 Punkt: 340409777, 544534014 Punkt: 340410423, 544532626 Punkt: 340409028, 544531960                  Punkt: 340408326, 544531414 Punkt: 0/39/28 Punkt: 0/39/27 Punkt: 0/39/26                  Punkt: 0/39/12                  Tatsächliche Nutzung: GFW: <b>Gebäude- und Freifläche Wohnen</b>                  Tatsächliche Nutzung: G: <b>Gartenland</b></p>

Abbildung 42: Auszug aus der Flurstücksliste

Bei den Grenzzügen werden nur die Art des Grenzzuges, der Anfangs- und Endpunkt, wie auch ein eventuell vorhandener Kreisbogen erwähnt.

Das Zeichnen des Flurstückumringes geschieht größtenteils durch eine Methode der Elternklasse Linienzug. Diese wurde so angelegt, daß sie sämtliche Flächenkonturen der Klassen Flurstück und Nutzungsfläche als auch die Linien eines Grenzzuges oder eines topographischen Objektes graphisch darstellen kann. Aber nicht nur die Konturen von Flächen lassen sich damit erzeugen, die Flächen selbst lassen sich ebenso über diese Methode einfärben. Was letztendlich gezeichnet wird, wird durch die Angabe einer CSS-Klasse gesteuert.

Erstellt werden dabei Pfade entsprechend des `<path>`-Tags von SVG. Die Methode durchläuft den Array der beschreibenden Punkte und Kreisbögen und setzt entsprechend den Einträgen der Liste die Steuerzeichen, Koordinaten und Radien.

Das Steuerzeichen für den ersten Punkt der Pfadbeschreibung ist zwangsläufig das „M“ damit der Anfangspunkt des Pfades gesetzt werden kann.

Bei allen weiteren Punkten wird überprüft, ob es sich eventuell um einen Kreisbogen handelt. Ist dies der Fall, wird das Steuerzeichen „A“ für einen Elliptical Arc gesetzt. Da in BGRUND nur Kreisbögen und keine Ellipsenbögen vorhanden sind, werden beide Ellipsenradien des Elliptical Arcs auf den vorgegebenen Radius des Kreisbogens gesetzt. Eine weitere Vereinfachung infolge der Tatsache, daß stets Kreisbögen und nicht Ellipsenbögen zu zeichnen sind, besteht darin, daß für die X-Achsen-Rotation immer „0“ angenommen werden kann, da sich eine Rotation bei einem Kreis nicht auswirkt. Auch für die Auswahl des Kreisbogens wird durchgehend „0“ festgelegt, da davon ausgegangen wird, daß jeweils der kleinere Bogenschnitt gemeint ist. Nur hinsichtlich der Richtung des Ellipsenbogens müssen die Informationen aus der BGRUND-Datei konsultiert werden. Sagen diese aus, daß der Kreismittelpunkt links von der Sehne liegt, wird der Parameter auf „0“, im anderen Fall auf „1“ gesetzt. Ansonsten mußte nur beachtet werden, daß der nachfolgende Punkt, das heißt seine gekürzten Koordinaten, ohne vorstehendes Steuerzeichen angeführt wird.

Ergänzend sollte hier angeführt werden, daß vor der Realisierung über Elliptische Kurven versucht wurde, die Kreisbögen mittels kubischer Bézierkurven zu definieren. Davon ist allerdings im nachhinein abzuratet. Nicht nur, daß das Programm durch die Berechnung des Anfasserpunktes aufwendiger und damit langsamer wird, es können prinzipiell keine korrekten Kreisbögen über Bézierkurven erzeugt werden. Je näher der Kreismittelpunkt an die Sehne rückt, desto paralleler verlaufen die Tangenten an den Kreisbogen durch die Bogenendpunkte. Als Resultat daraus wird der Bogen so sehr zum Anfasserpunkt gezogen, daß der Kreisbogen enorm verzerrt wirkt.

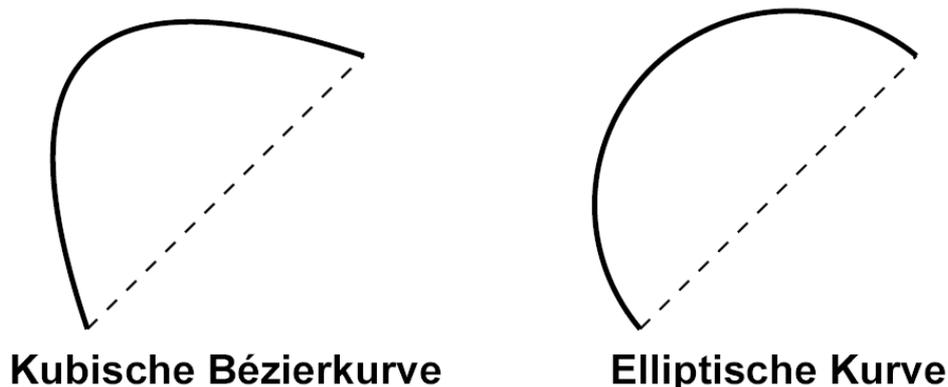


Abbildung 43: Vergleich kubische Bézierkurve mit elliptischer Kurve

Auszug aus der Beschreibung einer Flurstücksfläche in der erzeugten SVG-Datei:

```
<path id="9994/0/22/0" class="flst" d="M5627,-15570L6137,-15970 ...
L12051,-7288A1417 1417 0 0 0 12508,-6933 ... L5627,-15570"/>
```

Liegt bei dem aktuellen Element kein Kreisbogen vor, wird das Steuerzeichen „L“ für eine gerade Linie zum nächsten Punkt ausgegeben.

Nicht numerierte Punkte werden in dem beschreibenden Array mit Rechts- und Hochwert geführt, die Methode zur Zusammensetzung des Pfades muß somit diese nur um die Minimalwerte der gesamten Punkteliste kürzen und in den Pfad einfügen.

Bei numerierten Punkten wird eine weitere Funktion aufgerufen, um die dazugehörenden Koordinaten zu ermitteln. Während bei trigonometrischen Punkten aufgrund der geringen Anzahl die Liste der TPs so lange durchlaufen werden kann bis der entsprechende Punkt gefunden wird, würde dies bei allen anderen Punkten zu lange dauern. Bei einer BGRUND-Datei mit 650KB Größe können immerhin über 3000 Punkte anfallen. Hier zeigt sich der Nachteil von PHP, da es für diese großen Datenmengen aufgrund seines zur Laufzeit interpretierten Aufbaus zu langsam ist.

Als Folge müßte man in der Konfigurationsdatei php.ini die Zeitspanne erhöhen die einem Skript zur Verfügung steht, bevor es abgebrochen wird. Dies würde jedoch auch eventuelle schädliche Skripte begünstigen und ist somit aus sicherheitsrelevanten Aspekten nicht ratsam. Ferner soll der Anwender nicht durch unverhältnismäßig lange Wartezeiten gequält werden, und auch der Server würde auf diese Weise übermäßig belastet werden.

Als Lösung wurde zuerst ein bestimmter Suchalgorithmus erdacht, der den Punkt in der Mitte der sortierten Punkteliste mit dem gesuchten Punkt vergleicht und dadurch erkennt, ob sich der gesuchte Punkt in der ersten oder zweiten Hälfte der Punkteliste befindet. Dann wird ein temporärer Array gebildet, der die Hälfte der Punkteliste mit dem gesuchten Punkt aufnimmt. Diese Prozedur wird so lange wiederholt bis der Vergleichspunkt dem gesuchten Punkt entspricht. Aber auch diese Suchmethode erbrachte nicht die gewünschte Leistungssteigerung.

Aus diesem Grund wurde ein MySQL-Datenbanksystem eingesetzt. Bei jeder Verarbeitung einer BGRUND-Datei wird eine neue Tabelle in der Datenbank angelegt, in der alle numerierten Punkte (mit Ausnahme der TPs) des Umsetzungsvorganges mit der Schlüsselbezeichnung aus Punktnumerierungsbe-

zirk, Leit- und Folgepunktnummer gespeichert werden. Ansonsten werden nur die Koordinaten der Punkte in die Tabelle aufgenommen. Damit wird der Zugriff auf die entsprechenden Koordinaten enorm gesteigert, wodurch die Zeitspanne des gesamten Bearbeitungsprozesses auf etwa ein Drittel gesenkt werden konnte.

Nachdem eine adäquate Methode entwickelt war um die Linienzüge auszugeben, konnte sie verwendet werden um die Flurstücksgrenzen und Grenzzüge in das SVG-Dokument auszugeben.



Abbildung 44: Die Darstellung der Flurstücksgrenzen und der Grenzzüge

Durch die Parameter für die Flurstücksnummer konnte die Darstellung auf viererlei Arten erfolgen:

- Keine Darstellung in der Karte
- Gewöhnliche Darstellung, gegebenenfalls mit Schrägstrich zur Unterteilung in Stamm- und Unternummer
- Darstellung mit Bruchstrich. Dabei steht die Stammnummer oben und die Unternummer unten. Dies wird dadurch erreicht, daß die Stammnummer per CSS mit einem Unterstrich dekoriert und die Unternummer mit einem Zuschlag auf den Y-Wert versehen wird
- Nur Darstellung der Unternummer mit vorangestelltem Schrägstrich

Durch die angegebenen Koordinaten und Drehwinkel in der BGRUND-Datei konnten die Flurstücksnummern recht einfach in der Karte positioniert werden.

Bei kleinen Flurstücken kann es aber vorkommen, daß die Flurstücksnummer aus Platzgründen nicht mehr innerhalb des zugehörigen Flurstücks plaziert werden kann. Deswegen wird in solchen Fällen ein zusätzlicher Pfeil gezeichnet, der eine Zuordnung der Nummer zum Flurstück erleichtert.



Abbildung 45: Die Darstellung der Flurstücksnummern und der Zuordnungspfeile

Jede Flurstücksnummer in der Karte wird mit einem Link hinterlegt, so daß sich der Betrachter durch einen Klick auf die Flurstücksnummer die Informationen des angehörigen Flurstückes über die HTML-Datei anzeigen lassen kann.

#### 5.4.5 Flächen der tatsächlichen Nutzung

Noch bevor die Flurstückskonturen in die SVG-Datei ausgegeben werden, erfolgt die farbige Anlegung und das Zeichnen der Umringe der tatsächlichen Nutzungsflächen. Für die Farbuordnung, abhängig von der vorliegenden Nutzungsart, wurden die Farbvorgaben der staatlichen Vermessungsverwaltung berücksichtigt.

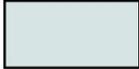
	<b>Gebäude- und Freifläche</b> RGB-Farbwert: 251 185 186
	<b>Betriebsfläche, Erholungsfläche, Fläche anderer Nutzung</b> RGB-Farbwert: 214 228 228
	<b>Verkehrsfläche</b> RGB-Farbwert: 225 225 137
	<b>Landwirtschaftsfläche</b> RGB-Farbwert: 221 236 202
	<b>Waldfläche</b> RGB-Farbwert: 097 235 097
	<b>Wasserfläche</b> RGB-Farbwert: 153 240 255

Abbildung 46: Die Farbvorgaben der tatsächlichen Nutzungsflächen

Bei dem Programmsystem BGRUNDviewer wurden die Farbgruppen jedoch noch weiter unterteilt, so daß beispielsweise die Straßenflächen mit einem leicht abweichenden Farbton von dem der Wegflächen dargestellt werden. Weiße Flächen weisen hingegen darauf hin, daß das entsprechende Flurstück bzw. die Nutzungsfläche nicht komplett im Abrufrahmen enthalten war oder, daß ganz einfach keine Informationen zur tatsächlichen Nutzung vorliegen.

Die Grenzen der tatsächlichen Nutzungsflächen werden durch eine gestrichelte Linie angezeigt. Durch die Kombination der unterschiedlichen Farben und der Nutzungsartengrenzen erhält der Betrachter der Karte eine recht deutliche Vorstellung über die örtlichen Verhältnisse. Entsprechend der Katasterkartenvorschrift werden zusätzlich die Abkürzungen für die tatsächlichen Nutzungen ausgegeben, sofern in der BGRUND-Datei keine Angaben zur Unterdrückung der jeweiligen Darstellung vorhanden sind.

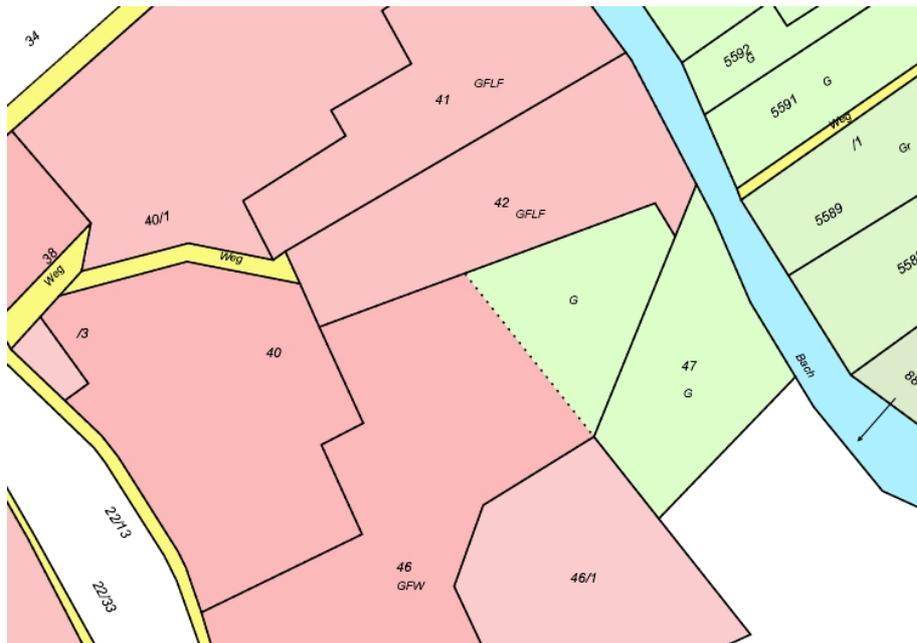


Abbildung 47: Die Darstellung der tatsächlichen Nutzungsflächen

#### 5.4.6 Gebäude

Für die Gebäude werden in der HTML-Datei die Nutzung, Gemarkung und Flur sowie die Nummer des Flurstücks auf dem das Gebäude liegt, ausgegeben. Ferner werden durch die Methode der Elternklasse Linienzug die Punkte und Kreisbögen aufgelistet, durch die die Form des Gebäudes definiert wird. Darauf folgen Angaben zur Genauigkeitsstufe des Gebäudes. Für den Fall, daß das Gebäude eine Hausnummer oder gar einen Eigennamen enthält, werden auch diese Angaben angezeigt. Dasselbe gilt für die eventuell vorhandenen Ausgestaltungen, wie etwa Begrenzungen von unterirdischen Gebäuden oder Arkaden.

<p>Gebäudenutzung: Schu: <b>Schuppen</b>  Gemarkung/Flur: 9994/000, Neustetten  Flurstück: 59/009  Punkt: 340413787, 544526711 Punkt: 340414147, 544526868 Punkt: 0/2660/102 Punkt: 340413972, 544526348  Punkt: 340413787, 544526711  Genauigkeitsstufe des Gebäudes: digitalisiert aus Maßstab m &gt;= 1:500</p>
<p>Gebäudenutzung: <b>Kirche</b>  Gemarkung/Flur: 9994/000, Neustetten  Flurstück: 59/009  Punkt: 340414147, 544526868 Punkt: 0/2660/101 Punkt: 340414154, 544528922 Punkt: 340414490, 544528297  Punkt: 340415035, 544528572 Punkt: 340415359, 544527918 Punkt: 340414842, 544527645 Punkt: 340415206, 544526968  Punkt: 0/2660/102 Punkt: 340414147, 544526868  Genauigkeitsstufe des Gebäudes: digitalisiert aus Maßstab m &gt;= 1:500  Eigenname: Veitkirche  Eigenname: Veitkirche  Hausnummer: 2 Darstellung: gewöhnliche Darstellung (ggf. mit Schrägstrich)  Hausnummer: 2 Darstellung: gewöhnliche Darstellung (ggf. mit Schrägstrich)</p>
<p>Gebäudenutzung: <b>Rathaus</b>  Gemarkung/Flur: 9994/000, Neustetten  Flurstück: 135/000  Punkt: 0/97/22 Punkt: 340408660, 544525847 Punkt: 340408608, 544526045 Punkt: 340408753, 544526163  Punkt: 0/97/19 Punkt: 0/97/18 Punkt: 0/97/17 Punkt: 0/97/20  Punkt: 340409527, 544526436 Punkt: 0/97/21 Punkt: 0/97/22  Genauigkeitsstufe des Gebäudes: digitalisiert aus Maßstab m &gt;= 1:500  Hausnummer: 2 Darstellung: gewöhnliche Darstellung (ggf. mit Schrägstrich)  Hausnummer: 2 Darstellung: gewöhnliche Darstellung (ggf. mit Schrägstrich)</p>

Abbildung 48: Auszug aus der Gebäudeliste

Die Ausgabe der Gebäude in das SVG-Dokument verläuft in mehreren Schritten. Die erste Ebene mit Informationen zu den Gebäuden beinhaltet die Farbflächen. In den amtlichen Kartenwerken werden die Gebäude abhängig von ihrer Nutzung mit einer speziell festgelegten Schraffurart versehen. So werden beispielsweise Nebengebäude parallel zur vorherrschenden Gebäudeseite schraffiert, während bei Hauptgebäuden ein 45°-Winkel angestrebt und bei öffentlichen Gebäuden eine dickere Schraffurlinie verwendet wird.

