

# **Datenbankgestützte kartographische Visualisierung im Internet**

Anwendungen im großen Maßstab mittels Mapserver-Systeme

DIPLOMARBEIT

zur Erlangung des Magistergrades der  
Naturwissenschaften  
an der Fakultät der Human- und Sozialwissenschaftlichen  
der Universität Wien  
Studienrichtung Geographie  
Studienzweig Kartographie

eingereicht von  
PUCHER Alexander

Wien, Juni 2001

## **Danksagung**

Mein aufrichtiger Dank geht an ao. Univ.-Prof. Dr. Ingrid Kretschmer sowie Univ. Ass. Dr. Karel Kriz für die fachliche Unterstützung im Rahmen der Erstellung dieser Arbeit.

Herzlichen Dank allen Kollegen der Computer-Kartographie, die in den vergangenen Jahren mitgeholfen haben, den anfangs anonymen Studienbetrieb zu einem wirklichen Erlebnis zu machen. Ich erspare es mir an dieser Stelle, alle Namen persönlich anzuführen. Ihr wisst, wer ihr seid. Dennoch möchte ich es nicht verabsäumen, speziell den Mitgliedern des „OSI-Teams“ - Christian Fürpaß, Mag. Christian Resch und Robert Saul - meinen besonderen Dank auszusprechen. Ohne eure Mithilfe wären meine Nächte noch kürzer gewesen!

Danke an meine Vater für seine jahrelange Unterstützung.

Mein größter Dank aber gilt einer Person, die nie ein Wort dieser Arbeit gelesen hat, ohne die ich das alles aber nicht geschafft hätte - Nathalie. Ich liebe dich.

Für meinen Sohn David, der von alledem noch nichts weiß.

# Inhaltsverzeichnis

<b>Danksagung</b> .....	<b>2</b>
<b>1 Einleitung</b> .....	<b>7</b>
<b>2 Datenbanksysteme</b> .....	<b>12</b>
2.1 Begriffsbestimmung.....	12
2.1.1 Datenbank.....	12
2.1.2 Datenbankmanagementsystem.....	12
2.1.3 Datenbanksysteme.....	13
2.2 Datenspeicherstrukturen.....	14
2.2.1 Datei-Systeme.....	14
2.2.2 Datei-Verwaltungs-Systeme.....	15
2.2.2 Moderne Managementsysteme.....	16
2.3 Logische Datenmodelle .....	17
2.3.1 Hierarchisches Datenmodell.....	18
2.3.2 Netzwerkmodell.....	18
2.3.3 Relationales Datenmodell .....	19
2.3.4 Objektrelationales Datenmodell.....	20
2.3.5 Objektorientiertes Datenmodell.....	20
2.4 Datengrundlagen.....	21
2.4.1 Geometriedaten.....	21
2.4.1.1 Vektordaten.....	23
2.4.1.2 Rasterdaten.....	24
2.4.2 Sachdaten.....	24
2.4.3 Metadaten.....	25
2.5 Datenqualität, -kontrolle.....	27
2.5.1 Positionsgenauigkeit.....	29
2.5.2 Attributgenauigkeit.....	29
2.5.3 Logische Konsistenz.....	30
2.6 Bearbeitung von Datenbanken.....	31
2.6.1 Datenbanksprachen.....	32
2.6.1.1 Standard Query Language (SQL).....	33
2.6.2 SQL in der Praxis.....	34
2.6.2.1 Manipulation von Sachdaten.....	34
2.6.2.2 Manipulation von Geodaten.....	36
2.6.3 Die OpenGIS@Consortium-SQL92 Erweiterung.....	39
<b>3 Kartographie</b> .....	<b>42</b>
3.1 Kartographische Visualisierung im Wandel.....	42

3.2 Kartographische Kommunikation im Wandel.....	44
3.2.1 Lineares Modell .....	44
3.2.2 Dialogisches Modell .....	45
3.2.3 Monologisches Modell .....	46
3.3 Vom Geomodell zum kartographischen Modell.....	48
3.3.1 Geomodell.....	48
3.3.2 Thematisches Modell.....	50
3.3.3 Kartographisches Modell.....	51
3.4 Digitale Geodatenmodelle in Österreich.....	52
3.4.1 Bundesamt für Eich- und Vermessungswesen.....	52
3.4.1.1 Digitales Landschaftsmodell (DLM).....	52
3.4.1.2 Digitale Katastralmappe (DKM).....	54
3.4.2 Stadt Wien.....	55
3.4.2.1 Räumliches Bezugssystem Wien (RBW).....	55
3.4.2.2 Mehrzweckstadtkarte (MZK).....	56
3.5 Digitale Stadtkartographie.....	57
3.5.1 Möglichkeiten.....	58
3.5.2 Anforderungen.....	59
<b>4 Internet.....</b>	<b>61</b>
4.1 Chronologie.....	61
4.2 Bedeutung des Internet.....	63
4.3 Generierungstechniken.....	64
4.3.1 Statische Internetinhalte.....	64
4.3.2 Interaktive Internetinhalte.....	65
4.3.2.1 Serverseitige Technologien.....	65
4.3.2.2 Clientseitige Technologien.....	67
<b>5 Datenbankgestützte kartographische Visualisierung im</b>	
<b>    Internet .....</b>	<b>69</b>
5.1 Kartographische Ausdrucksformen im Internet .....	69
5.2 Datenbankgestützte kartographische Visualisierungssysteme.....	71
5.2.1 Anforderungen.....	71
5.2.2 Architektur.....	73
5.2.2.1 1-Schicht Modell (Monolithische Applikationen).....	74
5.2.2.2 2-Schicht-Modell.....	74
5.2.2.3 3-Schicht-Modell.....	74
5.2.3 Gliederung.....	76
5.2.3.1 Einteilung nach technischen Gesichtspunkten.....	76
5.2.3.2 Einteilung nach Anforderungen.....	77

<b>6 Datenbankgestützte Mapserver-Systeme .....</b>	<b>79</b>
6.3 Komponenten.....	79
6.3.1 Datenbanksystem.....	79
6.3.1.1 Relationale Systeme.....	80
6.3.1.2 Objektrelationale, -orientierte Systeme.....	80
6.3.2 Datenbankschnittstelle.....	81
6.3.3 Mapserver.....	82
6.3.4 Webserver.....	83
6.3.5 Benutzeroberfläche.....	84
6.4 Funktionsweise.....	85
<b>7 Praktische Umsetzung eines datenbankgestützten</b>	
<b>Mapserver-Systems im Internet .....</b>	<b>88</b>
7.1 Intention .....	88
7.2 Daten.....	89
7.2.1 Datengrundlagen.....	89
7.2.1.1 Sachdaten.....	89
7.2.1.2 Geometriedaten.....	90
7.2.2 Datenaufbereitung.....	92
7.2.2.1 Sachdaten.....	92
7.2.2.2 Geometriedaten.....	93
7.3 Technische Umsetzung eines Prototypen.....	96
7.3.1 Verwendete Komponenten.....	96
7.3.1.1 Datenbankmanagementsystem - MySQL.....	97
7.3.1.2 Mapserver - UMN MapServer.....	97
7.3.1.3 Benutzerschnittstelle - HTML.....	98
7.3.1.4 Datenbankschnittstelle - PHP 3.....	99
7.3.1.5 Webserver (mit Erweiterungen) - APACHE (mit PHP 3).....	99
7.4 Aufbau der Systemarchitektur.....	99
7.4.1 Physischer Aufbau.....	100
7.4.1.1 Konfiguration des UMN MapServers.....	100
7.4.1.2 Verknüpfung der serverseitigen Systemkomponenten .....	102
7.4.2 Funktionsweise des Prototyps.....	103
7.4.2.1 Suche in Datenbank -> Visualisierung in Karte.....	105
7.4.2.2 Suche in Karte -> Informationen aus Datenbank.....	110
<b>8 Zusammenfassung .....</b>	<b>115</b>
<b>Literaturverzeichnis.....</b>	<b>117</b>
<b>Verzeichnis der Abkürzungen.....</b>	<b>124</b>
<b>Abbildungsverzeichnis.....</b>	<b>125</b>

<b>Tabellenverzeichnis.....</b>	<b>126</b>
<b>Programmcode.....</b>	<b>127</b>
UMN MapServer Konfigurationsdatei ("map file").....	127
PHP - Template Datei.....	132
PHP-Skript: Datenbankschnittstelle .....	143

# 1 Einleitung

Kartographische Visualisierungen im Internet sind beinahe so alt wie das Internet selber. Bis vor wenigen Jahren handelte es sich hierbei vor allem um statische Abbildungen analoger Karten. Technische Limitierungen - etwa die verfügbare Bildschirmfläche oder die geringe graphische Auflösung gegenüber drucktechnisch hergestellten Produkten offenbarte jedoch bald die ungenügende Qualität solcher Lösungen. Die Qualitäten des Internet und hier vor allem die Möglichkeiten des raschen Informationsaustausches sowie der Interaktion zeigen sich demnach nicht so sehr in statischen Abbildungen, sondern vielmehr in der dynamischen Erstellung kartographischer Inhalte zur Laufzeit.

Systeme, die dem Nutzer die individuelle Gestaltung kartographischer Darstellungen ermöglichen, verlangen nach einer geordneten Datengrundlage, um die zu visualisierenden Daten exakt ansprechen zu können. Datenbanksysteme bieten diese Möglichkeit der strukturierten Speicherung großer Datenmengen.

Die Bedeutung von Datenbanksystemen in den Bereichen Kartographie und Geoinformation ist in den letzten Jahren stetig angestiegen. Einhergehend mit dieser wachsenden Bedeutsamkeit erweiterten sich auch die Möglichkeiten, datenbankunterstützte Systeme kartographisch zu nutzen. Primäre Aufgabe von Datenbanksystemen war es zunächst, Daten in strukturierter Form zu speichern und bearbeiten zu können. Diese Möglichkeit wurden bereits früh für den Einsatz in Geoinformationssystemen erkannt, sodass Datenbanksysteme seit vielen Jahren in diesen implementiert sind.

Man geht heute davon aus, dass über 80% der weltweit gespeicherten Daten einen Raumbezug aufweisen. Die Notwendigkeit der sinnvollen graphische Visualisierung dieser Daten tritt immer mehr ins Bewusstsein der Öffentlichkeit. Institutionen, die auf riesige Datenmengen zurückgreifen können - exemplarisch seien die Europäische Union und die UNO genannt - haben die Zeichen der Zeit erkannt und stellen große Bemühungen an, sinnvolle Konzepte zur graphischen Darstellung ihrer Daten zu entwickeln. Bei diesen Überlegungen müssen zwei Aspekte berücksichtigt werden. Zum einen wäre die Dynamik zu beachten, mit der sich Datenbestände vergrößern und verändern, zum Anderen der Drang nach Information seitens der Nutzer. Die heutige Kommunikationsgesellschaft verlangt nach aktueller, weltweiter Information in Sekundenschnelle. Hinzu kommt der

Wunsch der multimedialen Präsentation sowie die Option, Art und Aussehen individuell - interaktiv - zu beeinflussen.

Neue technische Entwicklungen erweitern das Einsatzgebiet von Datenbanksystemen um die Option, räumliche Daten innerhalb weniger Sekunden zu visualisieren und dem Nutzer zu präsentieren. Die Möglichkeit, direkt auf bestehende Datenbankinhalte zuzugreifen, und diese unmittelbar darzustellen, erlaubt es dem Nutzer, die Gestaltung der Abbildung nach eigenen Wünschen - in Echtzeit - zu beeinflussen. Der interaktive Zugriff auf eine zentrale Datengrundlage ermöglicht es somit dem Anwender, sein persönliches Graphikmodell zu erzeugen und dieses gegebenenfalls in Sekundenschnelle abzuändern oder zu adaptieren.

Systeme, die in der Lage sind, diese Aufgabe zu erfüllen, bestehen aus mehreren Modulen, die isoliert voneinander als eigenständige Software-Produkte zu verstehen sind. Erst die Summe aller Einzelkomponenten liefert ein voll funktionsfähiges kartographisches Visualisierungssystem. Ziel dieser Arbeit soll es sein, die Nutzung der einzelnen benötigten Systemmodule im Rahmen der modernen Kartographie zu hinterfragen, wobei besonderes Interesse der Einbindung und Nutzung von Datenbanksystemen gilt. Es soll primär geklärt werden, inwieweit der Einsatz von Datenbanksystemen die Möglichkeiten kartographischer Visualisierung im Internet erweitern und verbessern kann. Die Bearbeitung der Fragestellung soll sowohl aus einer theoretischen, als auch aus einer praktischen Sichtweise erfolgen. Der verwendete wissenschaftliche Ansatz ist dabei weder als technische, noch als applikationsbezogene Frage zu verstehen, sondern vielmehr als kartographische Fragestellung, die die Bedeutung von Datenbanksystemen, Darstellungswerkzeugen und Internet im Rahmen eines gemeinsamen Visualisierungssystems für die moderne Kartographie und Geoinformation aufwerfen soll.

Um die wissenschaftliche Fragestellung fachlich exakt zu positionieren, ist es notwendig, diese gegenüber anderen Interessensgebieten klar abzugrenzen. Dieser Vorgang kann nach unterschiedlichen Gesichtspunkten erfolgen:

**? Abgrenzung und Positionierung der Kartographie:**

Die Kartographie sieht sich heute dramatischen Veränderungen ausgesetzt, die vor allem auf den Einsatz neuer Technologien im Computer- und Telekommunikationsbereich zurückzuführen sind. Obwohl diese Veränderungen primär technischer Art sind, beeinflussen sie doch sehr stark die heutigen Methoden und Konzepte der

Kartographie. Die überwiegende Zahl der neuen technischen Innovationen - Internet, Multimedia, Datenbanksysteme, um nur einige zu nennen - wurde ursprünglich keineswegs für den Einsatz in der Kartographie konzipiert [TAYLOR 1996]. Kartographen bietet sich allerdings die einmalige Chance, diese Techniken aufzugreifen und für ihre speziellen Bedürfnisse zu nutzen. Viele Herausforderungen, denen sich die moderne Kartographie und Geoinformation in der Zukunft stellen muss, werden durch technische Innovationen unterstützt. Die zentrale Aufgabe des Kartographen wird es aber stets sein, die Technik als Werkzeug nutzend, kartographisches Wissen, Kreativität und Ästhetik zu hochwertigen Produkten zu vereinen.

? **Abgrenzung nach kartographischen Aspekten:**

Im Rahmen der vorliegenden Arbeit soll das Hauptaugenmerk auf Erläuterungen zur Erstellung großmaßstäbiger Darstellungen, insbesondere auf dem Gebiet der Stadtkartographie, liegen. Die Übertragung der vorhandenen Fragestellungen auf andere Maßstabs- und Themenbereiche der Kartographie ist aber durchaus denkbar und wünschenswert.

? **Abgrenzung nach Datenbank-technischen Gesichtspunkten:**

Datenbanken spielen heute bei der kartographischen Visualisierung von Daten eine wesentliche Rolle. Dies ist alleine schon daran zu erkennen, dass praktisch sämtliche kartographischen Verlage und Institutionen sowie kommunale Einrichtungen ihre Daten in strukturierter Form mittels Datenbankmanagementsystemen verwalten. Die Bedeutung der Visualisierung von Daten ist heute allgemein bekannt, woraus sich eine große Nachfrage nach Werkzeugen ergibt, die diese Aufgaben erfüllen. Der Einsatz von Datenbanksystemen in Softwareprodukten, die für kartographische Zwecke herangezogen werden können, ist vielfältig und reicht von einfachen digitalen Zeichenwerkzeugen über komplexere Visualisierungstools im Internet bis hin zu vollständigen Geoinformationssystemen. So bieten etwa bereits relativ einfach anzuwendende Desktop-Mapping-Pakete (z.B. Macromedia Freehand, Adobe Illustrator, LevelNine Digisys) die Möglichkeit, über eine genormte Schnittstelle auf Sachdaten in einem externen Datenbankmanagementsystemen zuzugreifen. Auf der anderen Seite der Komplexitätsskala stehen vollausgerüstete Geoinformationssysteme, die neben den Sachdaten auch die gesamte Geodatenstruktur in -zumeist relationalen oder objektorientierten - Datenbanken verwalten. Besonders der Bereich der Datenbankstrukturen und -modelle in komplexen Geoinformationssystemen kann als denkbares, ähnliches Thema zur vorliegenden Arbeit gesehen werden, da das Hauptaugenmerk primär auf die

Verwendung von Datenbanksystemen für kartographische Visualisierungssysteme gelegt werden soll.

? **Abgrenzung innerhalb des logischen Aufbaus eines datenbankgestützten Visualisierungssystems:**

Wie oben erwähnt sollen besonders die datenbankrelevanten Aspekte Internet-basierter Visualisierungswerkzeuge besprochen werden. Der Nutzen derartiger Systeme besteht darin, kartographische Information in interaktiver Form darzustellen. Um diese Aufgabe erfüllen zu können, sind verschiedene Systemkomponenten notwendig. Neben dem Datenbankmanagementsystem sind dies vor allem die Benutzeroberfläche und das Visualisierungsprogramm - der Mapserver. Im Rahmen dieser Arbeit soll insbesondere auf den Einsatz der Datenbankmanagementsysteme sowie der verwendeten Schnittstellen für kartographische Visualisierungssysteme eingegangen werden. Ebenso wichtig ist es allerdings, die Themenbereiche Mapserver und Benutzeroberfläche ausführlich zu behandeln.

Um bewerten zu können, inwieweit eine Datenbankunterstützung den Leistungsumfang eines kartographischen Visualisierungssystems erweitert, müssen grundlegende Aspekte der Datenbanksysteme erläutert werden. Dies soll im zweiten Kapitel erfolgen.

Kapitel drei greift die Fragestellung auf, wie ein datenbankgestütztes Internet-basiertes Visualisierungssystem die kartographische Kommunikation modellhaft verändert. Der Tatsache der Trennung der Datenspeicherung von der Visualisierung soll sowohl theoretisch, als auch praktisch an Hand ausgesuchter Geodatenmodelle Rechnung getragen werden. Ausführungen zur digitalen Stadtkartographie schließen diesen Abschnitt ab.

Die Beweggründe, das Internet als Ausgabemedium zu wählen, soll in Kapitel vier begründet werden. Neben einer kurzen Chronologie der Genese des Internet sind wichtige Aspekte zu nennen, die die heutige Bedeutung dieses weltweiten Netzwerkes - auch aus der Sicht der Kartographie - ausmachen. Daneben sollen die unterschiedlichen Techniken der Erstellung von Internetinhalten erläutert werden.

Die Möglichkeiten der Darstellung kartographischer Inhalte im Internet werden in Kapitel fünf besprochen. Hier sollen vor allem die Anforderungen erwähnt werden, die an datenbankgestützte kartographische Visualisierungssysteme im Internet gestellt werden, sowie Gliederungsmöglichkeiten denkbarer Lösungen angeboten werden.

Kapitel sechs beschreibt in allgemeiner Form den Aufbau eines datenbankgestützten Mapserver-Systems. Die einzelnen Module, die benötigt werden um eine voll funktionsfähige Applikation zu erstellen werden eingehend besprochen und gemäß ihres Einsatzgebietes auf ihre Brauchbarkeit untersucht, sowie deren Funktionsweise im Rahmen eines Mapserver-Systems erläutert.

Nach den theoretischen Erläuterungen zeigt Kapitel sieben die praktische Umsetzung eines datenbankgestützten Mapserver-Systems im Internet an Hand eines erstellten Prototypen. Sowohl der Aufbau des Systems, die verwendete Daten und Software-Komponenten, die Systemarchitektur sowie die Funktionsweise werden eingehend behandelt.

Den Abschluss bildet eine Zusammenfassung in Kapitel 8.

## **2 Datenbanksysteme**

Neben wichtigen Begriffsbestimmungen soll im Rahmen dieses Kapitels erläutert werden, welche unterschiedlichen Formen der Datenspeicherung Datenbanksysteme ermöglichen. Ebenso wird auf relevante Aspekte der Datenqualität sowie der Bearbeitung von Datenbanken näher eingegangen.

### **2.1 Begriffsbestimmung**

Um die korrekte Verwendung von Fachausdrücken im Rahmen der vorliegenden Arbeit gewährleisten zu können, ist es notwendig, diese Fachtermini im Vorfeld zu erläutern. Im Folgenden sollen daher ausgewählte Fachbegriffe aus der Welt der Datenbanksysteme näher spezifiziert werden.

#### **2.1.1 Datenbank**

Unter einer Datenbank (DB) versteht man eine Sammlung von Daten (häufig auch als Datenbasis bezeichnet). Rohdaten liegen zunächst in unstrukturierter Form vor. Erst die Implementierung von Daten in ein Datenbankmanagementsystem führt zu einer eindeutigen Strukturierung und ermöglicht somit die Pflege sowie die Auswertung dieser. Datenbanken können durch eine einzelne Datei, mehrere Dateien auf einem oder auf verschiedenen Computern in einem Netzwerk repräsentiert werden. Datenbanken dienen vor allem zur Speicherung und schnellen Abfrage großer Informationsbestände. Diese Abfragen werden mittels standardisierter Abfragesprachen durchgeführt, wobei sich Standard Query Language (SQL) mittlerweile als de-facto Standard etabliert hat.

#### **2.1.2 Datenbankmanagementsystem**

Ein Datenbankmanagementsystem (DBMS) ist ein Programmsystem, das zum Aufbau, der Pflege und der Auswertung einer Datenbank verwendet wird. Es stellt somit die "Software"

dar, derer man sich bedient, um große Datenmengen leichter und effizienter bearbeiten zu können. Einer der wichtigsten Gründe, weswegen Datenbankmanagementsysteme anderen Formen der Dateispeicherung vorgezogen werden, ist die Möglichkeit, Daten in strukturierter Form zu speichern und zu bearbeiten. Die Art und Weise dieser Struktur wird durch logische Datenmodelle beschrieben (vergleiche Kap. 2.3). Datenbankmanagementsysteme sind hochperformante Software-Applikationen, die häufig auf eigens dafür konzipierten und reservierten Datenbank-Servern betrieben werden. Benutzung und Administration von Datenbankmanagementsystemen wird demgegenüber mittels Datenbank-Clients durchgeführt. Dies sind Programme, die es dem Benutzer ermöglichen, über ein Netzwerk auf den Datenbankserver zuzugreifen [HEUER & SAAKE 2000].

In der Praxis zeigt sich häufig eine falsche Verwendung dieser technischen Termini. So werden Datenbankmanagementsysteme oft fälschlicherweise als Datenbanken bezeichnet. Dass dies zu weitreichenden Verwirrungen führen kann, scheint logisch, da es sich auf der einen Seite um eine Datenbasis handelt, auf der anderen Seite um das Softwareprodukt zu deren Bearbeitung.

### **2.1.3 Datenbanksysteme**

Die Einheit einer Datenbank mit dem zu ihrer Verarbeitung notwendigen Datenbankmanagementsystem wird als Datenbanksystem (DBS) bezeichnet. Ebenso wie Datenbanken müssen moderne Datenbanksysteme nicht zwingend auf einen einzigen Computer limitiert sein, sondern können ein über ein Netzwerk verteiltes System von Datenbank-Servern und deren Clients darstellen [VOSS 2000]. Gerade im Zeitalter der Massenkommunikation via Internet stellt dies eine wichtige Tatsache dar, da Informationen dem Nutzer automatisch von jenem Server geliefert werden können, der diese am schnellsten liefern kann. Die physische Distanz zum Speicherort der Daten ist somit nicht länger wichtig, vielmehr gewinnt die Datenübertragungsrate eine immer größere Bedeutung.

## 2.2 Datenspeicherstrukturen

Datenspeicherung kann in unterschiedlichen Formen erfolgen. Heute ist jene der Datenbankmanagementsysteme weit verbreitet. Die historische Entwicklung hin zu diesen Systemen kann in drei Stufen skizziert werden.

- ? Datei-Systeme
- ? Datei-Verwaltungs-Systeme
- ? Moderne Managementsysteme

### 2.2.1 Datei-Systeme

Bereits im den frühen 60er Jahren wurden die theoretischen Möglichkeiten von Datenbankmanagementsystemen erforscht. In der Praxis wurden Daten aber zunächst in einzelnen, voneinander unabhängigen Dateien abgelegt, es erfolgte somit eine anwendungsspezifische Datenorganisation. Jede Anwendung griff ohne spezielle Verwaltung auf bestehende Dateien zu.

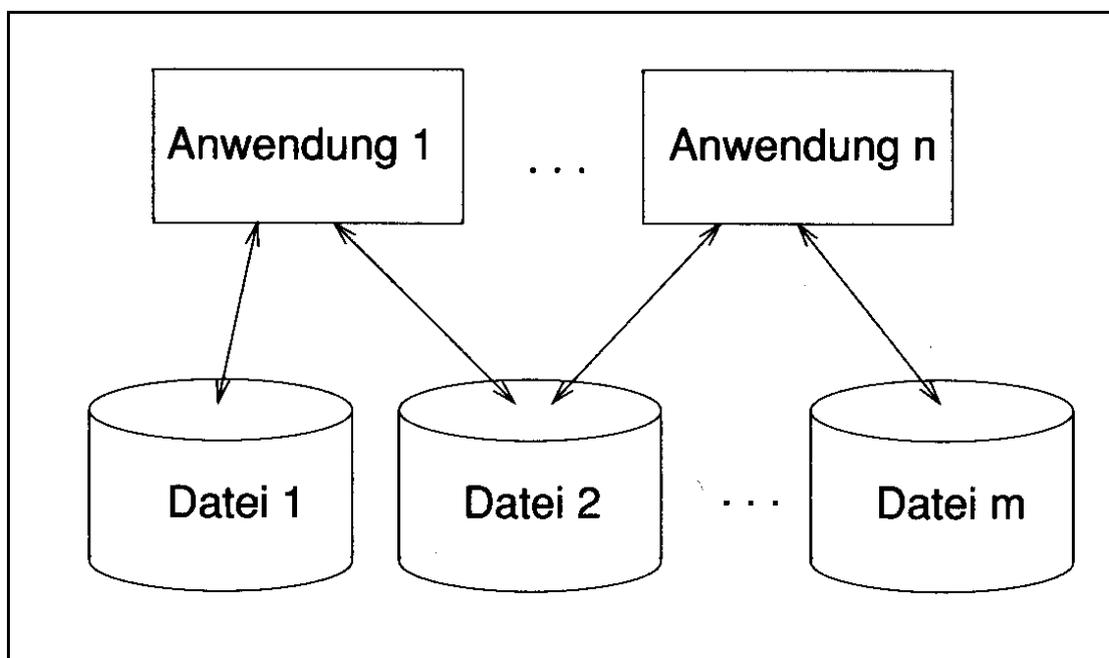


Abb. 1: Struktur eines Datei-Systems [HEUER & SAAKE 2000]

Dies war eine sehr ineffiziente Form der Datenverwaltung, da einzelne Anwendungen ihre Daten jeweils in eigenen Dateien abspeicherten. Dieser Sachverhalt der mehrfachen Speicherung identer Daten wird als Redundanz bezeichnet.

### 2.2.2 Datei-Verwaltungs-Systeme

In den späten 60er Jahren wurden Datei-Verwaltungs-Systeme verwendet, die um zusätzliche Dienstprogramme erweitert wurden und so etwa das Sortieren von Datenbeständen ermöglichten. Diese zwischen Dateien und Anwendungen platzierten Systeme führten zu einer verbesserten Verwaltung von Datenbeständen, waren aber in ihrer Leistungsfähigkeit noch weit von heutigen Datenbankmanagementsystemen entfernt.

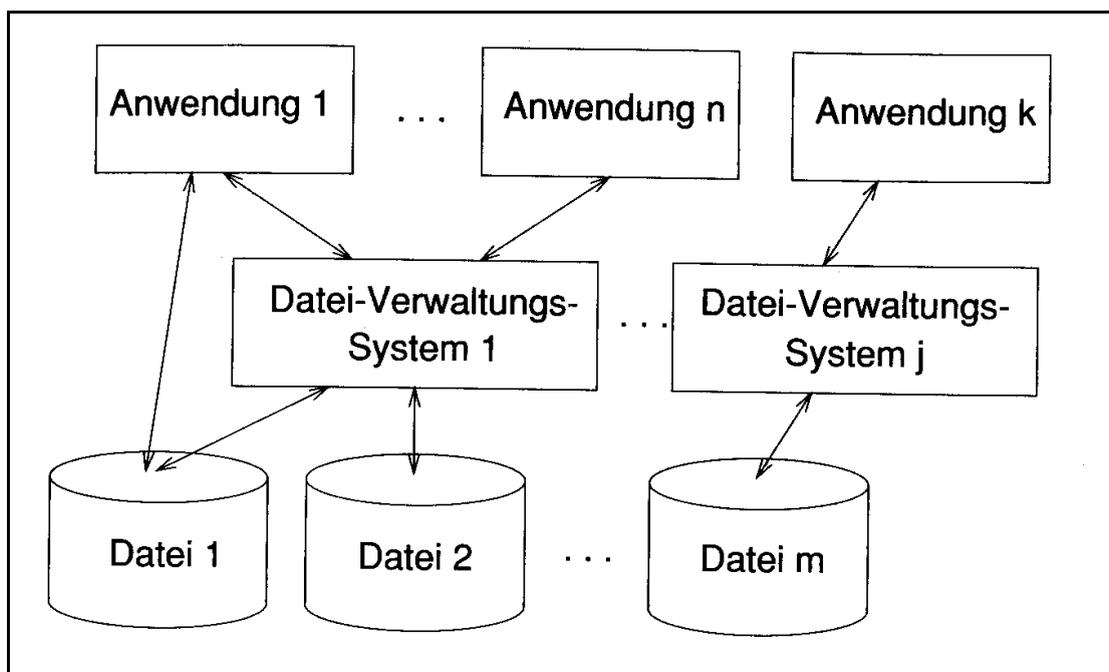


Abb. 2: Aufbau eines Daten-Verwaltungs-Systems [HEUER & SAAKE 2000]

Als Vorteil dieser Systeme gegenüber den Datei-Systemen kann eine gewisse Geräte-unabhängigkeit gesehen werden. Wie aus Abb. 2 ersichtlich muss Anwendung 1 nicht zwingend wissen, wo sich Datei 2 physisch befindet, da das Datei-Verwaltungs-System

diese Aufgabe übernimmt. Das Problem der Datenredundanz war aber auch mit diesen Systemen nicht zu lösen.

### **2.2.2 Moderne Managementsysteme**

Die redundanzfreie Datenhaltung wurde in den frühen 70er Jahren durch die Entwicklung von Datenbankmanagementsystemen erreicht. Das Prinzip dieser Systeme besteht darin, Daten einmalig in einer Datenbank zu speichern und diese von einem Datenbankmanagementsystem verwalten zu lassen. Alle Anwendungen greifen somit zentral auf die Daten zu.

Die Basis-Funktionalitäten, die von einem modernen Datenbankmanagementsystem erwartet werden, stellte erstmals CODD (1982) dar:

- ? Definition der Datenbankstruktur, Datenmodell und der Datenelemente, insbesondere der Anzahl und der Eigenschaften der Datenfelder sowie deren Verknüpfungen.
- ? Die Datenintegration erfordert die einheitliche Verwaltung aller von sämtlichen Anwendungen benötigter Daten. Hier verbirgt sich die Möglichkeit der kontrollierten, nicht-redundanten Datenhaltung des gesamten relevanten Datenbestandes.
- ? Verwaltung der Datenbasis, insbesondere die Eingabe, die Übernahme, das Einfügen, das Löschen sowie das Editieren von Datensätzen.
- ? Verarbeitung der Daten im engeren Sinne. Zu erwähnen wären in diesem Zusammenhang besonders das Filtern, das Sortieren sowie das Extrahieren von Datensätzen.
- ? Ausgabe und Übergabe von Informationen. Als Ausgabe versteht man das kontrollierte Exportieren von Datensätzen, als Übergabe den Datentransfer an andere Softwareprodukte mittels geeigneter Schnittstellen.
- ? Realisierung von Sicherheitsfunktionen.