Diese Art der graphischen Darstellung wäre auch für den BGRUNDviewer interessant gewesen, vor allem, weil die Karten auch in einem Schwarz-Weiß-Modus angezeigt werden können. Jedoch erwies sich SVG für Schraffuren nicht sehr geeignet. Über Füllmuster, den sogenannten Patterns, lassen sich zwar Schraffuren erstellen, und diese können zwar auch über eine eigene Transformation gedreht werden, aber eben nur zentral für das ganze Dokument. Dadurch ist es nicht möglich, die einzelnen Schraffuren entsprechend der vorherrschenden Gebäudeseite auszurichten. Weitere Probleme ergäben sich bei der Verwendung von Schraffuren dadurch, daß die Hausnummern, Nutzungen und Eigennamen der Gebäude von der Schraffur freigestellt werden müßten, um weiterhin deutlich erkennbar zu bleiben. Außerdem litt die Performance des SVG Viewers sehr stark bei der Benutzung von Schraffuren, was sich beispielsweise beim Verschieben und Zoomen des Bildausschnittes deutlich zeigte. Dies leuchtet ein, muß das Plug-In schließlich sämtliche Linienelemente der Schraffur dabei jedesmal neu berechnen.



Abbildung 49: Der Versuch Gebäude zu schraffieren

Aus all diesen Gründen wurde auf eine Ausgabe mit Schraffur verzichtet, eine farbliche Darstellung entspricht sowieso, gerade im Bereich des Internets, eher der modernen Auffassung einer Kartendarstellung. Als Überbleibsel der traditionellen Anzeige mit Schraffuren wurde der jeweilige Farbwert für die Gebäudefläche in Abhängigkeit vom angegebenen Schraffurstil vergeben. Dadurch werden öffentliche Gebäude in einem dunkleren Rotton wiedergegeben als bei normalen Hauptgebäuden der Fall ist. Nebengebäude werden hingegen in einem helleren Rot dargestellt.



Abbildung 50: Die Darstellung der Gebäude inklusive Ausgestaltungen

Um die Umrisse der Gebäude zu zeichnen, konnte ausnahmsweise nicht auf die Methode der Elternklasse Linienzug zurückgegriffen werden. Müssen nämlich einzelne Gebäudeseiten durch eine aufgeführte Ausgestaltung in einem besonderen Stil ausgegeben werden, muß das Programm wissen, daß eben diese Seite nicht auf herkömmliche Weise gezeichnet werden darf. Durch die durchgezogene Linie einer herkömmlichen Seite wäre die gestrichelte Linie einer besonderen Ausgestaltung nicht mehr sichtbar, und die Information ginge für den Betrachter verloren.

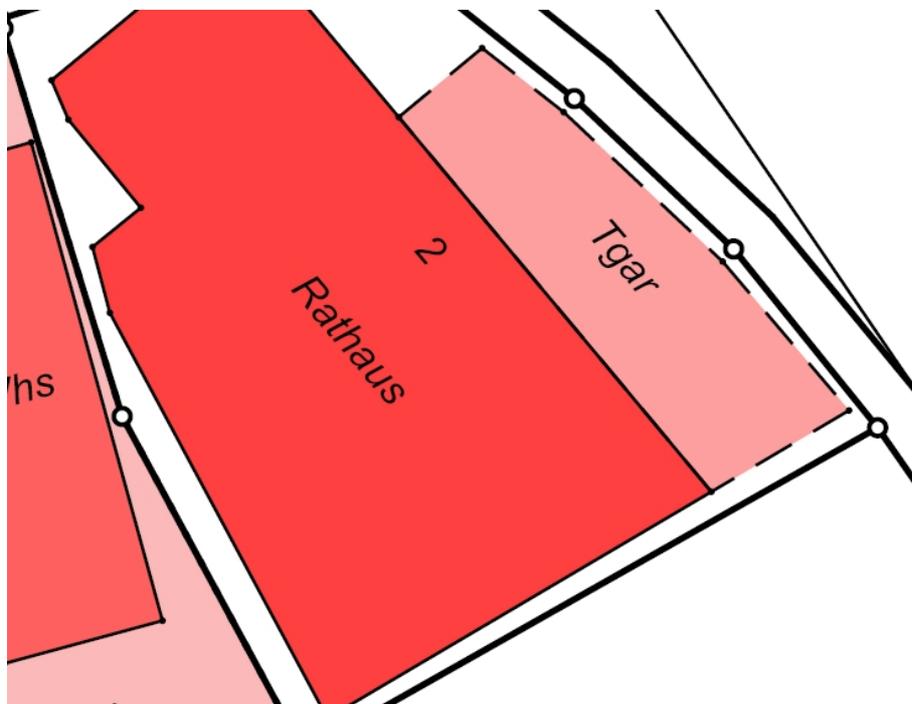


Abbildung 51: Die Darstellung von Ausgestaltungen bei Gebäuden

Anhand der Abbildung sieht man, daß die Gebäudeseiten der Tiefgarage nicht mit dem herkömmlichen, durchgezogenen Strich, sondern mit der besonderen, gestrichelten Variante dargestellt werden. Für die Ausgabe der Ausgestaltungen steht der Klasse Gebäude eine eigene Methode zur Verfügung.

Weitere Methoden gibt es für die Präsentation der Hausnummern, Gebäudenutzungen und Eigennamen. Für die Eigennamen und Hausnummern können unterschiedliche Werte für die Darstellung und Positionierung für die Maßstabsbereiche 1:1 bis 1:1000 und 1:1001 bis 1:2500 angegeben sein.

Außerdem wird in den BGRUND-Daten ein Steuerzeichen geführt, das für die Hausnummer die Darstellungsart festlegt. Dabei sind die gleichen Variationen möglich wie bei der Flurstücksnummer.

Die Gebäudenutzungen ermöglichen durch einen hinterlegten Link einen direkten Sprung zum betroffenen Gebäude in der HTML-Datei, wodurch sich der Anwender schnell zusätzliche Informationen zum Gebäude besorgen kann. Soll für das Gebäude keine Nutzung in der Karte eingetragen werden, wird statt dessen das Zeichen „>“ eingetragen, um weiterhin einen raschen Zugriff auf das HTML-Dokument zu ermöglichen.

### 5.4.7 Topographische Objekte

Da während der Entwicklungsphase des Programms nicht ausreichend Testdaten mit topographischen Objekten gemäß des kommenden Formats zur Verfügung standen, beschränkt sich die Funktionalität des BGRUNDviewers momentan noch auf die topographischen Objekte entsprechend des alten Formats. Teilweise wurde das Programm jedoch schon so angelegt, daß sich später mit relativ wenig Aufwand die zukünftigen punkt-, linien- und flächenförmigen Objekte integrieren lassen.

<b>Art: überdeckte topographische Linie</b> Datum der Objektbildung: 04.03.1996 Punkt: 340407600, 544533368 Punkt: 340407923, 544533853
<b>Art: topographische Linie</b> Datum der Objektbildung: 04.03.1996 Punkt: 340407923, 544533853 Punkt: 340408051, 544533583
<b>Art: deutliche Böschungunterkante</b> Datum der Objektbildung: 23.02.1996 Punkt: 340409260, 544543173 Punkt: 0/1838/21
<b>Art: topographische Linie</b> Datum der Objektbildung: 04.03.1996 Punkt: 340410209, 544526895 Kreis: L 658cm Punkt: 0/19027/21
<b>Art: offene Gebäudebegrenzungslinie</b> Datum der Objektbildung: 04.03.1996 Punkt: 340412926, 544526273 Punkt: 340413110, 544525908

Abbildung 52: Auszug aus der Liste mit den topographischen Objekten

Der Eintrag der topographischen Objekte in die SVG-Datei wird auch hier wieder größtenteils von der Klasse Linienzug übernommen. Der jeweils dazu verwendete Linienstil wird durch die Art des topographischen Objekts bestimmt, also abhängig davon, ob etwa eine Böschungskante, eine offene Gebäudebegrenzungslinie, ein Fahrbahnrand oder eine Nutzungsgrenze abgebildet werden soll.



Abbildung 53: Die Darstellung der topographischen Objekte

Klickt man auf ein topographisches Objekt in der Karte, öffnet sich die HTML-Datei mit den zugehörigen Informationen, und die Seite scrollt zum entsprechenden Eintrag.

### 5.4.8 Texte

Texte ergänzen die Karte um Gemeinde-, Ortsteil-, Gewann-, Straßen- und Gewässernamen. Diese werden von der Klasse Lage- und Gemeinename berücksichtigt. Es werden aber auch Texte für die Klassifizierungen von Straßen oder Gewässern ausgegeben. Ferner können für nicht komplett abgerufene Flurstücke bzw. Nutzungsflächen die Flurstücksnummern bzw. Abkürzungen der tatsächlichen Nutzung als Texte im SVG-Dokument berücksichtigt werden. Für diese Aufgaben und alle anderen Textarten ist die Klasse sonstiger Text zuständig.

<p>Text: <b>Raiffeisenstraße</b>            Art: Straßenname, rechtsgeneigte Schrift 2,5mm            Position: Rechtswert: 340412540, Hochwert: 544524130, Drehung: 59.4°            Speicherungsdatum: 11.03.2002            Gemeindegeschlüssel: 885070            Lageschlüssel: 88845            Lage- bzw. Gemeinename: Raiffeisenstraße</p>
<p>12 SonstigeTexte   Lage- bzw. Gemeinennamen</p>
<p>Text: <b>Üb</b>            Art: tatsächliche Nutzung, wenn noch keine flächenförmigen Objekte gebildet sind            Position: Rechtswert: 340399930, Hochwert: 544540980, Drehung: 0°            Speicherungsdatum: 25.11.1998</p>
<p>Text: <b>Fluß</b>            Art: tatsächliche Nutzung, wenn noch keine flächenförmigen Objekte gebildet sind            Position: Rechtswert: 340400720, Hochwert: 544541690, Drehung: 0°            Speicherungsdatum: 25.11.1998</p>

Abbildung 54: Auszug aus der Texteliste

Den Texten ist in der BGRUND-Datei ein Steuerzeichen vorangestellt, das angibt bei welchem Maßstabsbereich<sup>117</sup> sie jeweilig anzuzeigen sind. Da jedoch in SVG die Texte ebenfalls skaliert werden, reicht es vorerst, nur die Texte aus dem Maßstabsbereich 1:1 bis 1:1000 darzustellen, um ein befriedigendes Ergebnis zu erzielen. Langfristig sollte jedoch angestrebt werden für jeden Maßstabsbereich eigene Ebenen anzulegen, die dann, je nach aktuellem Abbildungsmaßstab der Karte, ein- bzw. ausgeblendet werden.

<sup>117</sup> 1:1 bis 1:1000 oder 1:1001 bis 1:2500

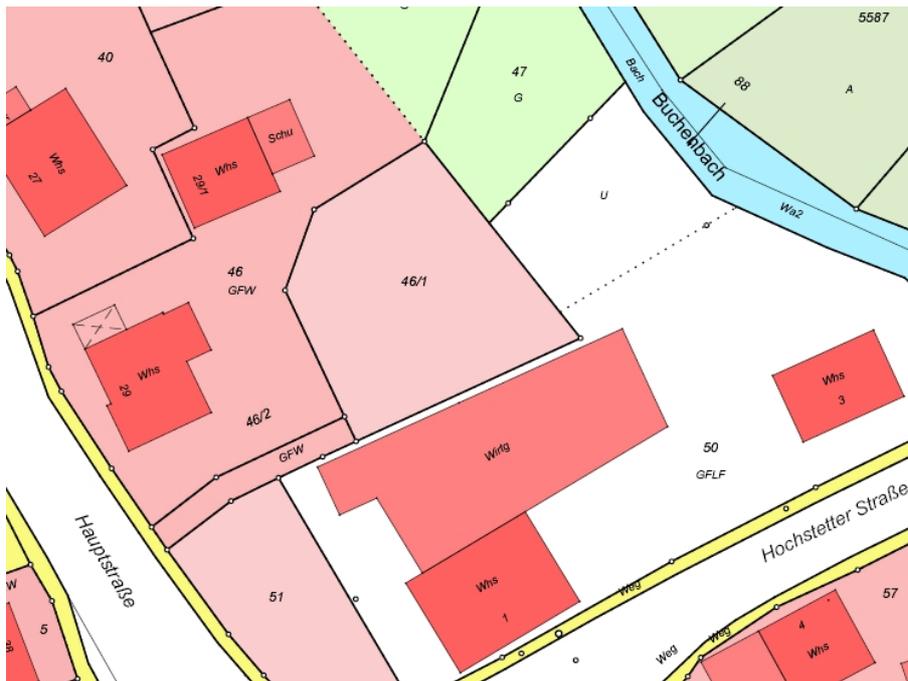


Abbildung 55: Die Darstellung der Lage- und Gemeindegrenzen sowie sonstiger Texte

## 5.5 Das Menü des BGRUNDviewers

### 5.5.1 Aktionen beim Aufruf der Karte

Durch den Event-Handler `onload` wird beim Öffnen des erstellten SVG-Dokuments eine JavaScript-Funktion gestartet. Diese Funktion speichert die ursprünglichen Parameter der `ViewBox`<sup>118</sup> als globale Variablen und berechnet den Maßstab aus den Abmessungen des Kartenbereiches und der `ViewBox`. Anschließend wird der berechnete Maßstab in dem dafür vorgesehenen Feld des Menüs eingetragen.

### 5.5.2 Verschieben des Kartenausschnittes

Das Menü des BGRUNDviewers erlaubt das Verschieben des Kartenausschnittes in Nord-, Nordost-, Ost-, Südost-, Süd-, Südwest-, West- und Nordwest-Richtung. Dabei kann unter drei unterschiedlichen Verschiebungsbeträgen ausgewählt werden. Je nachdem über welchem Bereich des Richtungspfeiles geklickt wird, beträgt der Verschiebungsbetrag 10%, 50% oder 90% des Kartenausschnittes. Um dem Anwender die einzelnen Sektionen des Pfeiles optisch zu verdeutlichen, sind diese zum einen durch Linien voneinander getrennt, zum anderen färben sich die Pfeile beim Überfahren mit dem Mauszeiger unterschiedlich ein. Dabei gilt: je heller der Pfeil wird, desto weiter würde sich der Kartenausschnitt bei einem Klick verschieben.