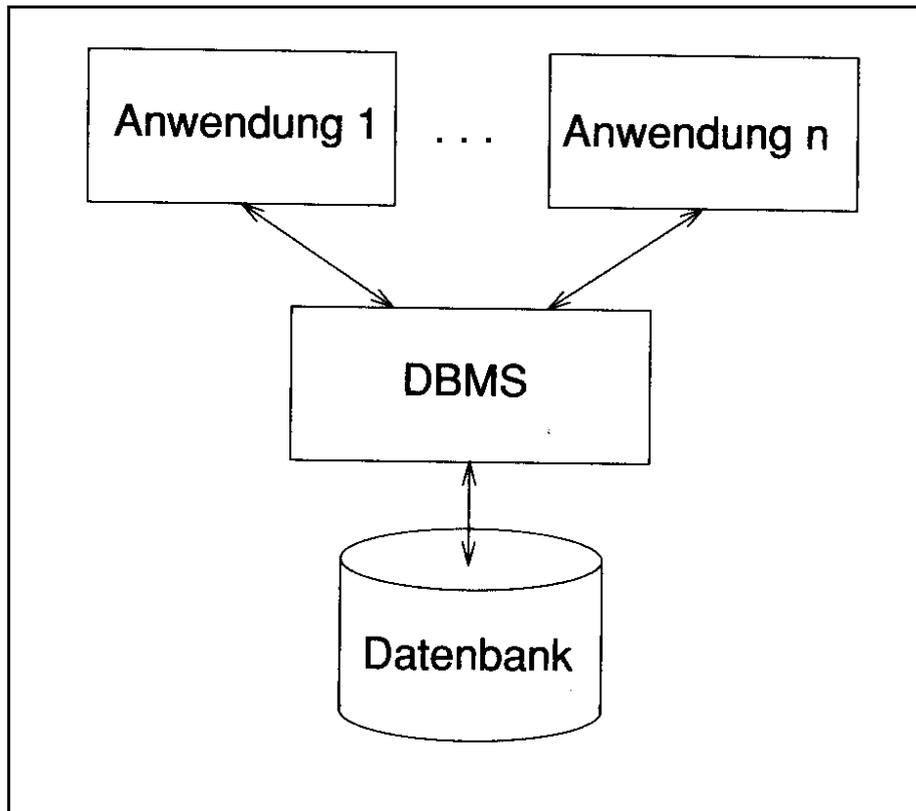


Abb. 3: Struktur moderner Datenbankmanagementsysteme [HEUER & SAAKE 2000]

## 2.3 Logische Datenmodelle

Neben den angesprochenen Anforderungen an ein Datenbankmanagementsystem stellt die benötigte Rechnerzeit, insbesondere bei Internet-basierten Anwendungen, einen nicht zu unterschätzenden Qualitätsfaktor dar. Um Datenzugriffe zu optimieren, ist es notwendig, eine dem Verwendungszweck passende Struktur der Daten aufzubauen. Die Art und Weise dieser Struktur wird in Datenmodellen beschrieben. Diese legen die Beziehungen der Daten untereinander fest und ermöglichen daher eine effiziente Datenverwaltung.

Neben unstrukturierten Datenbeständen differenziert man in der Regel folgende Datenmodelle, die sich durch die Art des Datengefüges unterscheiden [BILL 1991]:

- ? Hierarchisches Datenmodell
- ? Netzwerkmodell
- ? Relationales Datenmodell
- ? Objektrelationales Datenmodell

? Objektorientiertes Datenmodell

### **2.3.1 Hierarchisches Datenmodell**

Dieses Modell wird verwendet, wenn von einem Oberbegriff ausgehend viele Unterbegriffe abgeleitet werden können. Man spricht in diesem Fall von einer 1:n Beziehung der Daten. Dies bedeutet, dass von einem Element ausgehend mehrere Beziehungen bestehen können. Ein Beispiel für eine derartige Struktur wäre, dass ein Bezirk mehrere Gemeinden aufweist. Ein Datensatz wird mit allen hierarchisch von ihm abhängigen Datensätzen als Einheit betrachtet. Ein Zugriff auf einen bestimmten Datensatz kann somit nur über den Suchschlüssel des Objekts der obersten Ebene erfolgen. Diese Tatsache verdeutlicht das hierarchische Prinzip dieses Modells. Hierarchische Datenmodelle eignen sich nicht für die Speicherung raumbezogener Information, da die auftretenden komplexen Strukturen kaum sinnvoll abzubilden sind [HAKE & GRÜNREICH 1994].

### **2.3.2 Netzwerkmodell**

Im Gegensatz zu dem hierarchischen Modell liegen die Datensätze in einem Netzwerkmodell nicht in einer Abfolge von Rangunterschieden vor. Vielmehr ist es erlaubt, dass ein Datensatz eine beliebige Anzahl übergeordneter Datensätze aufweist, was die Möglichkeit bietet, neben 1:n auch n:m Beziehungen aufzubauen. Ein Zugriff kann somit über beliebige Datensätze erfolgen. Diesen Erweiterungen gegenüber dem hierarchischen Modell steht allerdings eine starre Organisation des Datenbestandes gegenüber, die das Anpassen an veränderte Rahmenbedingungen erschwert.

Das Wissen um die Interna des hierarchischen wie auch des Netzwerkmodells ist durchaus von Bedeutung, obgleich diese Strukturen heute in neu aufzubauenden Systemen kaum noch berücksichtigt werden. Es existieren aber durchaus noch derartige Datenbestände, die noch für längere Zeit im operativen Betrieb sein werden. Daneben verlangt die mögliche Umstrukturierung dieser in relationale oder objektorientierte Datenmodelle die Kenntnis der Speichermethoden der veralteten Systeme [HEUER & SAAKE 1999(b)].

### 2.3.3 Relationales Datenmodell

Im Zentrum des relationalen Modells steht die Tabelle. Innerhalb einer Tabelle stellen die Reihen die Datensätze, die Spalten die Datenfelder dar. Die Spalten können jeweils nur Inhalte eines Datentypes aufnehmen. Datentypen bestehen sowohl für numerische, als auch für alphanumerische Einträge. Übliche Datentypen sind "int", "tinyint" oder "float" für numerische sowie "char" oder "varchar" für alphanumerische Werte. Daneben existieren Datentypen für Datumseingaben sowie solche für binäre Dateien. Jede Tabelle muss zwingend ein Feld enthalten, dessen Inhalte eindeutige Werte darstellen. Diese, als Tabellenschlüssel (primary key) bezeichnete Spalte ermöglicht es, jeden Datensatz eindeutig zu identifizieren. Beziehungen zwischen beliebigen Tabellen lassen sich nun über idente Feldinhalte eines Tabellenschlüssels herstellen. Es ist somit möglich, Beziehungsvarianten herzustellen, die nicht nur auf 1:n und n:m beschränkt sind. Die folgende Tabelle zeigt mögliche Beziehungstypen zwischen zwei Tabellen in einem relationalen Datenmodell:

Relationale Beziehungstypen		Beispiel
1	zu eins	Jeder Fluss hat einen Ursprung
1	zu eins oder keins	Jeder Fluss mündet in einem oder in keinem Meer
1	zu mindestens eins	Jeder Fluss fließt durch mindestens ein Land
1	zu keins, eins oder mehreren	Jeder Fluss hat keinen, einen oder mehrere Zuflüsse

Tab. 1: Beziehungstypen in einem relationalen Datenmodell

Das relationale Datenmodell liegt gegenwärtig den meisten Datenbankmanagementsystemen zugrunde. Da die Entwicklung und Marktreife relationaler Datenbanksysteme etwa zeitgleich mit dem Aufkommen erster Geoinformationssysteme angesetzt werden kann, wurden sie früh in diese implementiert und sind bis heute das vorherrschende Datenbanksystem im GIS-Bereich.

Im Einsatz mit Geodaten zeigen relationale Datenbanksysteme aber ebenso wie hierarchische und Netzwerkmodelle Schwächen. Die beziehungsreichen Strukturen der realen Welt lassen sich nicht verlustfrei in einem relationalen Modell abbilden. Durch die

Implementierung von objektrelationalen, vor allem aber objektorientierten Datenmodellen wird versucht, diesen Umstand zu verbessern. Als besondere Innovation wäre hierbei die Verwendung von Eigenschaften und Methoden und im Falle eines rein objektorientierten Modells, deren Vererbung zu nennen.

### **2.3.4 Objektrelationales Datenmodell**

Um zu einem objektrelationalen Datenmodell zu gelangen, muss das relationale Modell um bestimmte objektorientierte Konzepte erweitert werden. Es handelt sich hierbei vor allem um Relationen als Klassen, Typkonstruktoren, Objektidentität, Beziehungen und Methoden. Das objektorientierte Datenmodell bietet die Möglichkeit, Verhalten und Strukturen von Objekten gemeinsam zu modellieren. Zu beachten ist allerdings, dass die Basis eines objektorientierten Datenmodells trotz allem relational ist. Dies zeigt sich dadurch, dass Datenbankabfragen nur in einer für das relationale Modell adäquaten Form formuliert werden dürfen (vgl. Kap. 2.6.1.1), wohingegen echte objektorientierte Datenmodelle eigene Abfragesprachen mit sich führen [HEUER & SAAKE 2000]. Gegenwärtig ist zu beobachten, dass Hersteller professioneller Geoinformationssysteme diese vermehrt auf objektionale Datenmodelle aufbauen (vgl. Kap. 6.3.1.2).

### **2.3.5 Objektorientiertes Datenmodell**

Die konsequente Weiterentwicklung semantischer und geschachtelter relationaler Datenmodelle führt zum objektorientierten Modell. Ziel dieses Typus ist es, die Struktur und das Verhalten komplexer Objekte möglichst gut abzubilden. Es wird hierfür eine Kombination der Ansätze der oben erwähnten Datenmodelle verwendet. Diese werden mit Konzepten der Objektorientierung zu einem höchst innovativen Datenmodell vereint [SAMLAND, HINTERMEIER & DÜRR 2000].

Der objektorientierte Ansatz wird von FRANK und EGENHOFER (1990) folgendermaßen charakterisiert: "Man kapselt ein Datenobjekt, welches ein Stück der realen Welt repräsentiert, zusammen mit den zugehörigen Methoden in einer einzelnen Einheit. Andere

Programmteile oder Objekte können das so eingekapselte Objekt nur über die mit dem Objekt angelegten Methoden ansprechen und zu Aktionen veranlassen."

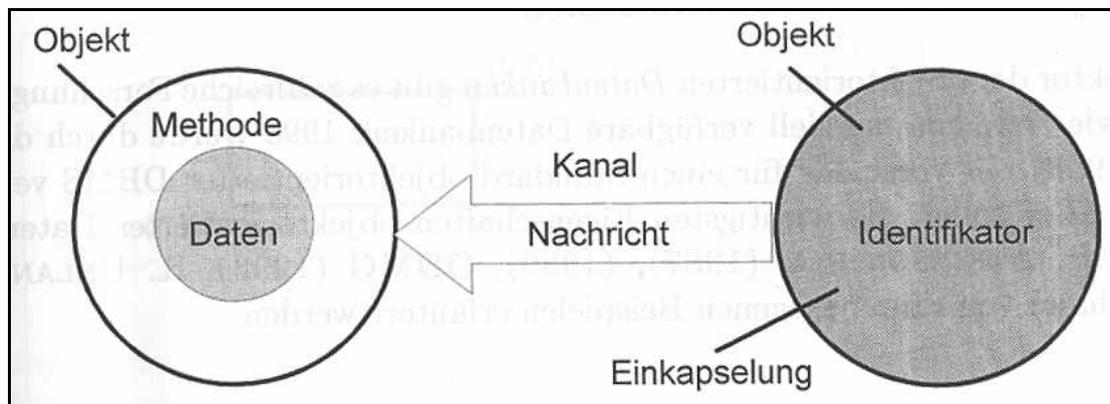


Abb. 4: Der objektorientierte Ansatz (BILL 1999)

Neben der Vererbung von Attributen und Methoden ist besonders die Objektidentität eine wesentliche Innovation gegenüber dem relationalen Datenmodell, da sie es erlaubt, gespeicherte Objekte von Attributen, die sie besitzen, zu unterscheiden.

## 2.4 Datengrundlagen

Die Entwicklung von Geoinformationssystemen hat gezeigt, dass eine direkte Beziehung zwischen der Informationsdarstellung und der Informationsverarbeitung besteht. Diese Tatsache setzt voraus, dass die Speicherung der Datengrundlagen einem Konzept unterliegt, das von der Datenerhebung ausgehend die Wahl des passenden Datenmodells berücksichtigt. Da die Speicherung kartographie-relevanter Daten in Datenbankmanagementsystemen sowohl Geometrie- als auch Sachdaten umfasst, ist es notwendig, diese näher zu erläutern.

### 2.4.1 Geometriedaten

Die Geometrie von räumlichen Objekten wird durch die Form und relative Lage von Punkten vollständig beschrieben. Der Punkt als topologischer Knoten ist der Träger der

geometrischen Information. Linien und Flächen können als Folge charakteristischer Punkte betrachtet werden. Innerhalb der Topologie ist nur die Tatsache wichtig, dass Punkte und Linien in einer bestimmten gegenseitigen Beziehung stehen und nicht die geometrische Form dieser Beziehungen. Der wesentliche Unterschied zwischen Topologie und Geometrie besteht darin, dass infolge topologischer Transformationen die Topologie unverändert bleibt, während sich die Geometrie sehr wohl ändert. Als Träger der topologischen Information gilt die Kante (Linie). Aus einer Kombination von Punkten und Linien ist es möglich, Flächen zu bilden. Die Geometrie von Objekten kann somit durch eine punkt-, linien- oder flächenhafte Darstellung repräsentiert werden. Innerhalb der Geometriedaten muss zwischen Vektor- und Rasterdaten unterschieden werden. Die punkt- und linienhafte Beschreibung von Objekten führt zu Vektordaten, die flächenhafte zu Rasterdaten.

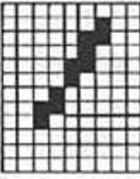
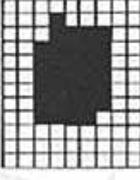
Element	Vektor		Raster	
	Digital	Analog	Digital	Analog
Punkt	x,y Koord.	•	Pixel	
Linie	x y Koord.- folge		Pixel	
Fläche	geschlossene x y Koord.- folge		Pixel	

Abb.5: Vergleich Rasterdaten - Vektordaten [BILL 1991]

### 2.4.1.1 Vektordaten

Die Grundelemente von Vektordaten sind der Punkt, die Linie und die Fläche. Mittels dieser können Nachbarschaftsbeziehung sowie Anfangs- und Endpunkt einer Linie eindeutig angegeben werden. Typische Anwendungsbereiche von Vektordaten liegen in der großmaßstäbigen Planungs- und Katasterkartographie. Daneben stellen Vektordaten die übliche Form der Geodaten in einem Geoinformationssystem dar. Die Gewinnung von Vektordaten kann sowohl mittels primärer Auswertung - etwa im Zuge der Photogrammetrie - als auch aus daraus abgeleiteter Information - etwa digitale Datenbestände oder Karten - im Rahmen der sekundären Auswertung erfolgen.