<sup>118</sup> Koordinaten für die linke obere Ecke sowie Breite und Höhe

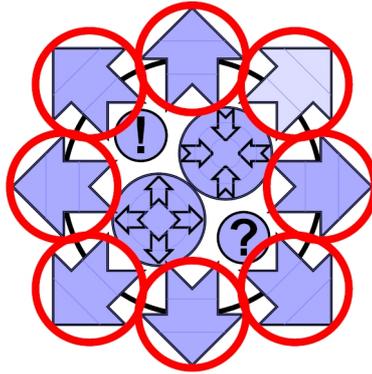


Abbildung 56: Die Schaltflächen für das Verschieben des Kartenausschnittes

### 5.5.3 Zoomen des Kartenausschnittes

Das Verändern des Abbildungsmaßstabes kann beim BGRUNDviewer auf zweierlei Weise erfolgen. Die erste Möglichkeit ist, die entsprechende Schaltfläche für das Ein- bzw. Auszoomen anzuklicken. Dabei gibt es, analog zu den Schaltflächen für das Verschieben, drei mögliche Stufen. Zur Verfügung stehen Zoomstufen von 75%, 50% und 25%, das heißt bei einem Ausgangsmaßstab 1:1000 könnte man entweder auf die Maßstäbe 1:750, 1:500 oder 1:250 hinein zoomen oder aber auf die Maßstäbe 1:1333, 1:200 oder 1:4000 heraus zoomen.

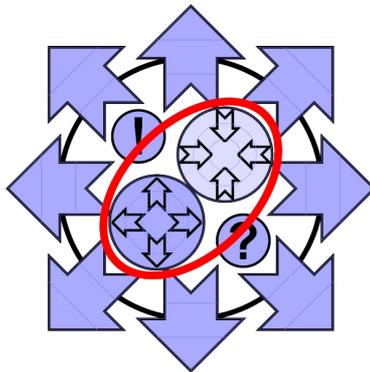


Abbildung 57: Die Schaltflächen für das Zoomen des Kartenausschnittes

Die zweite Möglichkeit für das Verändern des Abbildungsmaßstabes besteht über das Popup-Menü, das erscheint sobald der Mauszeiger über die Anzeigefläche des aktuellen Maßstabes geführt wird. Ausgewählt werden können die Maßstäbe 1:100, 1:250, 1:333, 1:500, 1:1000, 1:1500, 1:2500 und 1:5000.



Abbildung 58: Das Popup-Menü für die Auswahl des gewünschten Maßstabs

Unabhängig davon wie der Zoomvorgang ausgelöst wird, werden die Parameter für die ViewBox entsprechend neu gesetzt, der neue Maßstab aus den neuen Werten berechnet und in dem Anzeigefeld des Menüs aktualisiert.

#### 5.5.4 Wiederherstellung des originalen Bildausschnittes

Durch einen Klick auf die Schaltfläche mit dem Ausrufezeichen setzt eine Funktion in JavaScript die Parameter für die ViewBox auf die Werte, die beim Aufrufen der Karte mit den ursprünglichen Angaben der ViewBox als globale Variablen gespeichert wurden. Dadurch wird genau der Kartenausschnitt angezeigt, der auch beim Öffnen des SVG-Dokumentes abgebildet wurde, und das gesamte Gebiet wird sichtbar.

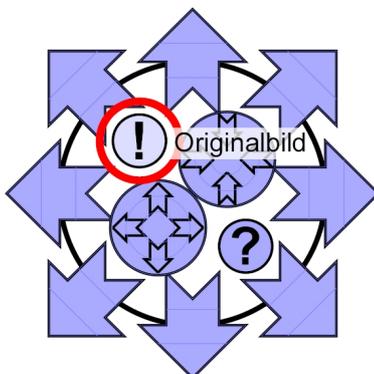


Abbildung 59: Die Schaltfläche für das Wiederherstellen des originalen Kartenausschnittes

### 5.5.5 Ein- und Ausblenden von Ebenen

Dadurch, daß alle Grafikelemente innerhalb bestimmter <g>-Tags zusammengefaßt sind, lassen sich die so gebündelten Einzelemente zentral steuern. Die <g>-Bereiche wirken als Aufbewahrungsbehälter für die enthaltenen Objekte, weshalb sie auch *Container* genannt werden. Setzt man nun für einen dieser Container das Attribut `visibility`<sup>119</sup> auf `hidden`<sup>120</sup>, werden alle darin enthaltenen Grafikobjekte unsichtbar. Umgekehrt kann man natürlich das Attribut wieder auf `visible`<sup>121</sup> setzen und die umschlossenen Elemente werden wieder sichtbar, sofern für die einzelnen Objekte keine anderen Vorgaben vorhanden sind. Auf diese Weise erhält man eine Funktion ähnlich dem Darstellen und Verbergen von Ebenen, wie man es von CAD- oder GIS-Produkten kennt.



Abbildung 60: Die Schwarz-Weiß-Darstellung der Karte

Eine farblose Darstellung könnte zum Beispiel dann interessant sein, wenn der entsprechende Kartenausschnitt auf einem Drucker ausgegeben werden soll, der keine Farben wiedergibt. Durch Anklicken der entsprechenden Schaltfläche im Menü werden die Ebenen mit den Farbfüllungen für die Nutzungsflächen und für die Gebäude ausgeblendet. Zwar bleiben einige Linien, etwa für besondere Grenzzüge oder Topographien weiterhin farbig, dies dürfte sich bei einem Ausdruck in Schwarz-Weiß allerdings kaum negativ auswirken.

<sup>119</sup> engl.: Sichtbarkeit

<sup>120</sup> engl.: versteckt

<sup>121</sup> engl.: sichtbar

Durch das Ein- und Ausblenden der einzelnen Grafikelementgruppen läßt sich die Karte entsprechend den individuellen Wünschen des Betrachters anpassen. Wenn z.B. kein Interesse an der Darstellung der Gebäude besteht, können diese durch einen Mausklick ausgeblendet werden. Da beim Überfahren der Schaltfläche mit dem Mauszeiger ein Untermenü aufklappt, ist sogar eine verfeinerte Auswahl möglich. Bei Gebäuden etwa kann man separat entscheiden ob die Gebäude insgesamt, die Abkürzungen der Gebäudenutzung, die Hausnummern, die Eigennamen der Gebäude oder die Einfärbung entsprechend der Gebäudekategorie angezeigt werden sollen. Diese Einstellungen lassen sich selbstverständlich nahezu uneingeschränkt miteinander kombinieren.

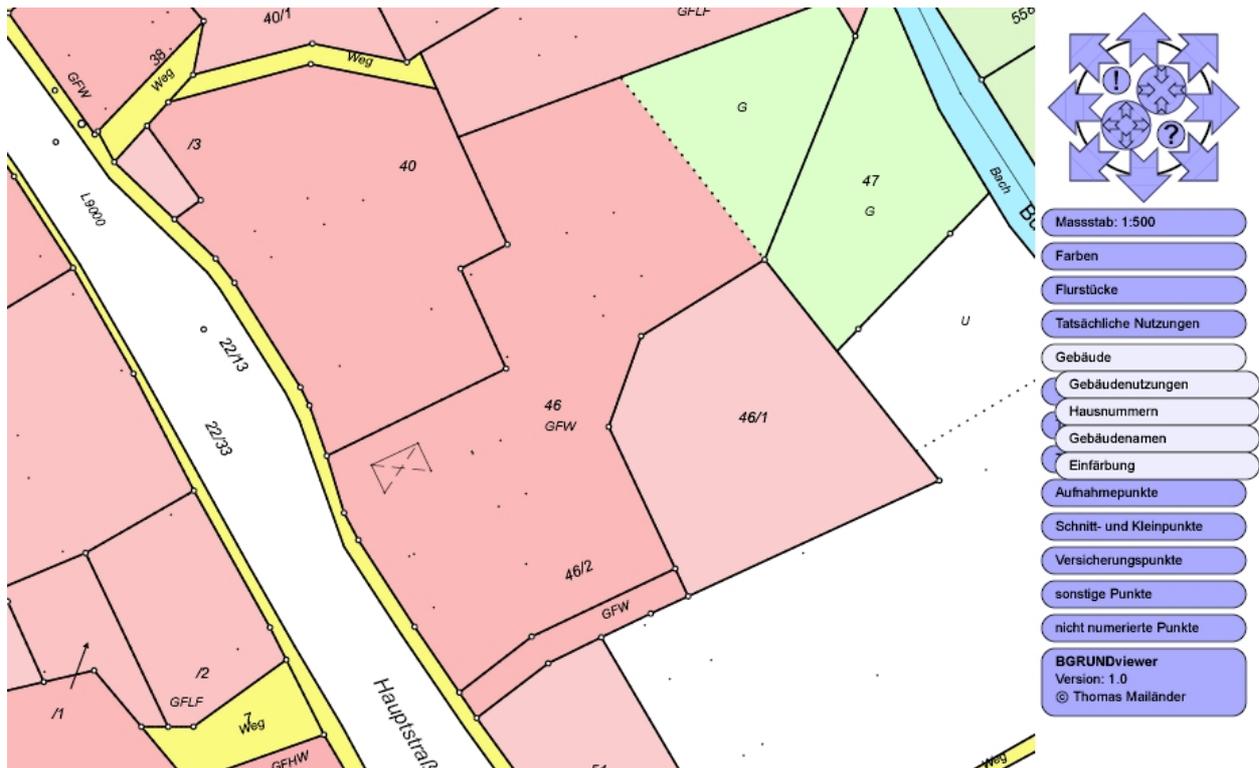


Abbildung 61: Die Darstellung bei unsichtbarer Gebäudeebene

Diese Unterteilungen gibt es in mehr oder weniger großem Umfang auch bei den anderen Objektgruppen. Wenn man das SVG-Dokument im Internet-Browser öffnet, sind fast alle Ebenen sichtbar. Die Ausnahmen hiervon bilden die Bereiche für Punktnummern, da deren Anzeige das Bild eventuell etwas unübersichtlich erscheinen läßt. Wird die Anzeige der Punktnummern jedoch vom Betrachter gewünscht, genügt ein Klick auf die jeweilige Schaltfläche, um die Darstellung einzuschalten.

## 6 Resümee

### 6.1 Die Zukunft von SVG

Für die kommenden Versionen 1.2 und 2.0 von SVG liegt eine Unmenge an Erweiterungs- und Verbesserungsvorschlägen vor. Für den Fall, daß das W3C auch künftig die Zügel für die weitere Entwicklung und Verbreitung der Scalable Vector Graphics so geschickt führt, steht dem Vektorgraphikstandard eine glänzende Zukunft bevor.

Letzten Endes wird aber wahrscheinlich die nächste Browsergeneration darüber entscheiden, ob sich SVG auf breiter Front durchsetzen kann. Das World Wide Web ist inzwischen so bedeutend geworden, daß es einen immensen Einfluß auf neue Technologien ausübt. Diese können, gepuscht durch das WWW einen kometenhaften Aufschwung erleben oder aber nach und nach in der Versenkung verschwinden. Zur Zeit stehen die Vorzeichen für SVG sehr gut. Solange allerdings ein recht großes Plug-In benötigt wird um die Grafiken in Internet-Browsern darzustellen, wird wohl der größte Teil der Internet-Gemeinde einen Bogen darum machen. Dazu kommt, daß momentan nicht einmal für alle Browser ein entsprechendes Plug-In zur Verfügung steht, und selbst bei den Browsern, die von den vorhandenen Plug-Ins unterstützt werden, kann man sich eigentlich nie hundertprozentig sicher sein, welche Funktionen der Erweiterungssoftware letztendlich umgesetzt werden.

So kann man nur hoffen, daß die Entwickler von Internet-Browsern bald auf den SVG-Zug aufspringen und sich dabei an die offiziellen Vorgaben des W3C halten. Denn sicherlich will niemand, daß es bei SVG eine ähnliche Entwicklung geben wird wie es bei HTML und JavaScript bzw. JScript der Fall war.

### 6.2 Mögliche Erweiterungen des BGRUNDviewers

Auch wenn der BGRUNDviewer in seiner jetzigen Form gut dazu eingesetzt werden kann, die Daten der Automatisierten Liegenschaftskarte optisch zu präsentieren, gibt es noch viele Ideen, wie zukünftige Versionen verbessert und erweitert werden könnten.

Zum Beispiel werden in BGRUND Enklaven von tatsächlichen Nutzungsflächen dadurch erzeugt, daß die Nutzungsartengrenze der umschließenden Nutzungsfläche eine Verbindung von einem ihrer Umringpunkte zu einem der Inselfläche erzeugt, von dort aus die Enklave einmal umfährt und anschließend wieder zu ihrem eigenen Umringpunkt zurückkehrt, um von da aus ihren eigentlichen Verlauf fortzusetzen. Als negativer Nebeneffekt wird diese Verbindungslinie bei der automatischen Ausgabe der Nutzungsfläche ebenfalls gezeichnet. Hier wäre es interessant, eine Methode zu entwickeln, die diese Darstellung unterbindet. Jene Methode sollte ebenfalls auf Gebäude anwendbar sein, da auch hier bei einem Vorhandensein von Innenhöfen die unerwünschte Verbindungslinie entsteht.

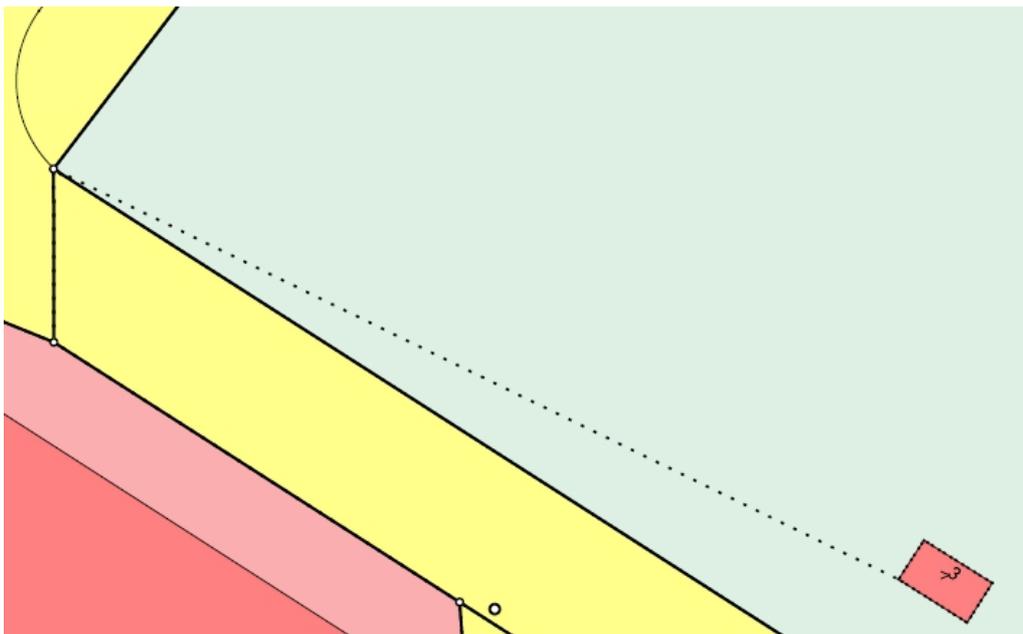


Abbildung 62: Die Umsetzung von Inselflächen in BGRUND

Die Ausgabe der tatsächlichen Nutzungsgrenzen beinhaltet ein weiteres Problem. Durch die Tatsache, daß die Nutzungsflächen normalerweise von zwei Seiten beschrieben und für die Nutzungsartengrenzen eine gestrichelte Linie verwendet wird, kann es zu unschönen Überlappungen kommen.