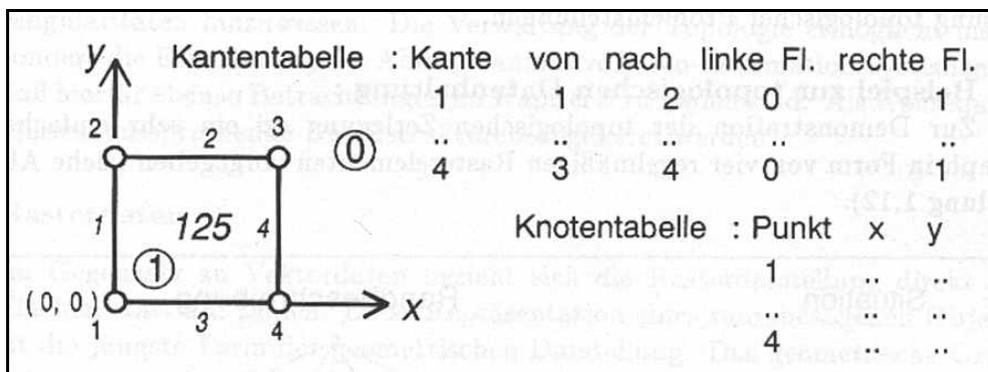


Abb. 6: Objektdarstellung mittels Vektordaten [BILL 1991]

Vektordaten weisen gegenüber Rasterdaten den Vorteil geringerer Datenmengen auf. Dies führt dazu dass vermehrt Bestrebungen bestehen, kartographische Ausdrucksformen im Internet mittels Vektorgraphiken darzustellen, da diese den anfallenden Datenstrom deutlich minimieren und so vordergründig eine erhebliche Qualitätssteigerung eines Produktes bewirken. Demgegenüber steht aber ein technisches Problem. Vektordaten können von Standard-Internetbrowsern nicht interpretiert werden. Es müssen daher Zusatzprogramme - Plug-Ins - installiert werden (vgl. Kap. 4.2.2.2). Neben dieser Tatsache sei noch erwähnt, dass nicht alle räumlichen Informationen durch Vektordaten dargestellt werden können. Satelliten- und Luftbilddaten liegen vielmehr als Rasterdaten vor.

### 2.4.1.2 Rasterdaten

Das geometrische Grundelement von Rasterdaten ist das Pixel (Bildelement). Die Summe der Bildelemente einer Rastergraphik ist in einer gleichförmigen, quadratischen oder rechteckigen Matrix zeilen- und spaltenweise angeordnet. Die Bildelemente enthalten Rasterwerte, die als Ausprägung einer Eigenschaft angesehen werden können, es besteht allerdings keinerlei logische Verbindung zwischen den Elementen.

Rasterdaten werden heute für die Speicherung bildhafter Darstellungen verwendet. Zur Datenerfassung muss die Vorlage gescannt werden. Bei diesem Arbeitsvorgang wird das Bild optisch mittels eines Lasers abgetastet und den Bildelementen die entsprechende Farbinformation als Rasterwerte zugeordnet [BILL 1991].

Zusammenfassend seien die Vor- und Nachteile von Vektor- und Rasterdaten angeführt:

	<b>Vektordaten</b>	<b>Rasterdaten</b>
<b>Graphische Grundstruktur</b>	Punkt, Linie, Fläche	Pixel (Bildelement)
<b>Betrachtungsweise</b>	Linienhaft	Flächenhaft
<b>Ordnung</b>	Objektlinien	Position der Pixel
<b>Objektbezug</b>	Leicht möglich	Sehr eingeschränkt
<b>Datenerfassung</b>	Punktuell, hoher Zeitaufwand	Flächenhaft, kurze Erfassungszeit
<b>Datenmenge</b>	Gering	Hoch
<b>Rechenzeit</b>	Kurz	Hoch

Tab. 2: Gegenüberstellung Vektordaten - Rasterdaten

### 2.4.2 Sachdaten

Geoinformationssysteme sind in der Lage, sowohl Geometrie- als auch Sachdaten zu verarbeiten. Sachdaten werden häufig auch als thematische Daten bezeichnet und stellen Attribute geometrischer Objekte dar. Sachdaten sind beschreibende Daten, die bis auf die geometrischen Merkmale sämtliche Informationen eines raumbezogenen Systems enthalten. Erst durch die Verknüpfung von Objekten mit ihren entsprechenden Attributen können diese

eindeutig identifiziert und klassifiziert werden. Moderne Methoden der Datenübertragungstechnik machen es heute möglich, Geometriedaten und Sachdaten auf verschiedenen Datenservern zu verwalten, diese aber in einer gemeinsamen Darstellung zu visualisieren.

### 2.4.3 Metadaten

Metadaten (auch Metainformation) werden häufig als "Daten über Daten" bezeichnet. Eine offizielle und verbindliche Definition des Begriffs Metadaten existiert bislang nicht. So ist etwa WEBSTER'S'S New World College Dictionary (1999) zu entnehmen, "Metadata is an abstraction from data. It is high-level data that describes lower-level data.", während die ISO Spezifikation 11179 (1995) Metadaten sehr pragmatisch als "The information and documentation which makes data sets understandable and shareable for users." beschreibt. Metadaten können somit als Informationen über Daten angesehen werden und bieten dadurch die Möglichkeit einer intelligenten und effizienten Verwaltung sowie Zugriff auf die beschriebenen Daten.

Der Versuch, einheitliche Standards der Metadatenerstellung für Geodaten zu entwickeln, wird vom Federal Geographic Data Committee unternommen. Ein Metadatensatz, der räumliche Daten beschreibt, sollte demnach folgende Informationen aufweisen [FGDC 1998]:

? **Identifikation:**

Der oder die Urheber, der Titel und das Veröffentlichungsdatum der Daten.

? **Informationen zur Datenqualität:**

Generelle Einschätzung der Güte der Daten, der Datengewinnung und der Datengrundlagen. (vgl. Kap. 2.5)

? **Informationen zur Organisation der räumlichen Daten:**

Namen und Typen geographischer Merkmale, Ordnungsschlüssel, geometrischen Repräsentationen sowie Angaben zur Anzahl der geometrischen Elemente.

? **Informationen zum Raumbezug:**

Koordinatensystem und Projektion sowie geodätische Angaben zu Höhen und Tiefen.

? **Geoobjekte und ihre Attribute:**

Systematik der Geoobjekte im Allgemeinen sowie Angaben über Attribute, Attributwerte und ihre Wertebereiche im Speziellen.

? **Informationen zur Datenweitergabe:**

Ansprechpartner, Organisation des Bestellwesens, verfügbare Datenformate, weitere technische Angaben.

? **Informationen zu den Metadaten:**

Ansprechpartner, die Aktualität und die Zugriffs- und Nutzungsrechte der Metadaten.

Mit dem rapiden Wachstum der weltweiten Datenmenge wurde die Notwendigkeit von Metadatenbanken erkannt. Diese bieten lediglich Informationen über Datensätze, nicht aber die Daten an sich. Der konkrete Nutzen von Metainformation zeigt sich beim Einsatz in Datenbanksystemen, insbesondere bei der Verwaltung räumlicher Informationen, da diese enorme Speichermengen aufweisen.

Die Hauptanwendungsmöglichkeiten von Metadaten beschreibt MOSSGRABER (1997) :

? **Suche in Datenbeständen:**

Das Anlegen und Suchen in Metadatenbanken bietet wesentliche Vorteile gegenüber der Suche in den Originaldatenbanken. Durch die Vernetzung weltweiter Dateninformationssysteme ist es möglich, die Brauchbarkeit eines Datensatzes anhand seiner Metadaten zu überprüfen, bevor die Originaldaten angefordert werden.

? **Datendokumentation:**

Existente Datensätze sind nur so gut wie ihre Dokumentation. Die Dokumentation von Daten in den Metadaten dient der langfristigen Werterhaltung von Daten, hilft aber auch, die Gültigkeit und Qualität der Daten besser einschätzen zu können.

In den letzten Jahren zeigt sich vermehrt der Bedarf nach "Data-Mining", der automatischen Durchforstung undokumentierter Datenbestände. Zahlreiche am Markt befindliche Software-Produkte bieten diese Möglichkeit.

? **Datentransfer:**

Beim Austausch von Daten muss dem Empfänger die Möglichkeit geboten werden, diese sinnvoll nutzen zu können. Dies wird durch Metadaten ermöglicht, die sowohl Informationen über den syntaktischen Aufbau der übertragenen Datei wie auch zur Interpretation der Daten enthalten.

## 2.5 Datenqualität, -kontrolle

Im Rahmen der vorliegenden Arbeit wurde darauf hingewiesen, wie wichtig die Phase der Konzeption und Planung der auf die Bedürfnisse des Anwenders passenden Datenbankstruktur sowie des Datenmodells ist. Es sei aber festgehalten, dass das beste Datenbankdesign nur so gut ist wie die in ihr enthaltenen Daten. Der Qualitätsaspekt von Daten soll daher nicht unerwähnt bleiben.

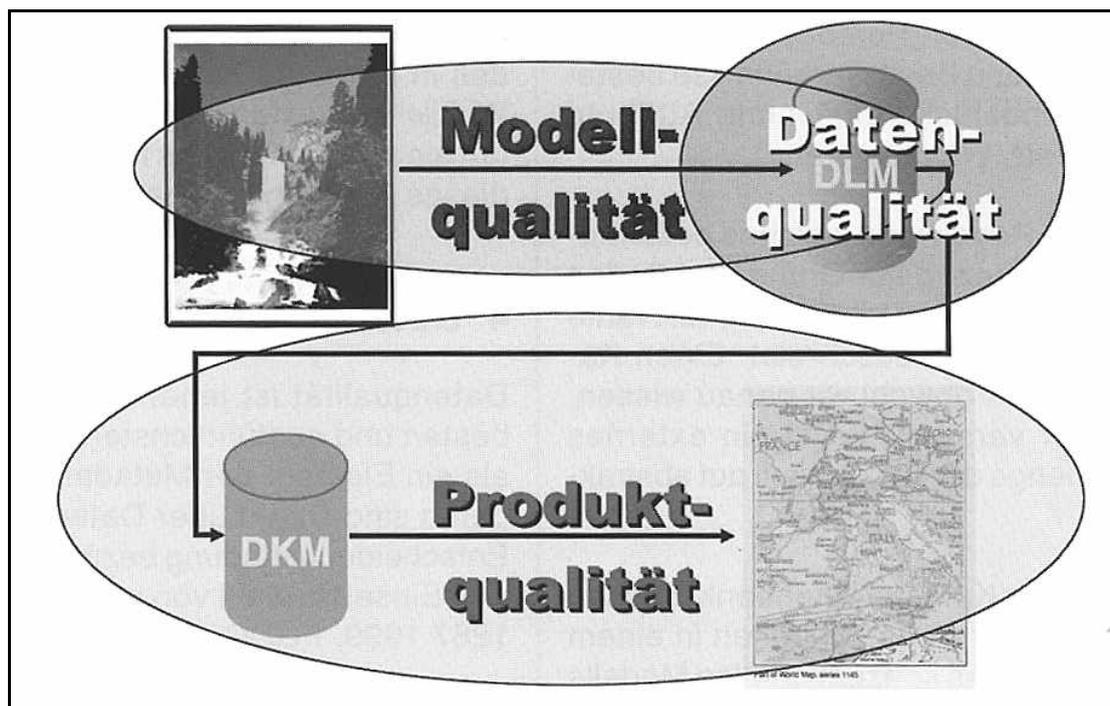


Abb. 7: Position der Datenqualität im kartographischen Qualitätsmodell (KAINZ 1999)

Gemäß ISO NORM 8402 ist Qualität "die Gesamtheit von Merkmalen einer Einheit bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen." Im

Rahmen der Kartographie und Geoinformation ist hiermit zumeist die Güte des Modells im Vergleich zur Realität zu verstehen. Geodaten weisen besondere Qualitätsmerkmale auf, da sie Raum-, Sach- und Zeitbezug haben müssen, um effizient genutzt werden zu können. Es gilt somit zu berücksichtigen [CASPARY 1992]:

- |                      |                        |
|----------------------|------------------------|
| ? Herkunft der Daten | ? Positionsgenauigkeit |
| ? Vollständigkeit    | ? Attributgenauigkeit  |
| ? Aktualität         | ? Logische Konsistenz  |

Die Herkunft der Daten beinhaltet neben den Erfassungsmethoden und der Entstehungsgeschichte auch den gegenwärtigen Zustand der Daten. Diese Merkmale sind im optimalen Fall Bestandteil der Metadaten und können als Lebenslauf der Daten angesehen werden. Das Merkmal der Vollständigkeit von Geodaten muss in einen Aspekt der räumlichen Vollständigkeit und einen der thematischen Vollständigkeit gegliedert werden. Räumliche Vollständigkeit entspricht der Übereinstimmung der Anzahl der Objekte des Modells mit der Realität. Thematische Unvollständigkeit führt zu fehlenden Attributdaten einzelner Objekte. Die zeitliche Komponente eines Objekts oder einer Relation, sein Erfassungs-, Veränderungs- und Aktualisierungsdaten geben einen Überblick über die Aktualität eines Datensatzes. Es besteht durchaus ein Zusammenhang zwischen räumlicher, thematischer und zeitlicher Unvollständigkeit von Daten. Infolge einer zeitlichen Unvollständigkeit der Aktualisierung von Daten kann leicht auch eine räumliche sowie thematische Unvollständigkeit entstehen.

Jene Qualitätsmerkmale von Daten, die in engem Zusammenhang mit der Speicherung und Bearbeitung von Geodaten in Datenbanksystemen stehen, sollen in der Folge näher erläutert werden.

- ? Positionsgenauigkeit
- ? Attributgenauigkeit
- ? Logische Konsistenz

### **2.5.1 Positionsgenauigkeit**

Die Positions- oder Lagegenauigkeit ist ein Maß für die Güte der koordinativen Lage von Objekten eines Modells im Vergleich mit der Realität. In der Regel wird die Lagegenauigkeit durch die Wurzel des mittleren Fehlerquadrats angegeben [KAINZ 1999]. Dieser Wert stellt statistisch den wahrscheinlichsten Wert aus einer Reihe von Messungen dar und berücksichtigt auftretende Abweichungen nach dem Prinzip der Gauss'schen Fehlerverteilung. Diese besagt, dass das Auftreten von Fehlern in alle Richtungen gleich wahrscheinlich ist, kleine Fehler aber häufiger als große Fehler auftreten.

Positionsfehler sind abhängig von der Erfassungsmethode, der Gebietsgröße und der zu erfassenden Datenart. Man unterscheidet demnach zwischen relativer und absoluter Genauigkeit.

Die relative Genauigkeit ist das Ausmaß der Lagefehler innerhalb eines Bezugssystemes, während die absolute Genauigkeit die Güte der Lage der Summe der Objekte in jenem Bezugssystem angibt. OSWALDER (1996) nennt als Beispiel alte Katastermappen, die "sehr gute relative Genauigkeit haben, in ihrer absoluten Genauigkeit jedoch große Abweichungen vom Bezugssystem aufweisen."

### **2.5.2 Attributgenauigkeit**

Attributgenauigkeit wird neben Angaben über die Entstehung der Daten, Datenherkunft, Aktualität und Vollständigkeit als semantisches Qualitätsmerkmal bezeichnet, da sie sich auf die gespeicherte semantische Information der Daten und Metadaten bezieht [SCHILCHER & ROSCHLAUB 1996]. KAINZ (1999) sieht in der Attributgenauigkeit die Möglichkeit, Attribute von Objekten in einem Datensatz mit den tatsächlichen Attributen der entsprechenden Objekte in der Wirklichkeit zu vergleichen. Exemplarisch seien Nachkommastellen bei numerischen Angaben und Höhenangaben als Attribute positionsbezogener Objekte genannt.

Attributfehler können ebenso wie Positionsfehler bei der Erfassung räumlicher Daten sowohl mittels primärer wie auch sekundärer Aufnahmemethoden anfallen. Sie werden in der Folge in grobe, systematische und zufällige Fehler unterteilt.

Grobe Fehler können durch Kontrollalgorithmen relativ einfach behoben werden, da sie häufig augenscheinlich auftreten. So würde etwa die Abfrage "...alle Höhenangaben in Österreich über 4000 Meter" bereits eine Eliminierung dieses Fehlers herbeiführen.

Systematisch auftretende Fehler sind häufig auf Eichfehler der Messgeräte oder einseitig wirkende natürliche Einflüsse wie Temperatur oder Luftdruck zurückzuführen. Soweit die Ursache der störenden Wirkung bekannt ist, kann sie rechnerisch gezielt ausgeschaltet werden [OSWALDER 1996].

Zufällige Fehler sind die am schwersten zu eliminierenden Ungenauigkeiten von Daten. Das Auffinden derartiger Fehler ist komplex, können sie doch menschlicher wie auch technischer Ursache sein. Ein Ansatz, zufällige Fehler zu minimieren, besteht darin, Messungen mehrfach durchzuführen und so eine Überbestimmung des gemessenen Objekts oder Attributs zu erhalten. Weiters ist es möglich, mittels Fehlerausgleichsrechnungen den wahrscheinlichsten Wert zu berechnen. Die Zuverlässigkeit - Reliabilität - einer Messung kann somit definiert werden als das relative Nichtvorhandensein von unsystematischen (zufälligen) Meßfehlern [BAHREBERG, GIESE & NIPPER 1990].