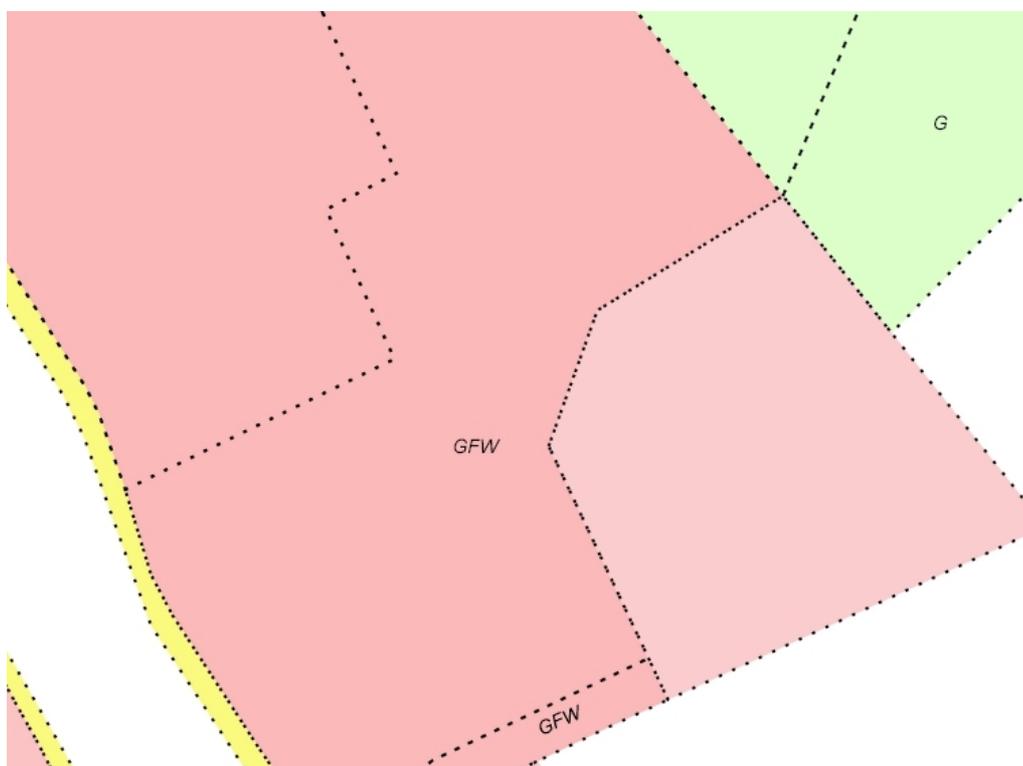


Abbildung 63: Überlappung der Nutzungsartengrenzen

Als Lösung dieses Problems würde sich anbieten, vor jedem Zeichnen einer Linie oder eines Kreisbogens zu überprüfen, ob diese Verbindung schon früher wiedergegeben wurde. Ob dies jedoch ohne bemerkenswerte Leistungseinbuße des Programmes realisierbar ist, darf bezweifelt werden.

In den amtlichen Kartenwerken wird bei der Darstellung der nicht nummerierten Punkte zusätzlich die Genauigkeitsstufe und der Abmarkungszustand berücksichtigt. Diese Informationen sollten in zukünftigen Versionen des BGRUNDviewers ebenfalls durch besondere Symbole optisch wiedergegeben werden.

Ferner können in der BGRUND-Datei spezielle Angaben für die vom Standard abweichende Positionierung einzelner Punktnummern aufgelistet sein. Bisher werden bei dem vorliegenden Programm alle Punktnummern einheitlich, entsprechend der Koordinaten des zugehörigen Punktes, einschließlich eines Zuschlags beim Rechtswert plazierte. Daher sollte der BGRUNDviewer so erweitert werden, daß Angaben in der BGRUND-Datei für eine außergewöhnliche Positionierung der Punktnummer berücksichtigt werden können.

Alternativ oder ergänzend dazu wäre eine Funktion sinnvoll, die es dem Benutzer erlaubt einzelne Texte, wie etwa Punkt- oder Flurstücksnummern zu verschieben, um die Übersichtlichkeit zu verbessern und die Darstellungsqualität zu erhöhen.

Sobald ausreichend BGRUND-Testdaten mit den Beschreibungen der topographischen Objekte gemäß des neuen Formates verfügbar sind, ist das Programm so auszubauen, daß auch punkt-, linien- und flächenförmige topographische Objekte eingelesen und in das SVG-Dokument ausgegeben werden können. Dabei sollte das Programm so aufgebaut sein, daß auch weiterhin Dateien mit der Beschreibung nach dem alten Format verarbeitet werden können.

Im Rahmen dieser Diplomarbeit konnte keine adäquate Methode entwickelt werden die es erlaubt, einen Zusammenhang zwischen mehreren einzelnen topographischen Objekten in Abhängigkeit von ihren Endpunkten und dem Objektschlüssel zu erstellen. Konkret würde dies etwa bei einer Böschungsfäche bedeuten, daß die einzelnen aufgeführten Linien als Bestandteil dieses topographischen Objekts erkannt werden würden. Dadurch ließe sich anschließend die gesamte Böschungsfäche definieren und somit gesondert optisch darstellen, beispielsweise durch eine zusätzliche Farbfüllung mit einer verringerten Deckkraft.

Zum aktuellen Zeitpunkt werden die allgemeinen Angaben und die Sachdaten zu den Punkt-, Flurstücks-, Gebäude-, Text- und topographischen Objekten in externen HTML-Dateien angezeigt. Eine viel elegantere Lösung wäre, diese Mitteilungen im Bereich des Menüs des BGRUNDviewers auszugeben, wie es ja schon bei den Vordaten realisiert ist. Da hier jedoch weniger Platz zur Verfügung steht, müßten diese Informationen voraussichtlich auf das Wesentliche gekürzt werden.

Die Daten aus dem Liegenschaftskataster unterliegen dem Datenschutz. Deshalb sollten vor allem die Einstellungen des Webservers hinsichtlich sicherheitsrelevanter Aspekte überprüft werden. Schließlich soll der Zugriff von Unbefugten auf die Daten unter allen Umständen verhindert werden. Zwar werden die Daten auf dem Server beim Verlassen der Webseite durch ein Skript gelöscht, in der Zwischenzeit wäre jedoch ein Fremdzugriff auf die Daten technisch möglich. Als Lösungsansatz könnten spezielle Zugriffsrechte eingesetzt werden, was unter Umständen eine Benutzerverwaltung und eine Anmeldung durch Benutzernamen und Paßwort erforderlich machen könnte.

Zwar kann der Anwender einen bestimmten Maßstab und Ausschnitt der Karte einstellen, es gibt aber noch keine Funktion, die es ihm ermöglicht, eigenen Text oder gar zusätzliche Grafiken in die Karte einzufügen. SVG bietet grundsätzlich die Möglichkeit, neue Grafikelemente über den Befehl `createElement` in ein bestehendes Dokument aufzunehmen. Problematisch wäre allerdings die Positionierung des erschaffenen Elements, da das Koordinatensystem des Kartenbereichs auf Zentimeterangaben beruht, während hingegen die Angaben für die aktuelle Mauszeigerposition<sup>122</sup> nur in Pixel ermittelt werden können.

Dasselbe Problem mit den nicht zueinander passenden Koordinatensystemen müßte auch dann gelöst werden, wenn man an einer bestimmten Stelle des Menüs fortwährend die aktuellen Koordinaten des Mauszeigers anzeigen lassen wollte, wie es bei vielen CAD- und GIS-Produkten gängig ist.

Durch die vorgegebene Hardware während der Entwicklung des Programmes erschien eine Begrenzung für die Größe der umzusetzenden BGRUND-Datei auf 650 Kilobyte als sinnvolle Maßnahme. Mit der Verwendung leistungsfähigerer Komponenten ließe sich die Dateigröße sicherlich noch erhöhen.

So zeigten sich bei Verwendung eines PCs mit einem 1,7 GHz-Prozessor als Client-Rechner wesentliche Leistungsunterschiede gegenüber seinem Gegenstück mit 450 MHz. Somit wären moderne Rechner mit Gewißheit problemlos in der Lage größere SVG-Grafiken zu verarbeiten.

Der kritische Aspekt liegt daher bei der Auswahl des Server-Rechners. Der Server, der auf der Internetseite [www.BGRUNDviewer.de](http://www.BGRUNDviewer.de) die Verarbeitung der verwendeten PHP-Skripte ausführt, ist zwar recht schnell, jedoch muß man bedenken, daß über jenen Server nicht nur diese Webseite betrieben wird. Aus diesem Grund ist eine übermäßige Belastung unbedingt zu verhindern.

Kommt das Programm hingegen in einem Intranet zum Einsatz, könnte speziell für die Verarbeitung der BGRUND-Dateien ein Server eingerichtet werden, wodurch mit hoher Wahrscheinlichkeit die Umsetzung größerer Dateien bewerkstelligt werden könnte.

Um dem Benutzer des BGRUNDviewers den Download sämtlicher erstellter Dateien zu vereinfachen, sollte dafür in zukünftigen Versionen des Programmes eine eigene Funktion zur Verfügung stehen. Da es mittels PHP und einem eigens dafür vorgesehenen Moduls machbar ist, Dateien auf dem Server zu komprimieren, würde sich durch die Kombination dieser Techniken eine hervorragende Methode für den effizienten Download der Daten ergeben.

Für den Betrachter der Karte wäre es sicherlich auch interessant, sich auf Wunsch Spannmaße zwischen zwei beliebigen Punkten anzeigen zu lassen, sei es um zukünftige Maßnahmen vor Ort besser planen zu können oder um einen bestimmten Grenzpunkt in der Örtlichkeit zu suchen.

Außerdem könnte eine Flächenberechnung für Flurstücke, Nutzungs- oder Gebäudeflächen ebenfalls eine sinnvolle Erweiterung darstellen, es müßte aber darauf hingewiesen werden, daß diese Angaben keinen offiziellen Wert liefern können, da allein die Einträge im Grundbuch eine rechtlich bedeutsame Rolle einnehmen.

---

<sup>122</sup> Die Koordinaten des Mauszeigers lassen sich durch die Funktionen `evt.getClientX()` bzw. `evt.getClientY()` ermitteln, `evt` steht hierbei für das Ereignis das die Funktion ausgelöst hat. In diesem Fall ist das Ereignis `onmousemove` sinnvoll, da somit die Funktion aufgerufen wird sobald die Maus bewegt wird.

### 6.3 Zusammenfassende Schlußbetrachtung

Zum Abschluß dieser Diplomarbeit kann festgehalten werden, daß mit dem erstellten Programm der beabsichtigten Zielgruppe, in erster Linie Privatpersonen und kleinere Unternehmen, ein Hilfsmittel zur Verfügung gestellt wird, mit dem die wichtigsten Daten der Automatisierten Liegenschaftskarte dargestellt werden können.

Die Menüfunktionen werden dem Anwender schon nach kurzer Einarbeitungszeit die Möglichkeit geben, sich die Karte nach den eigenen Wünschen anzeigen und ausdrucken zu lassen. Außer dem SVG Viewer von Adobe, einem dazu kompatiblen Internet-Browser und dem Zugang zur Internetseite des BGRUNDviewers werden dabei keine besonderen Software- und Hardwarevoraussetzungen gestellt.

Das Angebot kann kostenlos genutzt werden, das heißt, es fallen beim Benutzer lediglich die Kosten für seinen Internetzugang an.

Inwieweit das Programm letztendlich auch genutzt wird, hängt sicherlich davon ab, ob es in der Öffentlichkeit auf breiter Front publik gemacht werden kann. Dabei wäre es hilfreich, wenn die Bürger bei den staatlichen Vermessungsämtern und dem Landesvermessungsamt auf dieses Angebot hingewiesen werden würden.

Ansonsten müssen die ersten Reaktionen der Anwender abgewartet werden um weitere Anregungen zu gewinnen, welche Funktionen zusätzlich gewünscht sind bzw. was prinzipiell verbessert werden könnte.

Hinsichtlich SVG kann angeführt werden, daß es sich im Großen und Ganzen sehr gut für die Darstellung von Karten einsetzen läßt, wobei es vor allem durch die Skalierung ohne Qualitätseinbußen seine Stärken ausspielen kann. Natürlich gab es auch mehr oder weniger große Probleme, genannt dafür seien etwa die Gebäudeschraffuren oder die Ermittlung der aktuellen Mauszeigerposition.

Während der Entwicklung des Programmes wurde, mit Ausnahme des Betriebssystems, nur kostenlose Software verwendet. Die PHP-Skripte wurden mit einem gewöhnlichen Texteditor erstellt. Dies war daher gut durchführbar, da die Fehlermeldungen des PHP-Moduls bzw. des CGI-Programmes in den meisten Fällen recht brauchbare Informationen lieferten.

### 6.4 Danksagungen

An dieser Stelle möchte ich mich bei allen Dozenten und Mitarbeitern des Studienganges Vermessung & Geoinformatik der Fachhochschule Stuttgart – Hochschule für Technik, für die Betreuung und Unterstützung während meines Studiums bedanken.

Ganz besonders bedanke ich mich bei meinen Betreuern Herrn Prof. Dr.-Ing. Franz-Josef Behr und Herrn Prof. Dr.-Ing. Hardy Lehmkuhler für die Unterstützung bei meiner Diplomarbeit. Herr Prof. Dr.-Ing. Behr machte mich vor etwa einem halben Jahr auf die Scalable Vector Graphics aufmerksam, wodurch diese Diplomarbeit überhaupt erst entstehen konnte. Durch seine persönlichen Erfahrungen mit SVG konnte er bei auftretenden Schwierigkeiten immer wieder nützliche Tips geben, ließ mir aber ansonsten stets freie Hand bei der Ausführung meiner Diplomarbeit, was ich sehr zu schätzen weiß.

Bedanken möchte ich mich auch bei Herrn Bernhart vom Landesvermessungsamt, sowie bei Herrn Bölz und Herrn Grau vom staatlichen Vermessungsamt Plochingen für die freundlichen Auskünfte zum BGRUND-Format.

Ein ganz spezieller Dank gilt meiner Freundin Petra für die Aufmunterung, wenn es nicht wie gewünscht mit der Diplomarbeit voranging, und für die Nachsicht die sie mir, vor allem gegen Ende der Arbeit, entgegenbrachte.

Natürlich möchte ich mich auch bei meinen Eltern ganz herzlich dafür bedanken, daß sie mir während meines Studiums stets finanziell und vor allem auch menschlich beistanden.

## **Anhang: Die Dateien des Programmsystems BGRUNDviewer**

Die Dateien werden auf einer separaten CD-ROM abgegeben.

### **1. Upload der BGRUND-Datei**

**dateiupload.php**

### **2. Einlesen und Verarbeiten der BGRUND-Datei**

**dateiauslesen.php**

### **3. Auflistung der assoziativen Arrays als Verzeichnisse für das Programm**

**schluesselverzeichnisse.php**

### **4. Die Klasse Vordaten**

**vordaten.php**

### **5. Die Punkteklassen**

**punkt.php**

### **6. Die Klasse Punkteliste**

**punkteliste.php**

### **7. Die Klasse Linienzug**

**linienzug.php**

### **8. Die Klasse Grenzzug**

**grenzzug.php**

### **9. Die Klasse Flurstück**

**flurstueck.php**

## **10. Die Klasse Flurstücksliste**

flurstuecksliste.php

## **11. Die Klasse tatsächliche Nutzungsfläche**

nutzungsflaeche.php

## **12. Die Klasse Gebäude**

gebaeude.php

## **13. Die Klasse Gebäudeliste**

gebaeudeliste.php

## **14. Die Klasse topographisches Objekt**

topObjekt.php

## **15. Die Klasse Liste der topographischen Objekte**

topObjekteListe.php

## **16. Die Klasse Text**

text.php

## **17. Die Klasse Texteliste**

texteliste.php

## **18. Allgemeine Funktionen**

funktionen.php

## **19. Anlegen und Löschen des benötigten Verzeichnisses und der Dateien**

dateien.php

## **20. Einbindung der Datenbankfunktionen**

datenbank.php

## **21. Auflistung der durch die Verarbeitung erstellten Objekte**

**ergebnis.php**

## **22. Umschließender SVG-Bereich für die Aufnahme des Karten- und des Menüteiles**

**svg.php**

## **23. CSS- Definitionen und Anlage der Filter, Verläufe und Symbole**

**style.php**

## **24. JavaScript-Funktionen für die Interaktionen der erstellten Karte**

**svg.js**

## **25. SVG-Bereich mit der Kartendarstellung**

**karte.php**

## **26. SVG-Bereich mit der Menüdarstellung**

**menue.php**

## **27. Löschen des Verzeichnisses und der darin enthaltenen Dateien auf dem Server**

**loeschbestaetigung.php**

## Glossar

**AdV:** Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland; Arbeitskreis der Ständigen Konferenz der Innenminister und -Senatoren, in dem die für die Landesvermessung und das Liegenschaftskataster zuständigen Verwaltungen der Länder wirken. In der AdV werden fachliche Angelegenheiten von grundsätzlicher und überregionaler Bedeutung mit dem Ziel einer einheitlichen Regelung behandelt. Dies soll den Problemen eines föderalistisch organisierten Vermessungswesens in der Bundesrepublik entgegenwirken. Die AdV hat u.a. das ALKIS/ATKIS-Modell und AFIS konzipiert. Internet: [www.adv-online.de](http://www.adv-online.de)

**AFIS:** Amtliches Festpunktinformationssystem; zukünftiges Informationssystem der Vermessungsverwaltungen der Bundesländer mit Informationen zu Lage-, Schwere- und Höhenfestpunktfeld.

**ALB:** Automatisiertes Liegenschaftsbuch; enthält die Angaben über Flurstücke und Gebäude sowie die Ergebnisse der amtlichen Bodenschätzung bei landwirtschaftlichen Flächen. Die Eigentümer werden in Übereinstimmung mit dem Grundbuch nachrichtlich geführt. Das ALB liegt flächendeckend für ganz Baden-Württemberg vor.

**ALK:** Automatisierte Liegenschaftskarte; vermessungs- und kartentechnischer Teil des Liegenschaftskatasters, als Gegenstück zum Automatisierten Liegenschaftsbuch (ALB) und dem Zahlenwerk. Die ALK untergliedert sich in den Datenbankteil mit der Punktdatdatei, der Grundrißdatei mit Bezug zu Fachdatenbanken, der Messungselementdatei, Systemdateien und dem Auftragsbuch sowie dem Verarbeitungsteil zur grafischen Verarbeitung mit dem grafisch-interaktiven Arbeitsplatz (ALK-GIAP) zur Erfassung, Fortführung und Präsentation und zur Auftragsverwaltung.

**ALKIS:** Amtliches Liegenschaftskataster Informationssystem; Modell der AdV zur Integration von ALB und ALK in ein einheitliches Datenmodell in Analogie zu ATKIS, derzeit in der Entwicklung.

**Andreessen, Marc:** entwickelte zusammen mit Eric Bina den Mosaic-Browser am National Center for Computer Applications (NCSA) der Universität von Illinois. Gründete später Netscape und trug durch seinen Netscape Navigator zum großen Erfolg des World Wide Webs bei.



**Anti-Aliasing:** auch Kantenglättung genannt zur Verminderung des „Treppeneffektes“, der durch Pixeldarstellung bei schrägen und gekrümmten Linien entsteht. Durch Interpolation - farbliche Angleichung benachbarter Bildpunkte - werden die "Treppenstufen" ausgeglichen. Dabei kann es allerdings passieren, daß die Darstellung von Linien breiter erscheint.

**Anwendungsschicht:** Die Anwendungsschicht stellt anwendungsspezifische Protokolle für die Kommunikation zwischen zwei Prozessen zur Verfügung. In Abhängigkeit vom gewählten Dienst (Anwendung) wird ein spezifisches Protokoll verwendet, das die Anwendungsdaten (z.B. den HTML-Code bei HTTP oder den E-Mail-Header bei SMTP) überträgt.

**Apache:** weitverbreiteter Open Source Webserver; ursprünglich aus dem NCSA-Webserver V1.3 der Universität von Illinois in Urbana-Champaign hervorgegangen.

**ASCII:** Abkürzung für "American Standard Code for Information Interchange" (Amerikanische Standarddarstellung für den Austausch von Informationen); der ASCII-Standard hat sich in den 60er Jahren zur Zeit der Datenübertragung mittels Telex entwickelt. Für jedes Zeichen gibt es einen Zahlencode, der es ermöglicht, zwischen verschiedenen Systemen Texte auszutauschen. Mit 7 Bit erfaßt der ASCII-Code zunächst aber nur das "einfache" Alphabet (!, ", #, \$, ..., 0, 1, 2, 3, ..., A, B, C, D, ..., a, b, c, d, ..., {, |, }) und Steuerungs-codes, die z.B. für die Druckersteuerung benötigt werden - insgesamt sind das 128 Zeichen. Nicht berücksichtigt sind darin Sonderzeichen - also deutsche Umlaute, das "ß" und andere nationale Sonderzeichen, die erst im erweiterten ASCII-Code enthalten sind, der auf einer 8 Bit-Struktur aufbaut. Auch heute ist es in der Telekommunikation - z. B. per E-Mail - noch unüblich, Sonderzeichen zu verwenden, denn sie werden von anderen Systemen unter Umständen falsch interpretiert.

**ATKIS:** Amtliches Topographisch-Kartographisches Informationssystem; stellt die topographische Basisinformation der fachlichen Informationssysteme im großräumigen mittel- bis kleinmaßstäblichen Bereich dar. ATKIS ([www.atkis.de](http://www.atkis.de)) zielt auf einen bundeseinheitlichen, digitalen topographischen Basisdatenbestand über die Objekte, die Erscheinungsformen und das Relief der Erdoberfläche. ATKIS gilt heute als das landschaftsbeschreibende Geo-Informationssystem (GIS) der deutschen Landesvermessung. Die Landschaftsbeschreibung erfolgt in unterschiedlichster Form und kann somit verschiedenste Anwenderanforderungen erfüllen. ATKIS liefert objektbasierte, signaturbasierte und bildbasierte Beschreibungen der Erdoberfläche in Form folgender digitaler Modelle: Digitale Landschaftsmodelle (DLM), Digitale Geländemodelle (DGM), Digitale Topographische Karten (DTK) und Digitale Orthophotos (DOP). ATKIS-Datenbestände liegen bundesweit vor. Das ATKIS-Datenmodell wird derzeit durch die AdV überarbeitet und auf das in Konzeption befindliche ALKIS-Modell abgestimmt.

**Berners-Lee, Tim:** entwickelte am Schweizer Kernforschungsinstitut CERN Anfang der 90er das WWW, um wissenschaftliche Arbeiten besser kommunizieren zu können.



Dazu gehörte der erste Browser, er nannte ihn WorldWideWeb, der erste Server (<http://info.cern.ch/>) sowie das Protokoll HTTP und die Konzeption des URL. Später wurde der Name des Browsers für den neuen Internetdienst verwendet. Um die Verwechslung mit diesem zu vermeiden, benannte er den Browser in Nexus (engl. Verbindung) um. Das Programm lief unter dem Betriebssystem NextStep für den Rechner NeXT. Es hatte einen integrierten Editor. Tim Berners-Lee ist heute am MIT LCS und Direktor des W3C.

**Bézierkurve:** Algorithmus, nach Bézier benannt, der eine Kurve eines bestimmten Grades  $n$  durch  $n+1$  Kontrollpunkte definiert. In der Praxis sind Änderungen am Kurvenverlauf durch Verschiebungen der Kontrollpunkte nur bedingt möglich, da der direkte Einfluß auf die Punkte nicht möglich ist. Die Verallgemeinerung der Bézierkurve führte zur Splinekurve.

**Bina, Eric:** entwickelte zusammen mit Marc Andreessen an der NCSA der Universität Illinois den MO-SAIC-Browser, der den unglaublichen Erfolg des World Wide Web in die Wege leitete.

**BGRUND:** Bearbeitung und Benutzung der zentralen Grundrißdatei; Format für den Datenaustausch von Grundrißdaten der Automatisierten Liegenschaftskarte (ALK) in Baden-Württemberg. Das landesspezifische Ausgabeformat BGRUND transportiert nicht nur Schraffuren, Linien und Symbole, sondern enthält auch Sachdaten, die beispielsweise Flurstücken und Gebäuden zugeordnet sind.

**BMP:** Bitmap; Dateiformat für Grafikdateien. Bitmap-Dateien haben die Dateierweiterung ".bmp". Bei schwarz-weißen Bitmap-Grafiken stellt jedes Bit einen Pixel (Bildpunkt) der Abbildung dar. Farbdarstellungen sind mit mehreren Bits pro Pixel codiert. Bitmap-Grafiken sind nur bedingt skalierbar.

**Browser:** von engl. "to browse", dt.: schmökern, blättern, umherstreifen. Bezeichnung für ein Programm zum grafischen Zugriff auf das World Wide Web. Der Browser stellt eine in der Seitenbeschreibungssprache HTML verfaßte Webseite dar, indem er die Tags interpretiert und ausführt.

Die bekanntesten Browser sind der Netscape Navigator und der Internet Explorer. Die Funktionen eines Browsers können durch Plug-Ins erweitert werden. So enthalten die gängigen Browser zum Beispiel eine "Java Virtual Machine", in der Java-Programme laufen können, und durch ein Plug-In von Adobe (SVG Viewer) können Grafiken im SVG-Format angezeigt werden.

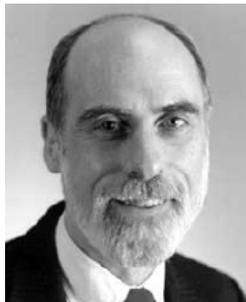
**Bug:** dt.: Wanze, Insekt; ein Fehler in der Programmierung, der das Programm im ungünstigsten Fall abstürzen läßt oder zumindest bewirkt, daß es nicht in der gewünschten Form abläuft.

Solche Programmierfehler können durch die Korrektur und Neukompilierung des Programm-Codes behoben werden. Bei Anwendungen allerdings, die schon an Endverbraucher ausgeliefert worden sind, können Bugs nur durch einen sogenannten Patch korrigiert werden. Ein Patch ist ein Code-Stück, das nachträglich in ein bereits installiertes Programm eingefügt wird.

Die Herkunft der Bezeichnung Bug ist nicht geklärt. Einigen Anekdoten zufolge verursachten Insekten bei den ersten Rechnern ständig Kurzschlüsse in den damals noch recht sensiblen Schaltkreisen, welche zu Fehlern bei der Ausführung von Programmen führten. Laut einer anderen These jedoch soll die Bezeichnung aus der Zeit stammen, in der Daten noch auf Lochkarten gespeichert wurden. Wurden diese Lochkarten von Ungeziefer angefressen, enthielten sie zusätzliche Löcher an der falschen Stelle, die zu Fehlern im Programm führten.

Der Begriff Bug wird manchmal auch verwendet, um einen Hardware-Fehler zu bezeichnen.

**Cerf, Vinton:** „Vater des Internet“; erster Präsident der International Network Working Group (INWG), veröffentlichte 1974 gemeinsam mit Robert Kahn das Transmission Control Protocol (TCP) und das Internet Protocol (IP), die gemeinsame Sprache aller Internet Computer.



**CERN:** Conseil Européenne pour la Recherche Nucléaire; europäisches Kernforschungszentrum in Genf, an dem Tim Berners-Lee 1989 die Grundlage für das World Wide Web entwickelte.

**Client:** dt.: Kunde, Auftraggeber; Begriff aus dem Netzwerkbereich: ein Client nimmt Dienste in Anspruch, deshalb wird eine an den Server angeschlossene Arbeitsstation als Client bezeichnet. Der Client schickt Anfragen des Benutzers in einem speziellen Protokoll an den Server und stellt dessen Antworten in lesbarer Weise auf dem Bildschirm dar.

**Compiler:** von engl. „to compile“, dt.: aufstellen, erarbeiten, erstellen, übersetzen; Ein Übersetzungsprogramm, welches Programme einer höheren Programmiersprache in die vom Prozessor ausführbare Maschinensprache umwandelt.

**CSV:** CSV-Dateien sind ASCII-Dateien, die häufig benötigt werden, um den Inhalt einer Datenbank (z.B. dBASE, ACCESS, SQL-Datenbank) zu extrahieren und in eine andere Datenbank wieder einzulesen. Dabei entspricht häufig ein Datensatz (Datenblatt) einer Zeile, und die Zellen werden durch ein Separationszeichen getrennt. Dieses Zeichen muß nicht ein Komma sein; häufig wird das Semikolon verwendet oder das "|" -Zeichen. Denn es darf natürlich als Separator (Trennzeichen) kein Zeichen werden, welches als Inhalt auch nur eines Datensatzes vorkommen könnte - z.B. in einem Namen wie "Müller, Maier und Partner".

**DFN:** Deutsches Forschungsnetz; 1984 gegründeter Verein deutscher Hochschulen und Forschungseinrichtungen für den Betrieb und die Nutzung eines eigenen Forschungsnetzes. Der DFN-Verein betreibt das Wissenschaftsnetz WiN, an das Einrichtungen im Bereich Wissenschaft, Bildung oder Kultur, sowie Schüler, Studenten und Journalisten angeschlossen sind.

**DNS:** Domain Name Service, -Server oder -System; dezentraler Dienst, der Rechner-Namen bzw. Internet-Adressen im Klartext (z.B. www.BGRUNDviewer.de) und IP-Adressen (z.B. 193.196.138.81) einander zuordnet. Für jeden Server beziehungsweise für jedes LAN mit Internet-Anschluß, muß ein DNS-Server diese Informationen verwalten. Sobald eine Seite im Internet angewählt wird, fragt der Browser zuerst einen Domain Name Server. Dieser meldet die entsprechende numerische Adresse zurück, worauf der Browser eine direkte Verbindung zur IP-Adresse aufbauen kann.

**DOM:** Document Object Model; das DOM legt fest, welche Komponenten eines Dokumentes beispielsweise für eine Programmiersprache zugänglich sind:

- Die Programmiersprache regelt dabei, *wie* der Zugriff erfolgt,
- das DOM gibt vor, *was* zugreifbar ist.

Hinter jeder Sprache, die etwa im Web-Browser die clientseitige Behandlung von HTML-Dokumenten ermöglicht, steckt implizit ein Document Object Model. Dadurch ist es beispielsweise möglich, per JavaScript ein Bild gegen ein anders auszutauschen.

**DXF:** Drawing Exchange Format; von der Firma Autodesk für das Softwarepaket AutoCAD entwickelt ist das DXF-Format der derzeit von den meisten CAD-Systemen unterstützte Standard für den Austausch von Vektordaten.

**DSL:** Digital Subscriber Line-Technologie; DSL bietet eine Technologie, um das Internet mit 1,5 Millionen Bits pro Sekunde über Kupferleitungen zu betreiben.

**Enquire::** von engl. „to enquire“, dt.: anfragen, sich erkundigen, nachforschen; von Tim Berners-Lee entwickeltes Hypertext-Programm, mit dem Berners-Lee Adressen, Gesprächsnotizen, spontane Ideen, Erlebnisse und Arbeitsergebnisse festhalten wollte. Diese Daten waren durch bidirektionale Links miteinander verknüpft, weshalb man auch behaupten kann, daß der Enquire quasi ein Vorgänger der WWW-Browser darstellt.

**Erwise:** Einer der ersten Browser mit grafischer Oberfläche (1992)

**Flash:** 1993 von Jonathan Gay entwickelte proprietäre, vektororientierte Werkzeuge der Firma Macromedia, mit denen sich multimediale Webanwendungen entwickeln lassen. Ein Flash-Plug-in macht es möglich, solche Anwendungen als Teil einer Webseite einzusetzen.

**FTP:** File Transfer Protocol; Datei-Übertragungsprotokoll, das auf dem Internet-Protokoll TCP/IP basiert. Mit FTP kann man Text-, Bild-, Programm-, Ton- und Videodateien von einem anderen Rechner,

dem FTP-Server, herunter- oder auf ihn hinaufladen. FTP-Server haben einen frei zugänglichen Bereich ihrer Festplatte, von dem man zum Beispiel Informationen, Public Domain- und Shareware-Programme kostenlos herunterladen kann. Nutzer können dabei auf Dateien zugreifen, die auf verschiedenen Rechnertypen liegen, unabhängig davon, welchen Rechnertyp und welches Betriebssystem sie selbst benutzen. Dazu braucht man ein FTP-Programm, das die Verbindung zum Server herstellt und die Daten überträgt.

**GEODIS:** Geodaten-Informationssystem; LVShop des Landesvermessungsamtes Baden-Württemberg, in dem die Produkte des Landesbetriebes Vermessung bestellt werden können. Diese gliedern sich in analoge Produkte wie Landkarten, Druckschriften oder Luftbilder und digitale Produkte, die sogenannten Geodaten. Darin enthalten sind die Daten der Automatisierten Liegenschaftskarte oder des Automatisierten Liegenschaftsbuches, aber auch Rasterdaten zu den topographischen Kartenwerken oder Auszüge aus dem Digitalen Landschaftsmodell (DLM), dem Digitalen Höhenmodell (DHM), Digitalen Orthophotos (DOP) oder dem Festpunktverzeichnis. Es ist im Internet unter [www.lv-bw.de](http://www.lv-bw.de) erreichbar.