### **2.5.3 Logische Konsistenz**

Die logische Konsistenz stellt den Anspruch der Widerspruchslosigkeit an Daten und deren Modellierung. Hierzu sind die Daten auf unterschiedliche Aspekte zu untersuchen:

- ? Bereichskonsistenz
- ? Formatkonsistenz
- ? Topologische Konsistenz

Im Rahmen der Prüfung der Bereichskonsistenz ist es möglich, Wertebereiche festzulegen, innerhalb derer die Daten zu liegen haben. Grobe Datenfehler - wie oben beschrieben - können damit gezielt aufgefunden und gegebenenfalls eliminiert werden. Neben der Formatkonsistenz, die angibt, inwieweit Daten dem Format der gewählten Datenstruktur

entsprechen, stellt besonders die topologische Konsistenz ein wichtiges Kontrollwerkzeug der Speicherung räumlicher Daten dar. Sie betrifft die Eindeutigkeit und Korrektheit der topologischen Beziehungen zwischen den geometrischen Objekten und ist Voraussetzung für die eindeutige Zuordnung von Attributen zu den geometrischen Objekten in einem Geoinformationssystem [OSWALDER 1996].

In KAINZ (1999) werden die Kriterien für das Erfüllen topologischer Konsistenz anhand von Flächenelementen, die als eine Menge von Knoten, Kanten und Flächen gespeichert sind, gezeigt.

- ? Jede Kante hat einen Start- und einen Endknoten.
- ? In Richtung vom Start- zum Endknoten gibt es zu jeder Kante eine Fläche, die links, sowie eine, die rechts liegt.
- ? Eine alternierend geschlossene Folge von Knoten und Kanten begrenzt jede Fläche.
- ? Eine alternierend geschlossene Folge von Kanten und Flächen begrenzt jeden Knoten.
- ? Kanten schneiden sich nur in Knoten.

Die topologische Konsistenz stellt ein wesentliches Prinzip moderner Geodatenverarbeitung in Geoinformationssystemen dar. So soll etwa gewährleistet sein, dass Straßen nicht "im Nichts" enden, oder dass Wasserflächen geschlossen sind. Die Prüfung topologischer Konsistenz wird anhand festgelegter Regeln vollautomatisch durchgeführt, wofür heute zahlreiche Software-Produkte am Markt erhältlich sind.

## **2.6 Bearbeitung von Datenbanken**

Datenbankmanagementsysteme stellen ein mächtiges Werkzeug dar, um große Datenmengen in strukturierter Form speichern und bearbeiten zu können. Der Datenbestand ist in den meisten Fällen nicht statisch, sondern unterliegt einer permanenten Veränderung. Neue Daten werden in das System aufgenommen, alte eliminiert.

Wie bereits erläutert, erfolgt die Speicherung großer Datenmengen nach wohl definierten Regeln und Normen. Das oben beschriebene, weitverbreitete relationale Datenmodell sorgt dafür, dass sich die importierten Daten in einer klaren Struktur befinden und korrekte Beziehungen zwischen den Tabellen bestehen. Das Wissen um den Aufbau der Datenstruktur ist essentiell für die Arbeit mit einem Datenbanksystem. In der Praxis erfolgt die Bearbeitung der Daten durch den Einsatz von Datenbanksprachen.

### **2.6.1 Datenbanksprachen**

Datenbanksprachen stellen die Kommunikationsschnittstelle eines Datenbankmanagementsystems dar. Sie sind mittels einer genormten Syntax in der Lage, mit diesen zu kommunizieren. Neben der Option, Daten abzufragen, sind das Erzeugen der Datenstruktur sowie seine Manipulation wichtige Funktionalitäten von Datenbanksprachen. Die verschiedenen Datenmodelle weisen Datenbanksprachen unterschiedlicher Komplexität auf.

In hierarchischen und Netzwerkmodellen existieren im Normalfall keine Bearbeitungssprachen im eigentlichen Sinne, sondern lediglich navigierende Operatoren. Diese erlauben es, Daten in diesen Modellen gezielt anzusprechen und zu bearbeiten. SQL, die "Structured Query Language", ist die Lingua franca der Datenbankwelt. Sie stellt den de-facto Standard der Abfragesprachen dar und liegt mittlerweile als ANSI-ISO-Norm (American National Standard Institute-International Organization for Standardization) vor. SQL nutzt in ihrer Syntax Ausdrücke der Relationalalgebra, erweitert um Operationen und Funktionen im Wertebereich. Es muss allerdings bedacht werden, dass SQL in der ursprünglichen Version auf die Bearbeitung relationaler Datenmodelle limitiert ist.

Neben SQL bestehen noch weitere Datenbanksprachen, die der Vollständigkeit halber nicht unerwähnt bleiben sollen, von ihrer Bedeutung her aber in keinster Weise an SQL herantreten. Die Sprache QUEL (QUERy-Language) wurde im Zusammenhang mit dem proprietären Datenbankmanagementsystem INGRES entwickelt und wird auch nur dort eingesetzt.

QBE (Query by Example) unterscheidet sich von den übrigen Abfragesprachen dahingehend, dass sie keine textuelle Sprache ist. Abfragen werden vielmehr durch Einträge in Tabellengerüste formuliert. Der Benutzer gibt Beispieleinträge in Tabellen an, die seinen

Anfragewünschen entsprechen. QBE ist heute in den meisten professionellen Datenbankmanagementsystemen neben SQL implementiert [HEUER & SAAKE 2000].

### 2.6.1.1 Standard Query Language (SQL)

SQL ist die mit Abstand am weitesten verbreitete Datenbanksprache. Sie besteht im wesentlichen aus folgenden Teilsprachen:

- ? Data Definition Language (DDL)
- ? Storage Structure Language (SSL)
- ? Interactive Query Language (IQL)
- ? Data Manipulation Language (DML)

In der Regel zeigt sich, dass für die praktische Arbeit mit Datenbankmanagementsystemen vor allem die DDL zur Erstellung von Datenstrukturen, sowie die DML zur Manipulation des Datenbestandes und zur Bearbeitung von Abfragen relevant sind.

Überblicksmäßig soll die Entwicklung von SQL bis zum derzeit gültigen Standard angeführt werden:

1974 wurde vom IBM-Forschungslabor in San Jose, USA die erste Datenbanksprache - SEQUEL (Structured English QUERY Language) - entwickelt. Im ersten Prototypen eines relationalen Datenbankmanagementsystems, 1976 entwickelt, war eine weiterentwickelte Form - SEQUEL2 - in Verwendung. Die ersten kommerziell verfügbaren relationalen Datenbankmanagementsysteme implementierten eine Untermenge von dieser Sprache und nannten diese SQL. Die Normierung von SQL durch die ANSI (American National Standards Institut) wurde bis 1986 durchgeführt. Die heute gebräuchliche Version SQL-92 wurde 1992 als ANSI- und ISO-Norm freigegeben.

Die in Kap. 2.3.4 angesprochenen objektrelationalen Datenmodelle bedienen sich einer erweiterten Version von SQL - SQL99, die jene dort beschriebenen Methoden und Verfahren der Objektorientierung berücksichtigt. Der Einsatz von Datenbanksystemen in Geoinformations- und Visualisierungssystemen führte zu einer wachsenden Bedeutung von

SQL in der Kartographie und Geoinformation. Die praktische Bearbeitung von Sachdaten sowie Geometriedaten mittels SQL soll im folgenden Abschnitt praktisch erläutert werden.

## 2.6.2 SQL in der Praxis

SQL ist eine textuelle Sprache, die eine natürlichsprachliche Interaktion mit Datenbanken zulässt. Die Sprache weist ein reichhaltiges Repertoire an semantisch fixierten Grundelementen (Wortschatz) und generellen Ausdrucksmitteln (Grammatik) auf [VOSSSEN & WITT 1991].

### 2.6.2.1 Manipulation von Sachdaten

Es soll eine Tabelle erstellt, Inhalte in diese eingefügt und schließlich wieder abgefragt werden. Beispielhaft sollen in die Tabelle die Nationen Europas und deren Hauptstädte aufgenommen werden.

Tabelle erstellen:

```
create table CAPITAL (ID tinyint (4) auto_increment,  
NAME varchar (40) not null,  
NATION varchar (40) not null,  
POPULATION int (8),  
primary key (ID) )
```

Neben den Einträgen der Länder und ihrer Hauptstädte, für die jeweils 40 beliebige Zeichen (*varchar*) zur Verfügung stehen, sowie der Einwohnerzahl wird ein fortlaufender (*auto\_increment*) Primärschlüssel (*ID*) definiert, der jeden einzelnen Datensatz eindeutig identifiziert.

Der nächste Arbeitsschritt besteht darin, die Tabelle mit Attributwerten zu füllen.

Eingabe von Werten:

```
insert into CAPITAL (NAME, NATION, POPULATION)
values ('Wien', 'Österreich', '1700000',
'Paris', 'Frankreich', '9400000')
```

Das Resultat dieser ersten beiden Befehlszeilen würde eine Tabelle mit folgendem Aussehen erzeugen.

<b>ID</b>	<b>NAME</b>	<b>NATION</b>	<b>POPULATION</b>
1	Österreich	Wien	1700000
2	Frankreich	Paris	9400000

Tab.3: Inhalt der Tabelle 'Capital'

Eine derart erstellte Tabelle kann nun über den Primärschlüssel (*ID*) mit beliebigen anderen Tabellen verknüpft werden. Die Möglichkeit, Beziehungen zwischen Sachdaten einer Datenbank herzustellen, wird in einem Geoinformationssystem um die Option erweitert, Attributdaten mit Geometriedaten zu verbinden. So wäre es in diesem Beispiel denkbar, die räumlichen Objekte Wien und Paris mit obiger Attributtabelle zu verknüpfen. Eine Abfrage der Tabelle CAPITAL nach unten angeführtem Schema würde als Resultat Wien liefern.

Abfrage:

```
select * from CAPITAL where POPULATION < 5000000
order by NAME
```

Dieses einfache Beispiel zeigt, wie Geometriedaten mittels einer Abfrage ihrer verknüpften Attributdaten bearbeitet werden können. Der Einsatz großer Geodatenbanken zeigte aber die Notwendigkeit auf, räumliche Daten direkt mit SQL zu manipulieren.

### 2.6.2.2 Manipulation von Geodaten

Tabellen mit Geometriedaten weisen die gleiche Struktur wie Tabellen mit Sachdaten auf. So ist es etwa denkbar, in einer Tabelle die Blockstruktur einer Stadt vektoriell als Polygone, bestehend aus mehreren Linien, abzuspeichern. Die Polygondaten werden in der einfachen Form  $Polygone\{LinienNR, KnotenNR, X, Y\}$  abgelegt. Als Beispiel sei folgende Tabelle dargestellt:

LinienNR	KnotenNR	X	Y
1	101	1	1
1	102	2	2
1	103	1	2
2	201	4	3
2	202	6	3
2	203	6	4
2	204	4	4

Tab.4: Inhalt der Tabelle 'Polygone'

Die Darstellung des Inhalts dieser Tabelle hätte folgendes Aussehen:

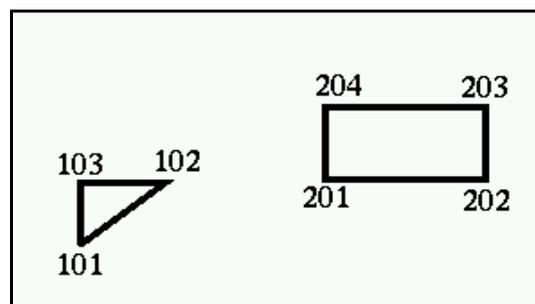


Abb.8: Darstellung der Tabelle 'Polygone'

Aus der Tabelle ist ersichtlich, dass ein Polygon als Summe der Linien gespeichert wird, die benötigt werden, um dieses räumliche Objekt zu bilden. Ein Objekt Linie wird seinerseits durch Punkte (Anfangs- und Endpunkt) gebildet. Diese Punkte besitzen die Merkmale X und Y, die Koordinaten des festgelegten Koordinatensystems sind. Es wäre naheliegend,

Abfragen der Tabelle über diese beiden Spalten vorzunehmen. Eine typische Abfrage des Datenbestandes wäre etwa: "Welche Polygone schneiden ein gegebenes Auswahlrechteck oder liegen vollkollkommen in diesem?"

Zur besseren Anschaulichkeit soll dieser Sachverhalt anhand einer konkreten Datengrundlage - der Wiener Mehrzweckstadtkarte - untersucht werden. Als Polygone sind in diesem Zusammenhang die Blockstrukturen der Häuserblöcke, als Punkte deren Eckpunkte zu verstehen.

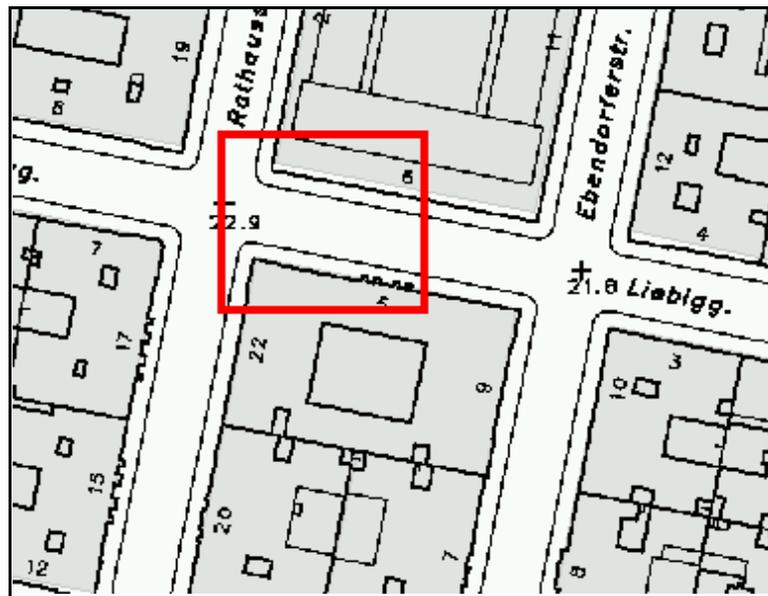


Abb. 9: Rechteckselektion (1)

Die Polygon-Abfrage soll als Ergebnis alle Knoten liefern, deren X-Wert größer als der X-Wert der linken unteren Ecke ( $x_{min}$ ), jedoch kleiner als jener der oberen rechten Ecke ( $x_{max}$ ) ist. Analoges gilt für die Y-Werte. Diese Abfrage, mit SQL durchgeführt, hätte folgendes, allgemeines Aussehen:

```
select KnotenNR from POLYGONE  
where (X > x_min) and (X < x_max) and (Y > y_min) and (Y < y_max)
```

Bei der Positionierung des Auswahlrechtecks wie in Abb. 9 würden zwei Knoten diese Kriterien erfüllen. Von diesen Knoten könnte auf die Linien und in weiterer Folge auf die zugehörigen Polygone geschlossen werden.

Es scheint, als würde SQL die Mächtigkeit besitzen, räumliche Informationen direkt abfragen zu können. Eine kleine Positionsänderung des Auswahlrechtecks zeigt aber auf, dass dies nur bedingt der Fall ist.