**GIF:** Graphics Interchange Format; Rastergrafik-Format das von allen grafikfähigen Browsern dargestellt werden kann, außerdem ist eine Komprimierung mit hohem Wirkungsgrad möglich. GIF 87a erlaubt es, mehrere Bilder in einer Datei zu speichern, um eine Animation ablaufen zu lassen. GIF 89a ermöglicht zusätzlich die Steuerung des Animationsablaufs und unterstützt transparente Bereiche.

**GNU-GPL:** GNU-Software wird auf der Grundlage der GNU General Public Licence (GPL) vertrieben. Diese basiert wiederum auf dem Prinzip des "Copyleft" (das Gegenteil von "Copyright"), das im Gegensatz zum restriktiven Urheberrecht das freie Kopieren und Modifizieren von Software ausdrücklich erlaubt und vom Hersteller verlangt, daß der Quellcode mitveröffentlicht wird. Der wesentliche Unterschied zu Public Domain-Software besteht darin, daß auch modifizierte Software unter den gleichen Bedingungen weitervertrieben werden darf. Eine "Enteignung", wie sie bei der Modifizierung oder Weiterentwicklung von Public Domain-Software durch Dritte geschehen kann, ist daher rechtlich ausgeschlossen. Es werden im Gegenteil hierdurch Anreize für Programmierer geschaffen, da sie sicher sein können, daß ihre Arbeit nicht mißbraucht wird und daß auch sie von etwaigen Verbesserungen ihrer Programme profitieren können.

Aus diesem Grund wird viel Software, die eigentlich kein Bestandteil des GNU-Projekts ist, unter der GNU-GPL-Lizenz vertrieben. Hierzu zählt etwa das Betriebssystem Linux. Es gibt mittlerweile neben UNIX noch zahlreiche andere Programme für Betriebssysteme, die auch unter der GNU-GPL-Lizenz vertrieben werden.

**Host:** dt.: Gastgeber, Hausherr; Großrechner und Server, an denen Arbeitsstationen angeschlossen sind, für die innerhalb eines Netzwerks besondere Dienste bereitgestellt werden. Einige Hosts sind z.B. News-, FTP- oder Name-Server, andere sind Router oder HTTP-Server, die das "Hypertext Transfer Protocol" bereitstellen, auf dem das World Wide Web basiert.

**HTML:** Hypertext Markup Language; Standardisierte Seitenbeschreibungssprache für WWW-Seiten im Internet bzw. Intranet. Sie definiert sowohl die Gestaltung, den Inhalt und die Grafik der Seite als auch die Links zu eigenen oder fremden Seiten. HTML-Dokumente können mit jedem beliebigen Texteditor erstellt werden, da sie aus reinem ASCII-Text bestehen. Spezielle Editoren erleichtern allerdings das Erstellen der Dokumente erheblich. Außerdem gibt es für viele Textverarbeitungsprogramme Add-Ons, die das HTML-Format mehr oder weniger begrenzt unterstützen.

**HTTP:** Hypertext Transfer Protocol; Kommunikationsprotokoll zwischen Webserver und Webbrowser zur Übertragung von HTML-Daten. Das HTTP-Protokoll stellt die oberste von mehreren Protokoll-"Schichten" zur Verwaltung im Internet dar:

**IIS:** Internet Information Services; von Microsoft bereitgestellte Dienste, mit denen aus einem Rechner mit den Server-Versionen der Betriebssysteme MS Windows NT 4.0, 2000 oder XP ein Internet-Server wird. Es stellt unter anderem die notwendigen Verwaltungsprogramme für TCP/IP sowie die Protokolle HTTP, FTP, SMTP und NNTP zur Verfügung.

**Internet:** von „Interconnected Networks“; weltweit größtes Computernetzwerk, das aus vielen miteinander verbundenen Netzwerken und auch einzelnen Ressourcen besteht. Zu den wichtigsten Leistungen des Internets - man spricht auch von "Diensten" – gehören:

- die elektronische Post (E-Mail),
- hypertextbasierter Content - also Inhalte - mit entsprechenden Suchdiensten (WWW),
- Dateitransfer (FTP) und
- Diskussionsforen (Usenet / Newsgroups).

Weitere Dienste sind:

- Gopher, Telnet, Finger, Whois (als Vertreter einiger "alter" Dienste),
- Chat, Telefon, Video, ICQ, ...

**IMAP:** Internet Message Access Protocol; Internet-Protokoll für den Zugriff auf zentrale Mail-Server, die E-Mail für andere Rechner bereithalten. IMAP ermöglicht die Bearbeitung bzw. Manipulation von E-Mails auf einem Mail-Server, da sie nicht -wie bei POP3 der Fall- nach dem Herunterladen vom Server gelöscht werden. Nutzer können daher über IMAP von verschiedenen Rechnern aus auf ihre E-Mails zugreifen, auch auf solche, die bereits abgerufen wurden.

**Intranet:** Internes Computer-Netz in Organisationen und Unternehmen auf Basis der Internet-Protokolle (TCP/IP), das oft (über eine Firewall) mit dem Internet verbunden ist.

**ISDN:** Integrated Services Digital Network; digitales Kommunikationsnetz für den Transfer von Daten, Texten, Sprache und Bildern über Telefonleitungen. Ein ISDN-Basisanschluß besteht aus zwei Datenkanälen (B-Kanäle) und einem Steuerkanal (D-Kanal).

**JavaScript / JScript:** JavaScript ist eine Skriptsprache für das World Wide Web, die von der Firma Netscape entwickelt wurde und Web-Seiten mit interaktiven und dynamischen Elementen versehen kann. JavaScript wurde in Anlehnung an, jedoch unabhängig von der Programmiersprache Java, entwickelt. JavaScript ist frei verfügbar und kann ohne Lizenz von jedermann in Browser-Software implementiert werden. Während der Netscape Navigator JavaScript vollständig unterstützt, ist beim Internet Explorer von Microsoft nur eine abgespeckte und geringfügig modifizierte Version der Sprache JScript, implementiert worden.

**JPEG/JPG:** Joint Photographic Expert Group; beliebter Komprimierungsstandard für unbewegte Bilder von der "Joint Photographic Experts Group". Das Prinzip der Kompression nach dem JPEG-Verfahren beruht unter anderem darauf, daß sehr ähnliche Farbwerte in dem zu komprimierenden Bild gleichgesetzt werden.

**JSP:** Java Server Page; Technologie, mittels derer Java-Servlets in HTML-Webseiten eingebunden werden können. Dadurch läßt sich die Seite, die ein Webserver einem Browser übermittelt, bestimmen. Wäh-

rend der HTML-Code das Layout bestimmt, wird durch die Servlets beispielsweise festgelegt, welcher Text in bestimmten Textfeldern erscheint. Das kann das Ergebnis einer Datenbankabfrage sein, die über das Servlet gesteuert wurde. JSP ist eine auf Java basierende Konkurrenz-Technologie von Sun zur Active Server Page (ASP) von Microsoft. Beide dienen zur Erstellung dynamischer Web-Seiten.

**Kahn, Robert:** entwickelte zusammen mit Vinton Cerf das Verfahren TCP/IP für die Kommunikation unter den dem Internet angeschlossenen Rechner.

**Kataster:** allgemein Register, Sammelbezeichnung für Verzeichnisse und graphische Darstellungen gleichartiger Gegenstände. Meist in Wortverbindungen genutzt, wie z.B. Liegenschaftskataster, Jagdkataster, Straßenkataster etc. Im spezielleren GIS-Kontext handelt es sich um eine systematische Datensammlung über Landeigentum und daran knüpfende Rechte. Ein solches Liegenschaftskataster beinhaltet eine Karten-, eine Buch- und eine Zahlensicht auf die Liegenschaften und stellt ein öffentliches Register der Eigentumsverhältnisse des Grund und Bodens eines Landes, Staates oder einer Kommune dar.

**Konstruktor:** Klassenmethode, die keinen Rückgabewert deklarieren darf und als Methodennamen den Klassennamen hat. Dabei sollten in 99 % aller Fälle unter allen Umständen

- Aus- und Eingaben vermieden
- alle Datenelemente initialisiert
- ggf. Übergabeparameter auf zulässigen Wertebereich getestet werden.

Ein Konstruktor wird automatisch bei der Definition eines Klassenobjekts aufgerufen. Ein explizites Aufrufen von Konstruktoren wird nur selten verwendet (z.B. bei der Erzeugung von temporären Klassenobjekten ohne Zuordnung eines Bezeichners).

Man unterscheidet bei Konstruktoren:

- Standardkonstruktor (Anlegen eines Objekts mit Standardwerten)
- Kopierkonstruktor (Anlegen eines Objekts mit den Werten eines bereits existierenden Objekts gleicher Klasse)
- Typumwandlungskonstruktor (Anlegen eines Objekts aus einem Wert einer anderen Klasse oder eines Standarddatentyps)
- allgemeiner Konstruktor (Anlegen eines Objekts aus mehreren Werten von Datentypen)

**LAMP:** Kombination aus Linux-Betriebssystem, Apache-Webserver, MySQL-Datenbanksystem und PHP-Skripten

**Lerdorf, Rasmus:** konzipierte 1994 PHP als ein Hilfsmittel für seine Homepage. Zuerst bestand es aus einem extrem simplifizierten Parser, der ausschließlich einige spezielle Makros verstand, und einigen Werkzeugen, die damals häufig auf Webseiten genutzt wurden: ein Gästebuch, ein Counter und einige andere. Der Parser wurde Mitte 1995 neu programmiert und in PHP/FI umbenannt. Das FI kam von einem anderen Paket, das Rasmus geschrieben hatte und das HTML-Formulardaten interpretierte. Er kombinierte die "Personal Home Page Tools"-Skripts mit dem "Formular-Interpreter", fügte noch mSQL-Unterstützung hinzu und PHP/FI war geboren.

PHP/FI wuchs mit unglaublicher Geschwindigkeit und immer mehr Leute begannen Code beizusteuern. PHP gehört heute zu den beliebtesten Skriptsprachen für das WWW.

**Liegenschaften:**

1. Grundstücke mit ihren wesentlichen Bestandteilen (Sachenrecht).

## 2. Grundstücke und Gebäude (Liegenschaftskataster).

**Liegenschaftskataster:** Die von der Katasterbehörde geführte, aus Büchern und großmaßstäbigen Karten bestehende, nach Gemeinden und Gemarkungen gegliederte öffentliche Einrichtung, in der alle Liegenschaften so nachgewiesen und beschrieben sind, wie es die Bedürfnisse des Rechtsverkehrs, der Verwaltung, der Wirtschaft und der Statistik erfordern. Das Liegenschaftskataster dient als amtliches Verzeichnis der Grundstücke und weist die Ergebnisse der Reichsbodenschätzung nach. In digitaler Form werden ALB und ALK (in Bayern DFK) geführt.

**LINUX:** Unix-ähnliches Betriebssystem für den PC. Linux-Versionen für andere Plattformen sind auch entwickelt worden. Es wird in weltweiter Zusammenarbeit mit anderen Programmierern von dem Finnen Linus Torvalds entwickelt und wurde nach ihm benannt.

Der Quellcode des Betriebssystems Linux steht unter der GNU Public License (GPL), ist also frei erhältlich und darf nach Belieben weiterentwickelt werden. Verschiedene Linux-Distributionen können kostenlos heruntergeladen, bzw. gegen eine vergleichsweise geringe Gebühr, von kommerziellen Anbietern auf CD-ROM erworben werden. Diese Gebühr deckt sowohl die Herstellungskosten der CD-ROMS als auch Zusatzleistungen wie Handbücher und Benutzer-Support.

Linux hat sich bereits als leistungsfähiges Betriebssystem im Server-Bereich erwiesen und setzt sich zunehmend auch im Anwenderbereich als Alternative zu Betriebssystemen wie MS-Windows durch. Es wird mittlerweile auch von zahlreichen namhaften Software-Herstellern unterstützt. Der Einsatz von Linux erfordert jedoch immer noch ein gewisses Maß an technischer Kompetenz.

**MathML:** Mathematical Markup Language; von XML abgeleitete Sprache für mathematische Ausdrücke.

**McCool, Rob:** Entwickler des ersten Webservers überhaupt. Dieser NCSA Webserver war der Vorläufer des Apache Webservers.

**Microsoft:** größter Hersteller von Software; die Firma Microsoft wurde 1975 von Bill Gates und Paul Allen in Redmond (US-Bundesstaat Washington) gegründet. Beide schrieben eine Programmiersprache für den ersten kommerziellen Microcomputer, den MITS Altair. Bekannt / berühmt wurde das Unternehmen durch die weit verbreiteten Betriebssysteme MS-DOS, WINDOWS, Windows für Workgroups, Windows 95 und Windows NT, sowie Standard-Software wie Word für DOS und Windows, sowie Excel. Außerdem bietet Microsoft Entwicklungsumgebungen für eine Vielzahl von Programmiersprachen an.

**MIT/LCS:** Massachusetts Institute of Technology / Laboratory for Computer Science (Boston, USA)

**MOSAIC:** der erste Web-Browser, der dank seiner grafischen Benutzeroberfläche (GUI) große Verbreitung fand. Der Erfolg von Mosaic beruhte auf seiner einfachen Bedienbarkeit. Mosaic wurde 1992 am National Center for Supercomputing Applications (NCSA) der Universität von Illinois (USA) entwickelt, und ist seit 1993 erhältlich. Anfang der neunziger Jahre waren alle gängigen Browser rein textorientiert. Mit Mosaic ließen sich erstmals Grafiken im HTML-Code einer Web-Seite verankern, die der Nutzer nur noch mit der Maus anzuklicken brauchte.

Zur Mosaic-Programmierergruppe am NCSA gehörte Marc Andreessen, der später Präsident der Netscape Communications Corporation wurde.

**MySQL:** relationales Datenbankmanagementsystem, mit dem Daten in Tabellen gespeichert, geändert, gelöscht oder abgefragt werden können. Für die Datenmanipulation verwendet es SQL, eine standardisier-

te Abfragesprache. Im Internet wird die MySQL-Datenbank in der Regel durch Perl- oder PHP-Anwendungen dazu benutzt, Pages auf Grund einer Query zu generieren. So können z. B. Sites im E-Commerce auf MySQL basieren. Auf Grund der Auswahl eines Artikels werden die verfügbaren Informationen aus der Datenbank abgefragt, durch ein Perl-Script aufbereitet und als HTML-Seite angezeigt. Seit Mitte 2000 ist MySQL unter die GNU-GPL gestellt.

**NCSA:** National Center for Supercomputing Applications; das NCSA ist ein Institut an der Universität Illinois (USA), an dem 1992 der Browser MOSAIC entwickelt wurde. MOSAIC, der erste benutzerfreundliche WWW-Browser mit grafischer Oberfläche (GUI), ist ein Vorläufer des Netscape-Browsers Navigator. Außerdem wurde dort der NCSA Webserver entwickelt, der Vorgänger des Apache Webserver.

**Netscape:** Firmenname des Herstellers der gleichnamigen portablen Browsersoftware. Gängiger Browser neben dem MS Internet Explorer. Außerdem erfand das von Marc Andreessen mitgegründete Netscape die Skriptsprache JavaScript.

**Netzwerkschicht:** Die Netzwerkschicht regelt die Weiterleitung der Datenpakete unter Zuhilfenahme der Leitungsschicht bzw. Datensicherungsschicht. Sie übernimmt die Adressierung der Pakete und deren Routing im Netz. Das Protokoll, das dieser Schicht meist zugrunde liegt, ist das Internet-Protocol (IP).

**Nexus:** Von Tim Berners-Lee, dem Erfinder des World Wide Web, entwickelter Browser mit integriertem Editor. Nexus war unter seinem ursprünglichen Namen WorldWideWeb der erste Browser überhaupt.

**Open Source:** Open Source steht für Programme, deren Quellcode veröffentlicht worden ist. Dadurch kann jeder Interessierte das Funktionieren der Software nachvollziehen und Anwendungsprogramme schreiben oder Veränderungen vornehmen. Das erste große Open Source Projekt war Unix. Ein weiteres Beispiel ist Linux.

Nach der Definition der Opensource.org umfaßt dieses Konzept allerdings noch weitere Kriterien für die Lizenzierung von Softwarenutzung. Darunter fallen zum Beispiel das Wegfallen von Beschränkungen bezüglich der Veränderung, der Anwendungsbereiche und der Weiterverbreitung.