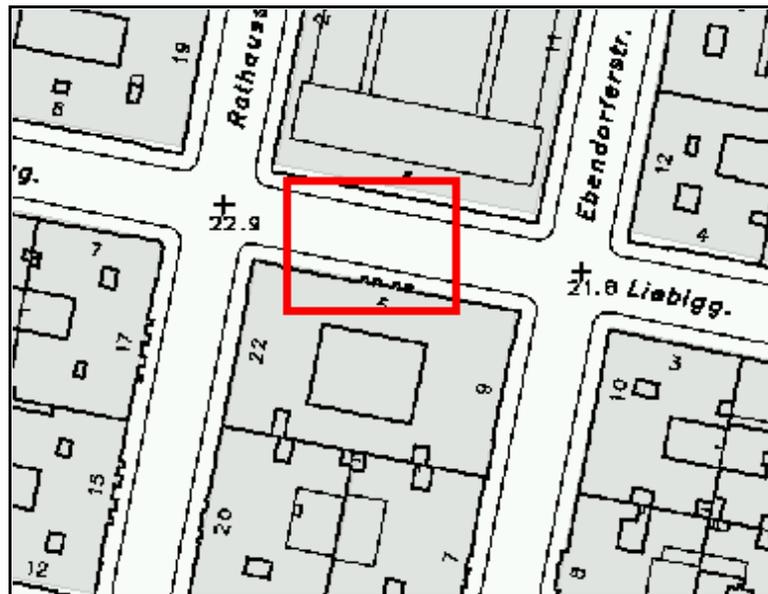


Abb. 10: Rechteckselektion (2)

Die Wiederholung der obigen Abfrage gemäß Abb. 10 würde keinen Knoten innerhalb der Selektion finden, und somit auf die Frage "Welche Polygone schneiden ein gegebenes Auswahlrechteck oder liegen vollkommen in diesem?" eine falsche Antwort, nämlich "keine" liefern.

Eine vollständige Manipulation von Geometriedaten erfordert somit eine Erweiterung des relationalen Datenmodells und eine Ausweitung des derzeit gängigen SQL-Standards.

Einen Überblick über notwendige Maßnahmen gibt BILL (1999):

- ? Unterstützung neuer Datentypen (Punkt, Linie und Polygon)

- ? Unterstützung neuer Operatoren (Schneiden, Ausblenden etc.)
- ? Unterstützung dieser raumbezogenen Datentypen und Operatoren in der Abfragesprache
- ? Unterstützung räumlicher Indizierungen zur Verbesserung der Antwortzeiten bei raumbezogenen Abfragen

Eine Implementierung dieser neuen Funktionen in SQL zeichnet sich ab. Beispielgebend sei die OpenGIS® Consortium-SQL92 Erweiterung genannt, da dieses Projekt weltweit unterstützt wird und demnach beste Chancen hat, allgemeine Gültigkeit und Standardisierung zu erlangen.

### **2.6.3 Die OpenGIS®Consortium-SQL92 Erweiterung**

Das OpenGIS®Consortium ist ein Zusammenschluss praktisch aller namhaften Anbieter von Geoinformationssystemen mit Datenbanksoftware-Herstellern, Computerproduzenten und diversen Institutionen, Universitätsinstituten und Privatpersonen, die im GIS-Umfeld beheimatet sind. In mehreren Arbeitsgruppen dieser Interessensgemeinschaft wird weltweit daran gearbeitet, offene Standards für die Arbeit mit Geometriedaten zu erstellen. In OpenGIS®Guide (2001) wird die Intention des OpenGIS®Consortium als "Definition einer Technologie, die einem Anwendungsentwickler (oder Anwender) ermöglicht, jede Art von geokodierten Daten, Geo-Funktionalitäten oder Prozesse, die im Internet verfügbar sind, innerhalb seiner Umgebung und eines einzelnen Arbeitsablaufs zu nutzen", beschrieben.

Im Rahmen dieses Vorhabens wurde ein Konzept erarbeitet, welches das SQL92-Schema mit relevanten Aspekten der Geoinformation erweitern soll. Ziel der "OpenGIS®Simple Feature Specification For SQL" (1999) ist die Speicherung, Verwaltung, Handhabung und Abfrage von räumlichen Objekten mittels SQL. Deutlich sichtbar ist das Bestreben, Konzepte der Objektorientierung in die Spezifikation einfließen zu lassen. Gemäß der Theorie der Objektorientierung werden Klassen definiert. Diese entsprechen Objekten mit ähnlichen Eigenschaften. Geometrie dient als Hauptklasse, die diverse Methoden wie Dimension, Geometrietyp und "Bounding Box" (umschließende, rechteckige Fläche) aufweist. Im Rahmen der "OpenGIS® Specification" wird strikt zwischen Tabellen, die

Attributdaten enthalten, und jenen mit Geometriedaten unterschieden. Dies ist notwendig, um die Abfrage der Daten mit der jeweils entsprechenden Syntax durchzuführen. Attributdaten werden weiterhin mit SQL92 bearbeitet, räumliche Daten dagegen mit der erweiterten Form. Diese bietet die Möglichkeit, folgende räumliche Beziehungen zwischen geometrischen Objekten zu testen.

<b>Methode</b>	<b>Verhalten</b>
Equal	Objektgeometrie räumlich <i>gleich</i> zu anderer
Disjoint	Objekt räumlich <i>getrennt</i> zu anderem
Intersect	Objekt <i>schneidet</i> anderes räumlich
Touches	Objekt <i>berührt</i> anderes räumlich
Crosses	Objekt <i>kreuzt</i> anderes räumlich
Within	Objekt ist räumlich <i>innerhalb</i> eines anderen
Contain	Objekt <i>beinhaltet</i> anderes räumlich
Overlap	Objekt <i>überlagert</i> anderes räumlich
Relate	Objekt ist mit anderem räumlich <i>verbunden</i>

Tab. 5: Methoden zum Testen räumlicher Beziehungen gemäß  
OpenGIS®Consortium-SQL92 Erweiterung

Neben dem Testen der diversen Beziehungen zwischen Objekten soll es möglich sein, räumliche Analysen durchzuführen. Folgende Methoden sollen zur Verfügung stehen:

<b>Methode</b>	<b>Verhalten</b>
Distance	Kürzeste Distanz zwischen zwei Punkten
Buffer	Punkte innerhalb einer Pufferzone
Convex Hull	Konvexe Hülle um Objekt
Intersection	Kreuzungspunkte zweier Objekte
Union	Vereinigungsfläche zweier Objekte
Difference	Differenzfläche zweier Objekte
Symmetric Difference	Gleichmäßige Differenzfläche zweier Objekte

Tab. 6: Methoden zur räumlichen Analyse gemäß  
OpenGIS®Consortium-SQL92 Erweiterung

Sowohl die erwähnten Testmöglichkeiten räumlicher Beziehungen wie auch die Methoden zur räumlichen Analyse stellen Prinzipien dar, die als Standardanwendungen in Geoinformationssystemen gelten können. Neu ist allerdings der Ansatz, diese gänzlich in einer Datenbank mittels einer standardisierten Datenbanksprache durchzuführen. Das Datenbanksystem erfährt somit eine Isolierung von den übrigen Komponenten eines kartographischen Visualisierungssystems. Dies führt zu einer Systemarchitektur, die deutlich besser zu handhaben ist, da sie modular aufgebaut ist (vgl. Kap. 6)

## **3 Kartographie**

Anhand graphischer Kommunikationsmodelle soll gezeigt werden, welche Bedeutung Datenbanksysteme und das Internet in der heutigen Kartographie haben. Weiters sollen der modellhafte Ablauf der Erstellung eines kartographischen Modells sowie Beispiele aus Österreich gezeigt werden. Wesentliche Aspekte der digitalen Stadtkartographie bilden den Abschluss dieses Abschnitts.

### **3.1 Kartographische Visualisierung im Wandel**

HAKE und GRÜNREICH (1994) beschreiben die Kartographie als "ein Fachgebiet, das sich befasst mit dem Sammeln, Verarbeiten, Speichern und Auswerten raumbezogener Informationen sowie in besonderer Weise mit deren Veranschaulichung durch kartographische Darstellungen." Bis vor wenigen Jahren war die "kartographische Darstellung" gleichbedeutend mit der Visualisierung der Informationen auf einer Karte. Obwohl es daneben durchaus kartographische Produkte wie Panoramen oder Blockbilder - die wir heute gemeinhin als kartenverwandte Ausdrucksformen bezeichnen - sowie taktile Beispiele gab, war das verkleinerte, vereinfachte, orthogonale Abbild eines Teiles der Erdoberfläche als analoge, gedruckte Karte das Hauptprodukt der Kartographie.

Diese Situation hat sich durch technische Innovationen in den letzten Jahren - als Schlagworte seien exemplarisch Personal Computer, Internet und Datenbanken genannt - deutlich geändert. Besonders das Internet bietet dem Kartographen von heute Möglichkeiten, die noch vor wenigen Jahren undenkbar waren. PETERSON (1997) nennt als Gründe, die für das Internet als Publikationsort kartographischer Darstellungen sprechen, geringe Kosten, Zeitersparnis, Interaktionsmöglichkeiten und die Möglichkeit der Einbindung von Animationen. Dies mögen durchaus plausible Argumente sein, digitale Kartographie der analogen vorzuziehen. Demgegenüber steht allerdings die Tatsache, dass das Internet derzeit noch keine Qualitätssteigerung der kartographischen Darstellung gegenüber einer "klassischen" gedruckten Karte bieten kann. Dies ist leicht nachvollziehbar, wenn man um die Limitierungen bei der Visualisierung von Graphiken am Bildschirm weiß. Neben der geringeren Auflösung und den damit verbundenen graphischen Problemen sei

hier besonders der limitierte Platz eines Bildschirms gegenüber einer gedruckten Karte zu erwähnen.

Wenn somit die Steigerung der Qualität gegenüber den Produkten der traditionellen Kartographie nicht möglich ist, wodurch ist der Erfolg der Internet-Kartographie zu erklären? Die Ausweitung der kartographischen Produktpalette infolge der Technologisierung scheint eine mögliche Antwort zu sein. Neben der gedruckten Karte gibt es heute eine Vielzahl von Möglichkeiten, räumliche Information zu präsentieren. Informationen mit Raumbezug sind nicht mehr auf die alleinige Visualisierung mittels einer statischen Karte beschränkt, sondern können neue Formen kartographischer Präsentation annehmen. Welche Präsentationsform sich am besten eignet, hängt direkt mit der Information, die es zu visualisieren gilt, zusammen.

Einen Überblick, wie Informationen am besten vermittelt werden, bietet WEIDENMANN (1988):

- ? **Räumliche Strukturen** werden am besten mittels Karten und kartenähnlichen Darstellungen abgebildet.
- ? **Abstrakte Strukturen** wie qualitative und quantitative Beziehungen werden sehr gut durch bildhafte Darstellungen in Form von Graphiken und Diagrammen vermittelt.
- ? **Dynamische Sachverhalte**, wie etwa räumliche Prozesse, werden positiv durch eine dynamische Darstellung mittels Animation oder Simulation dieser Sachverhalte beeinflusst.
- ? **Logische Beziehungen** und Schlussfolgerung werden am besten durch verbale Präsentation mittels Sprache und Text erzeugt.

Es bleibt somit festzuhalten, dass Karten zur Visualisierung räumlicher Sachverhalte absolute Berechtigung haben, jedoch daneben andere Formen bestehen, die mit den Mitteln der analogen Kartographie nicht oder nur unzureichend dargestellt werden können.

Die vielseitigen technischen Möglichkeiten, die der Kartographie heute geboten werden, stellen somit keineswegs eine Ablöse, sondern vielmehr eine Erweiterung der bisherigen kartographischen Visualisierungsformen dar.

## 3.2 Kartographische Kommunikation im Wandel

Im vorhergehenden Abschnitt wurde beschrieben, wie die laufenden technischen Innovationen, insbesondere das Internet, einen Wandel der kartographischen Visualisierungsmöglichkeiten herbeiführen. Die Möglichkeiten der Präsentation räumlicher Information im Internet bringen eine Veränderung der Kartographie als Kommunikationssystem mit sich, die es in der Folge zu erläutern gilt.

### 3.2.1 Lineares Modell

Die gedruckte kartographische Darstellung ist eine Vereinigung von Datenspeicher und Datenpräsentation. Mit dem Druck der Karte sind die inhaltlichen und konzeptionellen Überlegungen abgeschlossen, eine nachträgliche Änderung ist nicht erwünscht, oftmals auch nicht möglich [KRIZ 1999]. Dieses Modell des Informationstransports vom Kartenhersteller zum Kartennutzer galt lange Jahre als Theorie der Kartographie und wurde als gegeben hingenommen. Zur graphischen Darstellung dieses Sachverhaltes bediente man sich eines Modelles der Informationstheorie. In diesem linearen Flußdiagramm der Informationsübertragung beschrieb man den Weg der Information vom Sender durch einen Kanal zum Empfänger [ROBINSON & PETCHENIK 1975].

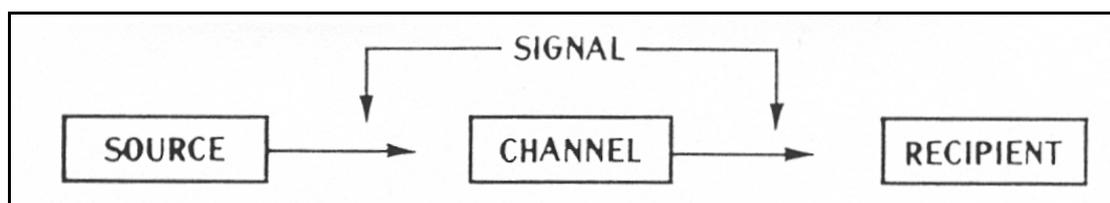


Abb.11: *Modell der linearen kartographischen Kommunikation*  
(ROBINSON & PETCHENIK 1975)

Der Nutzer der Information hat keinerlei Einfluss auf die Gestaltung der Darstellung, da das Modell nur einen gerichteten, linearen Weg von Hersteller zum Nutzer vorsah, nicht jedoch die Reflexion.

Die Entwicklung, Datenspeicherung und -visualisierung zu trennen, führte zu einer neuen, flexibleren Struktur des Modells, die Interaktion des Informations- und somit Kartennutzers und dem Hersteller der Karte ermöglicht.

### **3.2.2 Dialogisches Modell**

1968 stellte KOLACNY eine Erweiterung des linearen Modells vor, das Prozessmodell der Kommunikation kartographischer Informationen. Dieses erlaubte eine vielseitige Anpassung, da es neben Begriffen der Informationstheorie auch Aspekte der Kommunikations- und anderer Sozialtheorien enthält. Die technischen Entwicklungen bis hin zum heutigen Stand waren damals noch nicht absehbar, dennoch gelang es KOLACNY, viele Konzepte und Prinzipien der heutigen "modernen" Kartographie vorauszusehen.

FREITAG (2000) modifizierte das Modell dahingehend, dass er gegenwärtige Entwicklungen der Kartographie miteinbezog und so ein zeitgemäßes Modell der dialogischen kartographischen Kommunikation schuf. Die Basis des Modells stellt noch immer den Informationsfluss vom "Kartenmacher" zum "Kartennutzer" dar. Erweitert wird es aber mit der Option, das "Wissen des Kartennutzers" mit dem "Wissen des Kartenmachers" zu verschneiden. Es entsteht somit ein Raum, in dem Hersteller und Nutzer kommunizieren können und der die Möglichkeit bietet, Ziele und Wünsche des Kartennutzers in den kartographischen Produktionsablauf einzubinden.

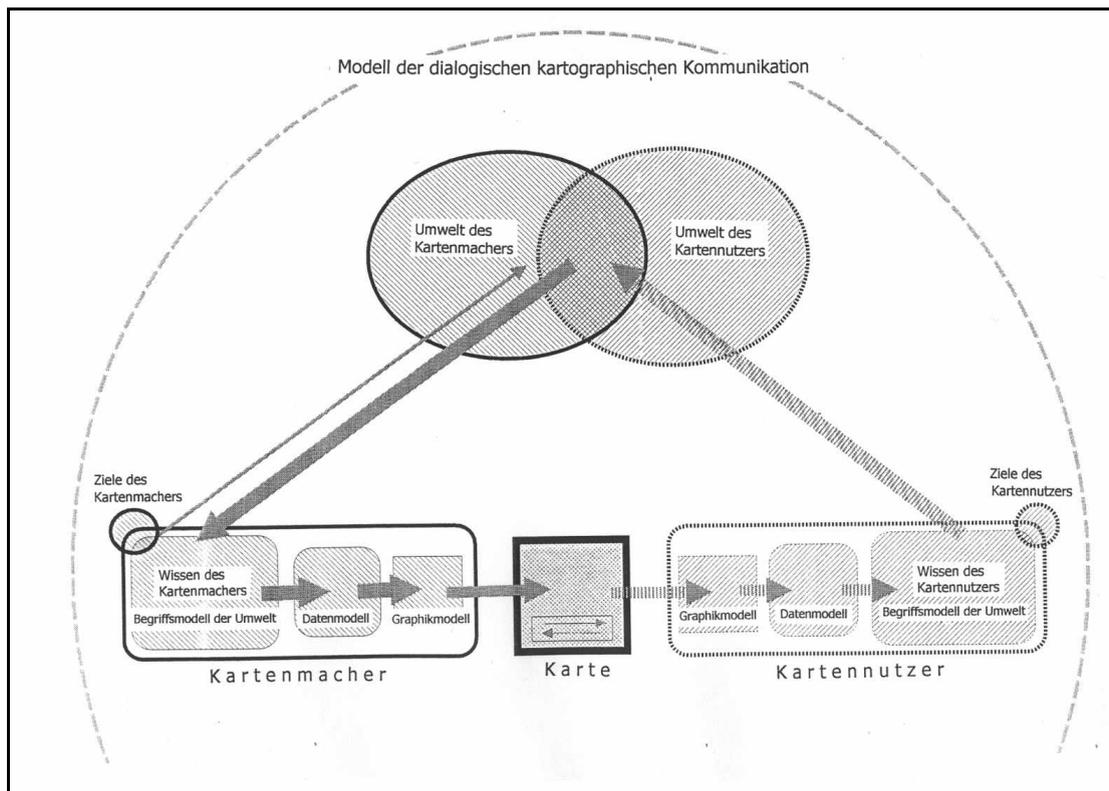


Abb.12: *Modell der dialogischen kartographischen Kommunikation*  
(FREITAG, nach KOLACNY 2000)

Aus einem linearen System, in dem die inhaltlichen und konzeptionellen Überlegungen einseitig und limitiert sind, wird ein vernetztes System des offenen Dialogs zwischen Experte und Nutzer. Die Tatsache, dass der Kartennutzer Einfluss auf die Entwicklung einer Karte nehmen kann, ändert aber nichts an der Tatsache, dass die eigentliche Produktion auf Seiten des Kartenmachers verbleibt.

### 3.2.3 Monologisches Modell

FREITAG (2000) geht in einer zusätzlichen Modifizierung KOLACNYs Modell auf eine weitere Entwicklung der gegenwärtigen Kartographie ein - der Verschmelzung von Kartenmacher und Kartennutzer.

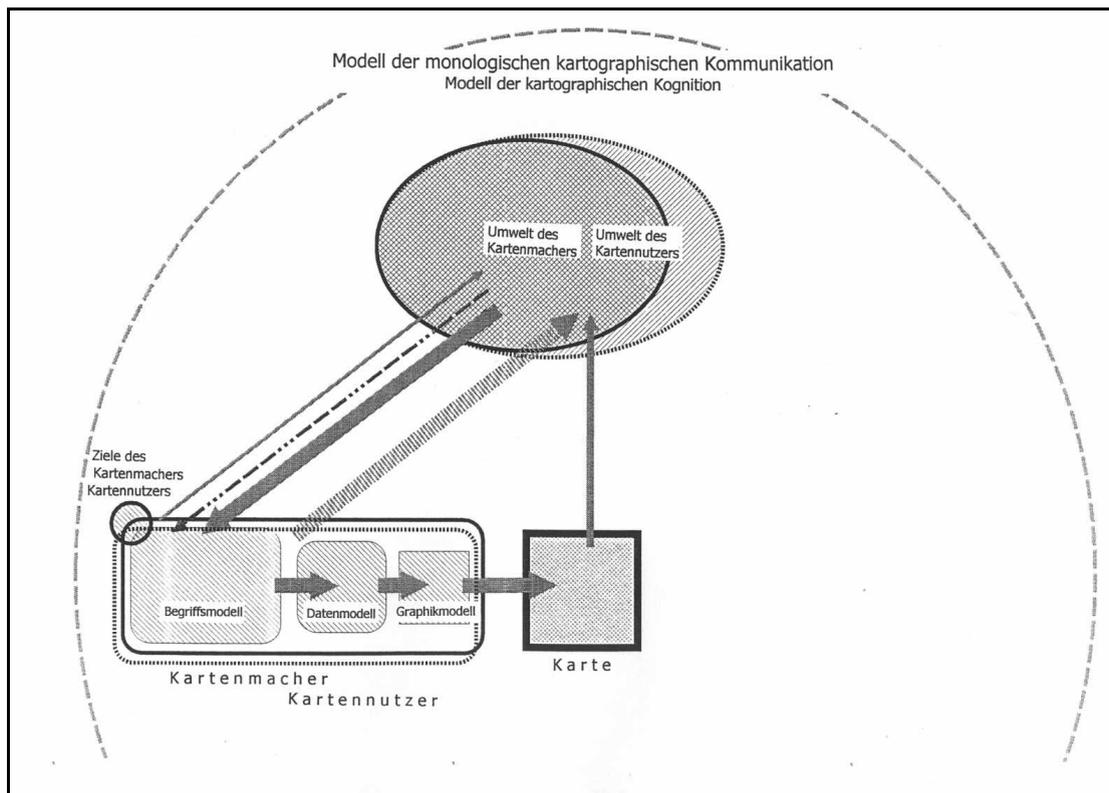


Abb.13: *Modell der monologischen kartographischen Kommunikation*  
(FREITAG, nach KOLACNY 2000)

Kartographische Applikationen im Internet bieten dem Nutzer heute die Möglichkeit, Aussehen und Gestaltung der Visualisierung aktiv zu beeinflussen, ohne den direkten Kontakt mit dem Kartenhersteller zu suchen. Dies verlangt ein System, das sowohl von Hersteller- als auch von Nutzerseite volle Zugriffsmöglichkeit auf die Gestaltungs- und Manipulationswerkzeuge als auch die Daten bietet. Datenbankgestützte Visualisierungssysteme bieten diese Möglichkeiten in hohem Maße.

Aus einem dialogischen wird ein monologisches System. Die "Umwelt des Kartenschaffenden" ist vollständig in der "Umwelt des Kartennutzers" enthalten, da es sich um ein und die selbe Person handelt. Das Modell der monologischen kartographischen Kommunikation ist, ähnlich wie in den Ausführungen zum Wandel der kartographischen Visualisierung, nicht als Substitution der bisherigen Verfahren zu sehen, sondern bietet vielmehr eine Möglichkeit, auf die veränderten Anforderungen der digitalen Kartographie modellhaft einzugehen.

Die oben gezeigten Modelle der kartographischen Kommunikation stellen keineswegs den letzten Stand der Forschung dar. Neben diesen, vor allem in den 70er Jahren des 20. Jahrhunderts weit verbreiteten, entstanden in den letzten Jahren Modelle der Theorie der Kartographie, die neben der Kommunikation auch die Aspekte der Kognition und des Formalismus berücksichtigen. Im Zentrum dieser Modelle steht vielfach der Begriff der Visualisierung [TAYLOR 1996]. DiBIASE zeigt 1990 ein "Rechteck des geowissenschaftlichen Forschungsprozesses", das den Zusammenhang von visuellem Denken im privaten Bereich und visueller Kommunikation im öffentlichen Bereich darstellt. TAYLOR erweitert diese Gedanken 1991 zu einem "Dreiecksdiagramm der neuen konzeptionellen Grundlage der Kartographie". McEACHREN entwickelt 1994 einen "Kartographiewürfel" als graphisches Modell einer modernen Theorie der Kartographie. Dieser ist in der Lage, durch die Anordnung seiner drei Achsen - öffentliche-private Kartennutzung, geringe-intensive Wechselbeziehung zwischen Karte und Nutzer, bekannte-unbekannte Daten - jede Form kartographischen Wirkens darzustellen [FREITAG 2000]. Die ausführliche Behandlung der kartographischen Kommunikationsmodelle im Rahmen dieser Arbeit wurde durchgeführt, da diese die Trennung von Datenspeicherung und Datenvisualisierung, wie sie heute Standard ist, am besten wiedergeben.

### **3.3 Vom Geomodell zum kartographischen Modell**

Sowohl im dialogischen wie auch im monologischen Modell der kartographischen Kommunikation ist die Trennung des Datenmodells vom Graphikmodell und der Karte deutlich zu erkennen. Es erfolgt dadurch die bereits erwähnte Trennung zwischen Speicherung und Visualisierung der Information. Diese Vorgehensweise ist heute vielfach in der Praxis des kartographischen Produktionsprozesses zu finden und soll daher eingehend diskutiert werden.

#### **3.3.1 Geomodell**

Ein Geomodell (auch digitales Objektmodell) beinhaltet das Abbild der Erdoberfläche nach topographischen Gesichtspunkten. Topographische Erscheinungsformen und Sachverhalte der Landschaft werden erhoben, klassifiziert und gespeichert. Hierfür ist es notwendig,

topographisch relevante Umweltobjekte auf klar definierte Elemente zu reduzieren, um sie als solche speichern zu können. In den Kommunikationsmodellen FREITAGSs stellt dieser Schritt den Übergang vom Begriffsmodell zum Datenmodell dar (vgl. Abb. 12, 13).

Informationen in einem Geomodell sind nicht durch kartographische Bearbeitung verändert worden, sondern liegen als reines, maßstabsunabhängiges Vektordatenmodell vor. Der Einsatz von Rasterdaten ist prinzipiell möglich, um etwa Datenlücken zu schließen, wird aber weitestgehend vermieden, da es nicht möglich ist, diese in ausreichender Form zu attributieren. Die Verwendung von Rasterinformationen beschränkt sich somit vielfach auf die Hinterlegung der Vektordaten zur besseren Orientierung.

In der Vergangenheit sind vermehrt geometrische Datenmodelle entwickelt worden, die sich überwiegend an graphischen Kartendarstellungen orientierten. So hat das digitale Geomodell für HAKE & GRÜNREICH (1994) "für fachthematische Geo-Daten die gleiche Funktion wie der topographische Kartengrund in der klassischen thematischen Kartographie". Diese Modelle wiesen, aus heutiger Sicht, nur eine beschränkte Leistungsfähigkeit auf. Die Bedeutung von Geomodellen geht heute über die Funktion eines digitalen Kartengrundes hinaus. Die geometrische Modellierung - die Beschreibung, Bearbeitung und Speicherung der zugrundeliegenden Geometrie von raumbezogenen Objekten, in dem analytische und approximierende Verfahren eingesetzt werden - stellt heute einen wesentlichen Faktor beim Aufbau geometrischer Datenmodelle dar [BILL & FRITSCH 1991].

Geomodelle sind in der Regel attributorientiert aufgebaut. Die Landschaft wird demnach grob nach Objektarten, fein nach Attributen gegliedert. Beim Aufbau eines Geomodells bedient man sich heute zumeist des objektorientierten Datenmodells (vgl. Kap. 2.3.5). Als kleinstes Element dieses Modells fungiert das Objekt, ein konkreter, geometrisch begrenzter, durch Attribute beschriebener und mit Namen versehener Teil der Landschaft [PAUL 1997]. Jedes Objekt besitzt einen eindeutigen Objektidentifikationsschlüssel, der Verknüpfungen zu anderen Daten ermöglicht.

Das Geomodell unterliegt einer Erfassungsgeneralisierung. Dies entspricht geometrisch einer Vereinfachung der in der Realität erfassten Objekte - etwa Rundung von Nachkommastellen - sachlich einer Bildung von Objektklassen ähnlicher Objekte während der Datenerfassung. Die Erfassungsgeneralisierung ist vor der Speicherung der Daten im Geomodell vom Topographen durchzuführen [HAKE & GRÜNREICH 1994].

### 3.3.2 Thematisches Modell

Im Gegensatz zu der in Geomodellen gespeicherten Geometrie dient das thematische Modell (auch semantisches Modell) der Beschreibung, Bearbeitung und Speicherung der zugrundeliegenden Thematik eines räumlichen Objektes. Erst durch die Verknüpfung der abstrakten Geometriedaten mit ihren beschreibenden Attributen aus dem thematischen Modell ergibt sich so eine sachlogische Beziehung, die eine sinnvolle Darstellung erst möglich macht [WILMERSDORF 1992].

Das thematische Modell ist ebenso wie das Geomodell abstrakt, es kann als Zwischenmodell hin zum kartographischen Modell angesehen werden. Der Aufbau eines thematischen Modells ist von der Art der Anwendung abhängig, da diese die Struktur und Gliederung des Modells vorgibt. In der Praxis zeigen sich allerdings grundlegende Konzepte, die sich bei der Erstellung thematischer Modelle als zielführend erwiesen haben [BILL & FRITSCH 1991]. Die grobe Gliederung des Geomodells erfolgt in einer hierarchischen Struktur, die der Datenkatalog (auch Objektartenkatalog) beschreibt. In ihm wird eine Klassifizierung der Umwelt nach fachspezifischen Gesichtspunkten durchgeführt, wobei nach dem Prinzip der semantischen Ähnlichkeit der Objekte vorgegangen wird. Gleichartige Objekte werden demnach zu Objektarten, diese wiederum zu Objektgruppen und schließlich zu Objektbereichen zusammengefasst [HAKE & GRÜNREICH 1994]. Die Struktur des Datenkatalogs ist zumeist offen gestaltet, um das Hinzufügen weiterer Elemente auf jeder Ebene des Modells möglich zu machen. Dieser Aufbau erlaubt es, einzelne Bereiche des Modells zu extrahieren und somit sachlogische Teilmodelle zu schaffen.

Neben der Einteilung in Objektarten weisen alle Objekte Attribute auf, die Werteausprägungen bestimmter Merkmale darstellen (vgl. Kap. 2.4.2). Somit ist es möglich, Selektionen nicht nur auf Basis der Objektarten, sondern bis zur Ebene der Objektattribute durchzuführen. Die Erstellung des Datenkatalogs sowie einer Attributübersicht hat vor dem Aufbau des Geomodells zu erfolgen, um Informationen zielgerichtet in das Modell einfügen zu können.

Ebenso wie das Geomodell unterliegt auch das thematische Modell einer Vereinfachung. Die thematische Modellgeneralisierung entspricht der oben erläuterten Struktur, in der thematisch ähnliche Objekte zu übergeordneten Objektgruppen zusammengefasst werden. Sowohl Geo- als auch thematische Modelle werden heute in modernen

Datenbankmanagementsystemen verwaltet. Dies bietet die Möglichkeit, Datenredundanz bewusst zu vermeiden, lässt aber die Möglichkeit offen, die Modelle beliebig zu erweitern.

Zusammenfassend seien die Arbeitsschritte zum allgemeinen Aufbau eines räumlichen Datenmodells - des Geomodells samt thematischem Modell - angeführt [HAKE & GRÜNREICH 1994]:

- ? Klassifizierung aller relevanten Umweltobjekte entsprechend einem Objektartenkatalog
- ? Bildung der Geo-Objekte mit Festlegung der geometrischen und semantischen Information sowie der expliziten Relationen
- ? Objektweise digitale Erfassung der Geo-Daten
- ? Bildung der Objektrelationen
- ? Konsistenzprüfung
- ? Speicherung des Modells in einer Datenbank

Die so gespeicherten und attribuierten Objekte bieten nun die Möglichkeit der kartographischen Visualisierung in einem kartographischen Modell.

### **3.3.3 Kartographisches Modell**

Ziel der Erstellung eines kartographischen Modells ist es, aus dem abstrakten Geomodell, nach Verknüpfung mit Sachdaten über ein thematisches Modell, ein kartographisches, darstellungsorientiertes Modell zu schaffen [WILMERSDORF 1992]. GRÜNREICH (1992) sieht die Aufgabe der Visualisierung des thematischen Modells pragmatisch darin, "die im Computer gespeicherten Modelle sowie die aus Analyse- und Verarbeitungsprozessen hervorgehenden Ergebnisse in eine für den Menschen verständliche Form zu übertragen." Um diesem Wunsch gerecht zu werden, müssen beim Übergang vom thematischen zum kartographischen Modell vor allem zwei wesentliche Aspekte berücksichtigt werden, die Symbolisierung und die kartographische Generalisierung.