Programme, deren Quellcode dagegen geschützt und unter Copyright gestellt ist, nennt man proprietär.

**Patch:** Flickwerk, oft praktizierte Methode zum Reparieren von Fehlern in Software, indem man derartige Patches downloaden kann. Eine hohe Zahl von Patches in einem Programm deutet auf eine unsaubere und schlechte Qualität der Programmierung hin. Meist werden Patches in die Folgeversion integriert bzw. die Fehler beseitigt.

**Parsen:** Vorgang mit dem Parser, der eine Software ist, die den Datenstrom eines Dokumentes analysiert und entsprechend der Syntax aufbereitet. Beim Parsen werden die Informationen des Dokumentes in die Elemente gefiltert, in die die Informationen strukturiert sind.

Bei SGML- und XML-konformen Auszeichnungssprachen, wie HTML oder XHTML, bedeutet dies, daß der Text gemäß den entsprechenden Tags aufbereitet wird, bevor er dann durch die Layout-Engine angezeigt wird.

Parsen wird u. a. auch durchgeführt bei der syntaktischen Prüfung, z. B. bei Validierung oder Kompilierung.

**PDF:** Bezeichnung eines auf PostScript basierenden und vom Hause Adobe ([www.adobe.de](http://www.adobe.de)) entwickelten Dateiformats, um elektronisch formatierte Dokumente speziell im Hinblick auf das Internet speichern und austauschen zu können. Zum Betrachten von PDF-Dateien wird eine spezielle Software wie z.B. der

kostenlose Adobe Acrobat Reader benötigt. PDF wurde vom Hause Adobe aus PostScript heraus entwickelt und um Funktionalitäten zum Definieren von Hyperlinks, Kompression und Verschlüsselung erweitert. PDF-Dateien sind in den seltensten Fällen zur Weiterverarbeitung gedacht. Üblicherweise dient PDF der Definition von elektronisch publizierten Dokumenten in ihrer endgültigen Form.

**PHP:** Hypertext Preprocessor (früher: Personal Homepage); Erweiterung für Internet-Server (ähnlich ASP für Unix und Windows), die es ermöglicht, schnell und mit verhältnismäßig wenig Aufwand dynamische Webseiten für Multimedia- oder E-Commerce-Anwendungen im Internet zu erstellen. Bei PHP handelt es sich um eine serverseitige und in den HTML-Code eingebettete Skriptsprache. Das bedeutet, daß der Client, der die Informationen (z.B. mit einem Browser) anfordert, nie den eigentlichen Skriptcode, sondern nur die resultierende HTML-Seite zu Gesicht bekommt. Bemerkenswert an PHP ist die breite Unterstützung für eine ganze Reihe von SQL-Datenbanken.

**Plug-In:** von engl. „to plug in“, dt.: anschließen, einstecken, einstöpseln; Zusatzprogramm für einen Browser, das dessen Funktionen erweitert. Mit Hilfe dieser Erweiterungs-Software lassen sich besondere Dateiformate lesen bzw. ausführen, die im HTML-Standard nicht vorgesehen sind. Zum Beispiel lassen sich Musik- und Video-Daten abspielen oder Vektorgrafiken betrachten. Wird ein Element auf einer Webseite angeklickt, das mit einem bestimmten Plug-In verknüpft ist, so öffnet sich automatisch die spezielle Anwendung im Browser-Fenster, und die Musik, der Film oder die Multimedia-Sequenz kann abgespielt werden.

**PNG:** Portable Network Graphic Format; das vom World Wide Web Consortium (W3C) entwickelte und als Standard verabschiedete Format ist lizenzfrei und soll GIF- und JPEG-Bilder ablösen - komprimierend und ohne gravierende Qualitätseinbußen.

**POP:** Post Office Protocol; dient dem Zugriff auf Server, die E-Mails für andere Rechner bereithalten. Es existieren die Versionen POP, POP2 und POP3, die nicht untereinander kompatibel sind.

**Quellcode:** Originärer ASCII-Text, der ein Programm in einer höheren Programmiersprache darstellt und mit einem Editor bearbeitet werden kann. In dieser Form ist das Programm nicht lauffähig und muß erst mit einem Compiler in die Maschinensprache übersetzt werden (auch Source-Code).

**RealPlayer:** Browser Plug-In zur Darstellung multimedialer WWW-Inhalte in den Formaten RealAudio und RealVideo. Das Plug-In wird sowohl in einer kostenlosen als auch in einer kostenpflichtigen "Plus"-Version angeboten. Es handelt sich um einen Quasi-Standard für Audio-Video-Inhalte im Web, das u. a. auch kompatibel zum SMIL-Konzept ist.

**Server:** 1. Ursprünglich Bezeichnung für den zentralen Computer eines Netzwerks (network), samt der entsprechenden Software (u. a. Netzwerkbetriebssystem), der seine Leistungen und Daten den am Netzwerk teilnehmenden Computern (client) mittels Client-Server-Software (client-server) zur Verfügung stellt.

2. Darüber hinaus werden auch bestimmte Service-Einrichtungen im Internet bzw. Software, die von der Funktion her Dienstleistungen wie Datenbanksuche (database) etc. erfüllt, mittlerweile als "Server" bezeichnet.

**Shape:** plattformunabhängiges Vektordatenformat der Firma ESRI ([www.esri.com](http://www.esri.com)), die mit der Software ArcView eines der meist verbreiteten Desktop-GIS vertreibt.

**SMIL:** Synchronized Multimedia Integration Language; Multimedia-Standard für das Internet, der vom W3C beschlossen wurde und an dessen Entwicklung Unternehmen wie Microsoft, DEC und Philips beteiligt sind. SMIL ermöglicht es, Audio- und Videosequenzen verschiedener Quellen simultan abzuspielen. Zudem soll mit SMIL künftig die Bandbreite von Videodaten auf den Umfang bisheriger "low bandwidth media" reduziert werden.

**SMTP:** Simple Mail Transfer Protocol; Standard für den Austausch von E-Mails zwischen Servern im Netzwerk. E-Mail-Clients benutzen SMTP nur um E-Mails an einen Server zu schicken - nicht aber zum Empfangen.

**SQD:** Sequentiell Data Format; Datenformat für den Austausch von Vektordaten. Es besteht aus einer Reihe von Befehlen im ASCII-Format und ist für große kartographische Datenmengen geeignet. Standardformat für Siemens-CAD-Anwendungen (SICAD).

**SVG:** Scalable Vector Graphics; vom World Wide Web Consortium als „W3C Recommendation“ freigegebene XML-Erweiterung zur Beschreibung von zweidimensionalen Vektorgrafiken und Animationen, die in Webseiten eingebunden werden können. SVG berücksichtigt drei Arten von grafischen Objekten:

- vektorbasierende Geometrie (z.B. Linien und Kurven)
- Pixelbilder und
- Text

SVG liefert im Gegensatz zu JPEG oder GIF als Vektorformat kleine Dateigrößen, skalierbare Darstellungen, eine unbegrenzte Auswahl aus Farben und Schriften, präzise Layout- und Designoptionen, Animationen und hochaufgelöste Ausdrücke aus dem Webbrowser. Die SVG-Bildeffekte, die vom einfachen Weichzeichnen bis hin zur ereignisgesteuerten Animation reichen, ermöglichen es Benutzern, Webgrafiken und interaktive Bilder zu erstellen, die sowohl umfangreiche visuelle Inhalte als auch eine kurze Download-Zeit bieten.

**TELNET:** Kategorie von Programmen, die ähnlich einem Terminal-Programm, dem Benutzer direkten Zugriff auf einen anderen Computer im Internet ermöglichen. Wenn man die entsprechenden Zugriffsrechte auf diesen Rechner hat, kann man aus der Ferne Programme starten, Dateien bearbeiten oder auch hinauf- und herunterladen.

**TIFF/TIF:** Tagged Image File Format; standardisiertes Austauschformat für Rasterdaten und beinhaltet verschiedene Rasterformate. Man darf daher streng genommen nicht von dem TIF-Format sprechen, sondern muß zusätzlich die verwendete Speicherung angeben (z.B. packed bit, CCITT, etc.). Außerdem verwendet TIFF eine Reihe von Parametern (Tags), die Angaben über die Art der Speicherung, die Anzahl der Zeilen und Spalten, die Auflösung etc., enthalten. Nicht enthalten sind jedoch Angaben zur Georeferenzierung. Diese Information muß separat mitgeliefert werden bzw. gehört zum neuen Standard GeoTIFF.

**Transportschicht:** Teil der Infrastruktur eines Rechnernetzes, der zuständig ist für alles, was mit den Aufgaben des Datentransfers zu tun hat, wie Durchführung von Fehlerkontrollen, Neuordnung von Paketfolgen, Bearbeitung von Wiederholungsanforderungen untergeordneter Systeme und Zerlegung von Nachrichten in Einzelpakete.

**UNIX:** Uniplexed Information and Computing System; im Internet und auf Großrechnern weit verbreitetes Betriebssystem. UNIX ist eine Weiterentwicklung von Multics (Multiplexed Information and Computing Service/System), einem interaktiven Mehrbenutzersystem, das in den Sechzigerjahren von AT&T

entwickelt wurde. In den Siebzigern war UNIX das erste Betriebssystem, das in einer höheren Programmiersprache, nämlich "C", geschrieben wurde. Es ist weitgehend plattformunabhängig und multiuser- sowie multitasking-fähig.

**URI:** Uniform Resource Identifier; dient zur eindeutigen Identifizierung eines bestimmten Inhalts - Text, Grafik, Animation oder Audio, die "Ressource" - im Internet. In den meisten Fällen fungiert die URL als URI, in Form einer Website-, E-Mail-, Newsgroup-, oder FTP-Adresse. Eine andere Form ist ein URN (Uniform Resource Name), der einem bestimmten Gehalt, etwa einer Webseite, zugeordnet wird und konstant bleibt, auch wenn sich die URL ändert.

**URL:** Uniform Resource Locator; weil Objekte im Web - inklusive Dokumente, Bilder und andere Dienste - weltweit auf Tausende von Computern verteilt sind, ist ein konsistenter Weg notwendig, um diese zu lokalisieren. Die URL ist ein Standardformat, um jedem Objekt eine eindeutige Adresse zu geben und wird verwendet, um alles im Web sowie in anderen Internet-Services wie FTP auffinden zu können. Obwohl Variationen des Formats häufig auftreten, besteht eine URL aus vier grundsätzlichen Teilen:

- dem Protokoll, wie HTTP, FTP oder mailto
- der IP-Adresse oder Namen des Servercomputers, wie //www.BGRUNDviewer.de/
- dem Verzeichnis auf dem Server in dem das Objekt gefunden werden kann (in UNIX-Format), wie /demo/
- und dem Namen des Objekts selbst (normalerweise ein Dateiname), wie alk.svg.

Zusammen würden diese Teile wie folgt geschrieben:

<http://www.BGRUNDviewer.de/demo/alk.svg>

**Vektorgrafik:** Art der Verwaltung von Grafikdaten durch mathematische Funktionen (Gegenteil von pixelorientiert / Pixel-Grafik) - eine Linie wird durch zwei Punkte beschrieben, ein Kreis durch die Koordinaten des Mittelpunktes und seinen Radius usw. Jeder einzelne Bildpunkt ist Bestandteil eines so beschriebenen Elementes.

**Viola:** Einer der ersten Browser mit grafischer Oberfläche (1992)

**VML:** Vector Markup Language; von Microsoft und Macromedia entwickelte Erweiterung von XML, die ein Format zur Kodierung von Vektor-Informationen im Internet definiert. Zudem wird beschrieben, wie diese Informationen darzustellen und zu editieren sind.

**W3C:** World Wide Web Consortium; von verschiedenen mit dem Internet eng verbundenen Firmen und Konzernen gegründeter Interessenverband, unter der Leitung des Laboratory for Computer Science am Massachusetts Institute of Technology (MIT/LCS) in Cambridge, Massachusetts. Das Konsortium fördert Standards und die Interoperabilität von World Wide Web-Produkten. Ursprünglicher Sitz des Konsortiums war die Europäische Organisation für Kernforschung (CERN) in Genf, dort, wo die Technologie des World Wide Webs entwickelt wurde. Bisher konnte das Konsortium bei der Förderung der Zusammenarbeit im Bereich der Webtechnologien bei einer Anzahl von Privatfirmen, die ihre Geheimnisse oft nur widerwillig teilen, bescheidene Erfolge verbuchen.

**WAMP:** Kombination aus Windows-Betriebssystem, Apache-Webserver, MySQL-Datenbanksystem und PHP-Skripten

**Webmaster:** Verwalter einer Webseite

**Webserver:** Server, der HTML-Dokumente und andere Internet / Intranet-Ressourcen speichert und über HTTP versendet bzw. entgegennimmt (auch HTTP-Server genannt).

**Windows:** Betriebssystem von Microsoft; erstmals 1985 vorgestellt und 1990 mit der Version 3.0 marktfähig. Es brachte die bereits von Apple angewandte grafische Benutzeroberfläche mit Fenstertechnik und Maussteuerung, sowie Multitasking in die Microsoft-Welt. Windows war zunächst als Zusatz zum alten Betriebssystem MS-DOS gedacht, konnte sich später aber zu einem „echten“ Betriebssystem entwickeln. Mit Millionen-Aufwand führte Microsoft 1995 „Windows 95“ ein, darauf folgte 3 Jahre später „Windows 98“ und 2000 „Windows Me“. Für Unternehmens-Netzwerke stehen die Varianten „Windows NT“ (ab 1993), die Version „NT 4.0“ (ab 1997) und „Windows 2000“ zur Verfügung. Mit der Einführung von „Windows XP“ im Jahre 2001 wurden beide Produktlinien zusammengeführt.

**WLDGE:** Workdatei Liegenschaftsbuch Daten-Gewinnung mit Entschlüsselungen; Standardausgabeschnittstelle des Automatisierten Liegenschaftsbuches (ALB).

**WML:** Wireless Markup Language; In XML definierte Beschreibungssprache zur Darstellung von Internet-/WAP-Inhalten auf mobilen Kleinstgeräten wie Handys oder PDAs.

**WWW:** World Wide Web; englische Bezeichnung für "weltweites Netz". Gemeint ist der jüngste Dienst im Internet, der sich durch hohe Benutzerfreundlichkeit sowie multimediale Elemente auszeichnet und zur Verbreitung des Internets massiv beigetragen hat. Nicht selten wird das "World Wide Web" mit dem Internet gleichgesetzt, aber tatsächlich ist es nur eine Untermenge - also einer von mehreren Diensten. Der Zugriff auf die Informationen erfolgt über WWW-Browser - Netscape, Mozilla, Mosaic, Opera, Konqueror oder Internet Explorer, um nur ein paar zu nennen. Browser sind Programme, die allerlei Informationen in einem multimedialen vernetzten Hypertext-System finden.

**XLink:** XML-Linksprache, die es erlaubt, Elemente in XML-Dokumente einzufügen, um Links zwischen Ressourcen zu erzeugen und zu beschreiben. Die Sprache nutzt XML-Syntax um Strukturen zu erzeugen, die Links beschreiben können, die den einfachen unidirektionalen Hyperlinks des heutigen HTML ähneln, sowie noch ausgefeiltere Links.

**XSL:** Extensible Style Language; dient der Formatierung von XML-Dokumenten.

## Literaturverzeichnis

- **Flanagan, David: JavaScript – Das umfassende Referenzwerk** (1997), O'Reilly Verlag, ISBN 3-930673-56-8
- **Graf, Hagen: Jetzt lerne ich Apache 2** (2002), Markt + Technik Verlag, ISBN 3-8272-6404-9
- **Zandstra, Matt: Jetzt lerne ich PHP 4** (2001), Markt + Technik Verlag, ISBN 3-8272-5883-9
- **Born, Günter: Jetzt lerne ich XML** (2001), Markt + Technik Verlag, ISBN 3-8272-5924-X
- **Fibinger, Iris: SVG – Scalable Vector Graphics** (2002), Markt + Technik Verlag, ISBN 3-8272-6239-9
- **Salathé, Marcel: SVG Scalable Vector Graphics** (2001), Markt + Technik Verlag, ISBN 3-8272-6188-0

## Webseiten

- Fachhochschule Stuttgart – Hochschule für Technik  
<http://www.fht-stuttgart.de/>

## Die Seite überhaupt für Internet, WWW, HTML, CSS und JavaScript

- SELFHTML  
<http://selfhtml.teamone.de/>

## Das Internet und das World Wide Web

- Die Entstehung des Internets  
<http://www.allmystery.de/technik/internet/internet.shtml>
- Der Hintergrund von HTML: Die Ziele von Tim Berners-Lee  
<http://www.pms.informatik.uni-muenchen.de/lehre/seminar/html-metamorphosen/00ss/ausarbeitungen/Hintergrund/historisch.html>
- Entstehung des Internet  
<http://www.mbg.wn.bw.schule.de/hirni/a40/inet.htm>
- Geschichte des Internet  
[http://www.webdesign-referenz.de/internet\\_geschichte.html](http://www.webdesign-referenz.de/internet_geschichte.html)
- Eine kurze Geschichte des Internet  
[http://www.dfn-expo.de/Geschichte/Geschichte\\_Internet.html](http://www.dfn-expo.de/Geschichte/Geschichte_Internet.html)
- Entstehung und Zukunft des Internet und des World Wide Web  
<http://www.hist.unizh.ch/gs+edv/ss2000/geschichte.htm>
- Geschichte des Internet  
<http://www.geschichte-des-internet.com/>
- Die Geschichte des Internet und die Entwicklung seiner Benutzerstruktur  
<http://www.phil.uni-sb.de/fr/infowiss/papers/iwscript/geschint/>
- Geschichte des Internet  
<http://land.heim.at/yellowstone/230700/Internet.html>
- Das World Wide Web  
[http://www.gatetoweb.at/webdesign/internet\\_geschichte/das\\_world\\_wide\\_web.htm](http://www.gatetoweb.at/webdesign/internet_geschichte/das_world_wide_web.htm)
- Internet Protokoll Familie  
<http://de.wikipedia.org/wiki/TCP/IP>
- Datennetze  
[http://www.i-m.de/home/datennetze/aw\\_start.htm](http://www.i-m.de/home/datennetze/aw_start.htm)

## Apache Webserver

- The Apache Software Foundation  
<http://www.apache.de/>
- Web Server Survey  
[http://news.netcraft.com/archives/2003/04/13/april\\_2003\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2003/04/13/april_2003_web_server_survey.html)
- Web Server Survey  
[http://www.securityspace.com/s\\_survey/data/200305/index.html](http://www.securityspace.com/s_survey/data/200305/index.html)

- Server-Dienste  
<http://www.openoffice.de/linux/buch/server.html>
- Apache-Service  
<http://www.apache-service.de/>

## HTML – Hypertext Markup Language

- W3C – HyperText Markup Language  
<http://www.w3.org/MarkUp/>
- SELFHTML  
<http://selfhtml.teamone.de/>

## CSS – Cascading Style Sheets

- W3C- Cascading Style Sheets  
<http://www.w3.org/Style/CSS/>

## JavaScript

- Geschichte  
<http://home.pfaffenhofen.de/internetschule/javascript/referenz/geschichte.html>
- JavaScript-Informationen  
<http://www.mintert.com/javascript/>

## PHP – PHP: Hypertext Preprocessor

- Die Seite für PHP  
<http://www.php.net/>
- PHP-Center  
<http://www.php-center.de/>
- Die Geschichte von PHP  
<http://www.fh-wedel.de/~si/seminare/ws99/Ausarbeitung/lamp/lamp1.htm#diegeschichtevonphp>
- PHP: eingebettete Skriptsprache  
<http://www.heise.de/ix/artikel/1999/07/076/03.shtml>
- Die Geschichte von PHP  
<http://www.hacke.net/php/geschichte.php>
- PHP Magazin – die kommende Zend 2 Engine  
[http://www.phpmag.de/itr/online\\_artikel/psecom,id,207,nodeid,62,ps\\_lo,60.html](http://www.phpmag.de/itr/online_artikel/psecom,id,207,nodeid,62,ps_lo,60.html)

## MySQL – Datenbankmanagementsystem

- Die offizielle Seite  
<http://www.mysql.de/>
- MySQL, phpMyAdmin und PHP  
[http://v.hdm-stuttgart.de/~riekert/lehre/php/mysql\\_php.html#php\\_kontroll](http://v.hdm-stuttgart.de/~riekert/lehre/php/mysql_php.html#php_kontroll)
- MySQL im Griff  
[http://www.networkcomputing.de/workshop/ws2\\_0701.html](http://www.networkcomputing.de/workshop/ws2_0701.html)

- The phpMyAdmin Project  
<http://www.phpmyadmin.net/>

## XML – Extensible Markup Language

- W3C – Extensible Markup Language  
<http://www.w3.org/XML/>
- XML-Tutorial  
<http://www.aboutwebdesign.de/awd/content/1024415102.shtml>
- Extensible Markup Language  
<http://www.edition-w3c.de/TR/1998/REC-xml-19980210.html>
- XML-Einführung  
[http://informatik.hsr.ch/Content/Gruppen/Stud/Faecherkatalog/Infosem/Vortraege/WS0001/Resultate%5CXML\\_Einfuehrung\\_skript.pdf](http://informatik.hsr.ch/Content/Gruppen/Stud/Faecherkatalog/Infosem/Vortraege/WS0001/Resultate%5CXML_Einfuehrung_skript.pdf)
- XML-Tutorial  
<http://www.sachen-fuer-webmaster.de/artikel/92.html>
- Tutorial: XML  
[http://www.uzi-web.de/xml/xml\\_toc.htm](http://www.uzi-web.de/xml/xml_toc.htm)
- XML-Tutorial  
[http://www.better-office.com/Deutsch/Produkte/Entwicklung/boXML\\_Tools/XML\\_minus\\_Info/XML\\_minus\\_Tutorial.htm](http://www.better-office.com/Deutsch/Produkte/Entwicklung/boXML_Tools/XML_minus_Info/XML_minus_Tutorial.htm)
- Tutorials  
<http://www.jcpohl.de/technik/tecindex.html>

## SVG – Scalable Vector Graphics

- W3C – Scalable Vector Graphics  
<http://www.w3.org/Graphics/SVG/Overview.htm#8>
- Adobe SVG Zone  
<http://www.adobe.com/svg/>
- Kartendarstellungen mit SVG  
<http://www.carto.net/>
- SVG-Tutorial  
<http://www.aboutwebdesign.de/awd/content/1023384263.shtml>
- SVG-Tutorial  
<http://www.heise.de/ix/artikel/2002/12/052/>
- SVG-Tutorial  
<http://svg.tutorial.aptico.de/>
- SVG-Tutorial  
<http://www.karto.ethz.ch/td/tutorial/k2.html>
- Einführung in SVG  
<http://v.hdm-stuttgart.de/~sailer/SS03/Donnerstag/svg1.html>
- SVG-Tutorial  
<http://www.nls.at/~res/svg/svg01.htm>
- Scalable Vector Graphics – Gestaltung einer interaktiven Karte  
[http://www.carto.net/papers/alex\\_berger/AlexBergerStudienarbeit.pdf](http://www.carto.net/papers/alex_berger/AlexBergerStudienarbeit.pdf)
- SVG – Scalable Vector Graphics  
[http://www.carto.net/papers/svg/articles/paper\\_ugra\\_zurich\\_2001.pdf](http://www.carto.net/papers/svg/articles/paper_ugra_zurich_2001.pdf)

- SVG – Skalierbare Vektorgraphik für das Web  
<http://www.fz-juelich.de/zam/files/docs/vortraege/SVG.pdf>
- Vektorbasierte Bildformate  
[http://www-ra.informatik.uni-tuebingen.de/lehre/ws01/pro\\_internet\\_ausarbeitung/proseminar\\_speidel\\_ws01.pdf](http://www-ra.informatik.uni-tuebingen.de/lehre/ws01/pro_internet_ausarbeitung/proseminar_speidel_ws01.pdf)

## **Liegenschaftskataster**

- Landesvermessungsamt Baden-Württemberg  
<http://www.lv-bw.de/lvshop2/index2.asp>

## **Online-Lexika**

- ARCHmatic-Glossar und -Lexikon  
<http://www.glossar.de/glossar/index.htm>
- Netlexikon  
<http://netlexikon.akademie.de/query>
- Computerlexikon  
<http://www.computerlexikon.com/>
- Informations-Archiv  
<http://www.informationsarchiv.net/>
- Das Internet-Wörterbuch  
<http://www.networds.de/>
- Internet-Manual  
<http://www.internet-manual.de/index.htm>
- Geoinformatik-Lexikon  
<http://www.geoinformatik.uni-rostock.de/lexikon.asp>

## Erklärung

Die vorliegende Diplomarbeit wurde von mir zur Diplomprüfung im Sommersemester 2003 selbst verfaßt und ohne fremde Hilfe angefertigt. Die benützten Hilfsmittel und Literaturquellen sind in den entsprechenden Verzeichnissen aufgeführt.

Stuttgart, den \_\_\_\_\_

\_\_\_\_\_  
(Thomas Mailänder)

Gesehen:

Erstprüfer und Betreuer:

\_\_\_\_\_  
(Datum)

\_\_\_\_\_  
(Prof. Dr.-Ing. Franz-Josef Behr)

Zweitprüfer:

\_\_\_\_\_  
(Datum)

\_\_\_\_\_  
(Prof. Dr.-Ing. Hardy Lehmkuhler)

## Stichwortverzeichnis

- Advanced Research Projects Agency 14
- ALKIS 76, 124
- Amtliches Festpunktinformationssystem
  - AFIS 73
- Amtliches Liegenschaftskataster Informationssystem
  - ALKIS 73
- Amtliches Topographisch-kartographisches Informationssystem
  - ATKIS 73
- Andreessen, Marc 20, 124
- Anwendungsschicht 18
- Apache 21, 124
- Apache Software Foundation 22
- ARPA 14
  - ARPA-Net 14
- Aufnahmepunkte 98
- Ausgestaltungen 104, 107
- Automatisierte Liegenschaftskarte 74, 124
  - ALK 10, 74, 124
- Automatisiertes Liegenschaftsbuch 74
  - ALB 74
- Axmark, David 36
- Backbone 15
- Bakken, Stig Sæther 33
- Berners-Lee, Tim 19, 125
- Bézierkurve 55, 101, 125
- BGRUND 13, 73, 79, 125
- BGRUNDviewer 12, 77, 121
- Bina, Eric 20, 125
- Binary Version 22
- Bitmap
  - BMP 44
- Bray, Tim 40
- Browser 19, 126
- Bug 126
- Caraveo, Shane 33
- Cascading Style Sheets 25
  - CSS 25
- Cerf, Vinton 17, 126
- CERN 19, 126
- Client 18, 126
- Compiler 126
- Container 113
- CSV 74
- Datenschutz 117
- DENIC 17
- DNS 16, 127
- Document Object Model 27
  - DOM 127
- Document Type Definition
  - DTD 41
- Domain Name Service 16
- Domäne 17
- Drawing Exchange Format
  - DXF 75
- Dynamic HTML 27
  - DHTML 27
- ECMAScript 27
- Eich, Brendan 26
- Ein- und Ausblenden von Ebenen 113
- Ellipse 50, 55
- Enquire 19, 127
- Erwise 20, 127
- EuropaNET 15
- Extended Style Sheet Language
  - XSL 41
- Extensible Hypertext Markup Language
  - XHTML 23, 41
- Extensible Markup Language
  - XML 23, 39
- Filtereffekte 65, 69
- Flächenberechnung 118
- Flash 44
- Flurstück 85
- Flurstücksliste 86
- Flurstücksnummer 102
- FTP 18, 127
- Gateway 19
- Gebäude 73, 74, 77, 86, 104
- Gebäudeliste 88
- Gebäudenutzung 86
- Genauigkeitsstufe 81
- Generalized Markup Language 40
  - GML 40
- GEODIS 73, 77

- Geography Markup Language 40
- Goldfarb, Charles 39
- Graphics Interchange Format
  - GIF 44
- Grenzzug 85
- Gruppe 60
- Gutmanns, Andi 33
- Hash-Array 36
- Hausnummer 107
- Host 128
- HTML 19, 23, 128
- HTTP 18, 19, 129
- Hyper Graphics Markup Language 44
  - HGML 44
- Hypertext 19
- IIS 22
- Insellfläche 115
- Internet 14, 129
- Internet Explorer 20
- Internet Protokoll 18
- Internet-Information-Server 22
- Intranet 129
- IP-Adressen 16
- Java Server Pages 23
- Joint Photographic Expert Group
  - JPEG 44
- JScript 27, 129
- Kahn, Robert 17, 130
- Katasterkartenvorschrift 11, 104
- Koordinatensystem 74, 92
- Kreis 50
- Kreisbogen 84, 100
- LAMP 22, 130
- Larsson, Allan 36
- Lerdorf, Rasmus 32
- Liegenschaftsbuch 73, 124
- Liegenschaftskataster 73, 131
- linearer Verlauf 65
- Linie 51
- Linienzug 84, 100
- LINUX 131
- Liste der topographischen Objekte 89
- LiveScript 26
- Mauszeigerposition 118
- McCool, Rob 21, 131
- Microsoft 20, 131
- MILNET 15
- MIME 53
- MIT Laboratory for Computer Science 20
- MIT/LCS 20, 131
- MOSAIC 20, 131
- Mozilla 27
- MySQL 36, 101, 131
- Namensraum 47
- National Center for Supercomputing
  - Application 20
- National Science Foundation 14
  - NSFNet 15
- NCP 14
- NCSA 20, 21, 132
- Neigung 62
- Netscape 20, 132
- Netscape Navigator 20
- Netzwerkschicht 18
- Nexus 19, 132
- Nicht numerierte Punkte 98
- Nutzungsfläche 86
- Open Source 22, 132
- Open-Gis-Consortium 40
  - OGC 40
- Painter's Algorithm 51
- Patch 21, 132
- PDF 74
- Pfad 54
- PHP 32, 133
- php.ini 80, 101
- phpMyAdmin 37
- Plug-In 12, 45, 133
- Polygon 52
- Polylinie 52
- Portable Network Graphic Format
  - PNG 44
- Precision Graphics Markup Language 44
  - PGML 44
- preserveAspectRatio 49
- Punktnummer 81
- Quellcode 133
- radialer Verlauf 65
- RAND-Corporation 14
- RealPlayer 45
- Rechteck 50

- relationales Datenbank-Managementsystem
  - 37
  - RDBMS 37
- Root-Element 47
- Rotation 55, 62
- Routing 19
- Scalable Vector Graphics 44
- Schnitt- und Kleinpunkte 98
- Schraffur 87, 105
- Segal, Ben 19
- Sequentiell Data Format
  - SQD 75
- Server 18, 133
- Shape 75
- Skalierung 62
- SMIL 41, 70, 134
- Sourcecode 22
- Spannmaße 118
- Sputnik 14
- Standard Generalized Markup Language
  - 40
  - SGML 23, 40
- Suraski, Zeev 33
- SVG 44
- SVG Viewer 12, 45, 78
- Symbole 62
- Tagged Image File Format
  - TIFF 44, 134
- Tatsächliche Nutzungen 86
- TCP/IP 14, 17
- Textabschnitt 59
- Texte 59, 89, 109
- Texteliste 90
- Topographische Objekte 88, 108
- Translation 62
- Transmission Control Protocol 17
- Transportschicht 18
- Trigonometrische Punkte 98
- Universal Resource Identifier 19
- UNIX 22, 134
- URI 19, 135
- UTM 76
- Vector Markup Language 44
  - VML 41, 44, 135
- Verschieben des Kartenausschnittes 110
- Versicherungspunkte 82, 98
- ViewBox 48, 92, 110
- Viewport 47
- Viola 20, 135
- Vordaten 81, 95
- W3C 20, 135
- WAMP 22, 135
- Webmaster 135
- Widenius, Michael 36
- Windows 22, 136
- WinMySQLadmin 38
- Winstead, Jim 33
- Wireless Markup Language
  - WML 41
- WLDGE 74
- World Wide Web 19, 136
- Zend 33
- Zoomen des Kartenausschnittes 111