Die Symbolisierung sieht vor, die abstrakten, attribuierten Objekte in eine adäquate graphische Form zu überführen. Die hierfür verwendeten Signaturen entstammen in der

Regel einem Zeichenschlüssel, der die einheitliche semantische Visualisierung der Objekte gewährleisten soll. Qualitativ gleiches soll demnach überall auch gleich dargestellt werden. Der Zeichenschlüssel als Summe der zu verwendenden Signaturen hilft einerseits dem Bearbeiter, die richtige Anwendung der Zeichen sicherzustellen, andererseits wird dem Nutzer dadurch der richtige Gebrauch des kartographischen Produkts erleichtert [PAUL 1997].

Beim Übergang vom thematischen Modell zum kartographischen Modell erfolgt eine Abbildung des maßstabslosen Datenmodells auf ein maßstabsgebundenes, graphisches Modell. Es ist möglich, aus einem thematischen Modell mehrere kartographische Modelle unterschiedlichen Maßstabes abzuleiten. Eine Ableitung, nur durch Änderung des Maßstabsverhältnisses, würde allerdings zu kartographisch ungenügenden Resultaten führen. Es ist notwendig, die Objektdichte an den gewünschten Maßstab anzupassen. Diesem Problem nimmt sich die kartographische Generalisierung an. Mittels der elementaren Vorgänge der Generalisierung - Vereinfachen, Vergrößern, Verdrängen, Zusammenfassen, Auswählen, Klassifizieren, Bewerten - wird ein, dem gewünschten Maßstab angepasstes, graphisches Visualisierungsmodell erstellt [HAKE & GRÜNREICH 1994].

### **3.4 Digitale Geodatenmodelle in Österreich**

Nach den theoretischen Grundlagen der Erstellung kartographischer Datenmodelle soll eine Beschreibung digitaler Datenbestände in Österreich, die sich zur kartographischen Visualisierung im Internet eignen, erfolgen. Hierbei soll der Schwerpunkt im Allgemeinen auf Darstellungen im großmaßstäbigen Bereich, insbesondere im Einsatz digitaler Stadtkartographie liegen.

#### **3.4.1 Bundesamt für Eich- und Vermessungswesen**

##### **3.4.1.1 Digitales Landschaftsmodell (DLM)**

Das Digitale Landschaftsmodell von Österreich wird von der Abteilung "Landschaftsinformation" des Bundesamtes für Eich- und Vermessungswesen (BEV) aufgebaut. Es stellt ein in einem einheitlichen Raumbezug definiertes topographisches Modell von Österreich dar. Per Definition kann es somit als Geomodell bezeichnet werden, da es allen dort aufgezeigten Prinzipien entspricht. So wird das DLM von Österreich kartographisch nicht bearbeitet - die Daten liegen in originärer Form vor - und es unterliegt einem hierarchischen, objektorientierten Aufbau.

Die Modellierung des DLM erfolgt derzeit zweidimensional, wobei einzelne Objekte Höhenattribute aufweisen. Durch Kombination mit Daten der Geländehöhenbank des BEV ist es aber möglich, dreidimensionale Modelle zu erstellen. Die Geländehöhenbank wird als Objektbereich "Gelände" unabhängig von den restlichen Objektbereichen - Verkehr, Siedlung, Raumgliederung, Gewässer, Bodenbedeckung, Namen - geführt [PAUL 1997]. Die erste Realisierungsphase - die Erfassung der Straßen und Eisenbahnen - wurde 1994 abgeschlossen. Die Erfassung der Gewässer und des Namengutes ist seit Anfang 1997 fertig. Die Verwaltungsgrenzen bis zur Katastralgemeinde liegen ebenfalls vor [SCHUBERT 1997]. Um die Datenmenge des DLM nicht zu einer Größe anwachsen zu lassen, die ein praktisches Arbeiten erschweren würde, orientiert man sich daran, den Inhalt der Österreichischen Karte 1:50.000 (ÖK50) abzudecken.

Die Aktualisierung des Datenbestandes erfolgt sowohl laufend als auch in Form einer flächenhaften Aktualisierung. Die laufende Aktualisierung betrifft Objektarten, die aufgrund ihrer Bedeutung kontinuierlich zu aktualisieren sind. Zu nennen wären in diesem Zusammenhang zuallererst das übergeordnete Verkehrsnetz sowie das Gewässernetz. Die flächenhafte Aktualisierung kann heute als Nachfolge der Periodischen Kartenfortführung angesehen werden. Zielsetzung ist es, alle Objekte des DLM in regelmäßigen zeitlichen Intervallen mit dem Originalstand zu vergleichen [PAUL 1997].

Das Digitale Landschaftsmodell von Österreich ist maßstabsunabhängig und kann somit - theoretisch - im großmaßstäbigen Bereich eingesetzt werden. Im praktischen Einsatz zeigt sich allerdings, dass die fehlende Grundrisstreue der Objekte dies kaum möglich macht. Der strukturierte, hierarchische Aufbau des Modells macht es aber möglich, einzelne Objektarten - etwa "sonstige Straßen" - zu selektieren und diese als Zusatzinformation in großmaßstäbigen Modellen einzugliedern.

### 3.4.1.2 Digitale Katastralmappe (DKM)

1987 begann das Bundesamt für Eich- und Vermessungswesen mit dem Aufbau der Digitalen Katastralmappe von Österreich. Ziel war es, die analoge Katastralmappe digital vorliegen zu haben, um die Aktualisierung und Fortführung dieser zu erleichtern. Die digitale Erfassung erfolgte im Blattschnitt der analogen Vorlagen, im Maßstab 1:1.000. Im Zuge der Erstellung der DKM wurde nicht nur das analoge Mappenbild digitalisiert, sondern es wurden zusätzlich eine Reihe qualitätsverbessernder Maßnahmen - etwa die Berücksichtigung zusätzlicher Grenzpunkte aus der Koordinatendatenbank des BEV - durchgeführt.

Die digitalisierte analoge Katastralmappe stellt ein vektorielles Geomodell, im strengen Sinne, dar. Verortet werden Grundstücksgrenzen, Grundstücksnummern, Benützungsarten, Gebäude, Grenzpunkte sowie amtliche Vermessungspunkte. Diese graphischen Elemente besitzen zunächst noch keinerlei Attribute. Um die Vorteile der digitalen Speicherung voll nutzen zu können, war es notwendig, die einzelnen Objekte der DKM zu attributieren. Diese zusätzliche thematische Information ermöglicht heute diverse Analyseoperationen - etwa Ermittlung der Grundstücksgröße, Nachbargrundstücke, Nutzflächenbestimmung - auf Basis der DKM [KOLB & STURM 1998].

Die Erfassung und Aktualisierung der Digitalen Katastralmappe erfolgt heute dezentral in den Vermessungsämtern Österreichs.



Abb.14: Ausschnitte aus der Digitalen Katastralmappe (BEV)

## 3.4.2 Stadt Wien

### 3.4.2.1 Räumliches Bezugssystem Wien (RBW)

Mitte der 60er Jahre begann die Stadt Wien mit den Vorbereitungen zur Erfassung digitaler Informationen des Stadtgebietes. Als erste Maßnahme wurden die Blockabgrenzungen sowie Straßenachsen aus der analog vorliegenden Stadtkarte digitalisiert. Es stand damit ein räumliches Bezugssystem von Wien zur Verfügung. Um eine Qualitätssteigerung zu erreichen, wurden in einem nächsten Schritt die Luftbildauswertungen der MA41-Stadtvermessung digitalisiert und in das RBW integriert [BELADA 1994]. Die Blockabgrenzungen sowie das Straßennetz werden bis heute laufend aktualisiert und stehen dem interessierten Kunden damit auf neuestem Stand zur Verfügung.

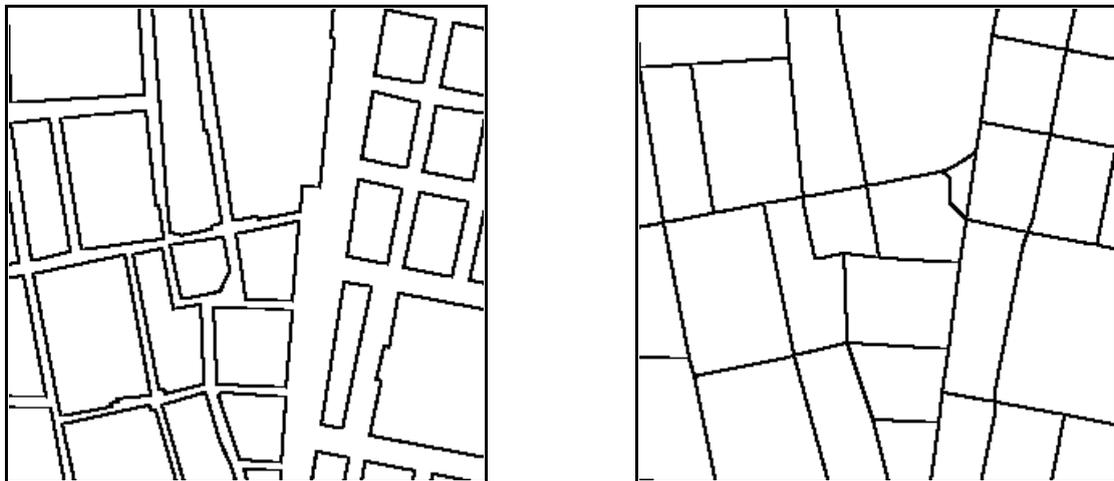


Abb.15: Ausschnitte aus dem Räumlichen Bezugssystem Wien  
(Links: Blockabgrenzungen; Rechts: Straßenachsen) (MA 41-Stadt Wien)

Die Genauigkeit des RBW ist mit jener der analogen Stadtkarte zu vergleichen. Die vielfach gewünschten Genauigkeitsanforderungen, die etwa im Straßenbereich im Zentimeterbereich liegen sollten, machten es notwendig, ein völlig neues Projekt, die Mehrzweckstadtkarte, in Angriff zu nehmen.

### 3.4.2.2 Mehrzweckstadtkarte (MZK)

Zielsetzung der Erstellung der Mehrzweckstadtkarte war es, das gesamte Stadtgebiet Wiens flächendeckend mit einer Genauigkeit und einem Detailreichtum abzubilden, die über jener der Stadtkarte und des RBW lag. Zusätzlich sollte eine Möglichkeit geschaffen werden, die Fortführung des Wiener Stadtkartenwerkes effizienter als bislang zu gestalten.

Die Realisierung der Mehrzweckstadtkarte sollte folgenden Anforderungen genügen [BELADA 1994]:

- ? Einheitliche Darstellungen
- ? Verwendung in großen Maßstäben, wie auch in Übersichtsmaßstäben
- ? Darstellung unterschiedlicher Inhalte auf Anforderungen
- ? Rasche Datenerfassung
- ? Datenerfassung in homogener Form
- ? Erweiterungsfähige Struktur der Datenbank
- ? Konsistenz der Daten bezüglich Strukturänderungen
- ? Konsequente Kennzeichnung der Daten in Form von Metadaten
- ? Rasche Fortführung

Um diesen Zielen zu entsprechen, wurde mit zwei verschiedenen Erfassungsquellen gearbeitet. Das gesamte Stadtgebiet, mit Ausnahme der großen Grünflächen, wurde im Straßenbereich terrestrisch vermessen. Informationen innerhalb der Blockstrukturen wurden durch Luftbildauswertung erfasst. Die graphischen Informationen aus beiden Verfahren werden halbautomatisch zusammengeführt und ergeben somit das topographische Modell. Im parallel dazu erstellten thematischen Modell werden die Attribute der Geoobjekte gespeichert. Die konsequente und exakte Attributierung der Objekte macht es heute möglich, Inhalte der MZK sowohl nach Erfassungsmethode als auch nach thematischen Gesichtspunkten zu visualisieren. Zur Visualisierung der MZK werden die vektoriiellen Informationen kartographisch bearbeitet, etwa durch Hinzufügen von Namengut. Die Neuvermessung des verbauten Stadtgebietes und die Implementierung der Daten in das Geoinformationssystem der MA-41 (Stadtvermessung) waren 1998 abgeschlossen. Seit diesem Zeitpunkt wird die Mehrzweckstadtkarte laufend aktualisiert.

Die Ausgabe der Mehrzweckstadtkarte kann in analoger wie in digitaler Form geschehen, wobei die digitale Verarbeitung der Daten bereits die wichtigste Nutzung darstellt. Die digitale Ausgabe kann sowohl in Vektor- als auch in Rasterformat erfolgen. In regelmäßigen Intervallen werden Rasterungen des gesamten Stadtgebietes durchgeführt, wobei diese die fortgeführten Informationen bis zum Stichtag der Anfertigung enthalten. Als Ausgabemaßstab wird derzeit 1:2.000, in Zukunft aber auch 1:5.000 und 1:10.000 angeboten. Die Ausgabe in Vektorform kann mit Hilfe diverser Datenschnittstellen erfolgen. Die Ausgabe gemäß ÖNORM A2260/A2261 ist in Vorbereitung.

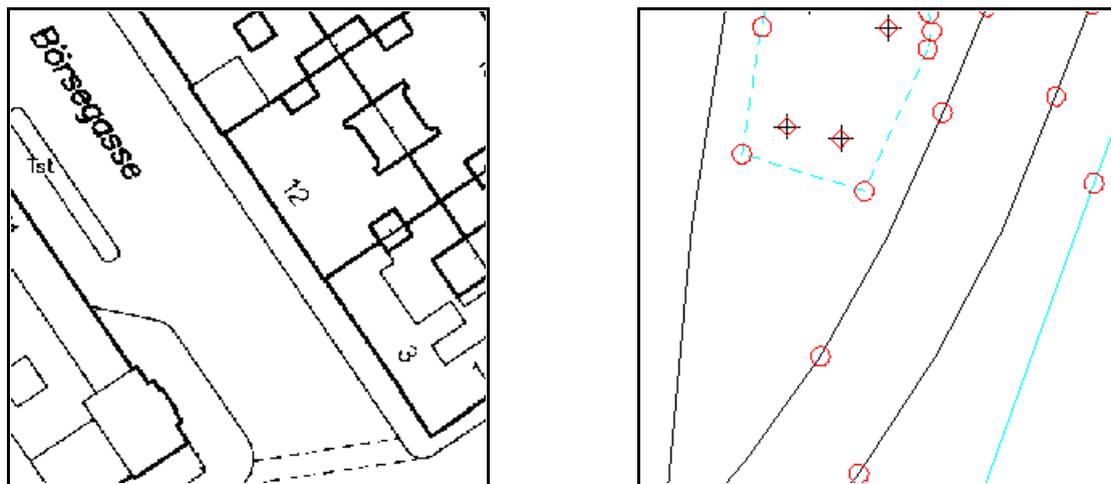


Abb.16: Ausschnitte aus der Mehrzweckstadtkarte Wien  
(Links: Rasterformat; Rechts: Vektorformat) (MA 41-Stadt Wien)

### 3.5 Digitale Stadtkartographie

Die Stadtkartographie, eine vergleichsweise junge Disziplin, hat sich heute ihren Platz im Rahmen der wissenschaftlichen Kartographie gesichert. Der Begriff "Stadtkartographie" wurde erstmals 1962 als Titel eines Aufsatzes von PAPE & STAHNKE verwendet. Die Aufgabe der Stadtkartographie sollte demnach sein, "grundsätzliche Fragen der städtischen Kartenherstellung" zu behandeln. Der Einfluss graphischer Datenverarbeitung, Geoinformationssystemen und der Internet-Technologien hat aber auch vor der Stadtkartographie nicht Halt gemacht und den Aspekt der "städtischen Kartenherstellung" deutlich erweitert. Dies führt zu neuen Sichtweisen, die in Kap. 3.2 allgemein für die gesamte Kartographie erläutert wurden, im Speziellen aber auch in der großmaßstäbigen

Kartographie des städtischen Raumes ihre Gültigkeit haben. Analoge Produkte wie Stadtkarte oder Stadtplan sind heute nur mehr als Teil des städtisch-kartographischen Publikationsspektrums zu sehen, obgleich insbesondere Stadtpläne wohl noch lange Zeit als ein Standardprodukt der klassischen Kartographie gelten werden.

### 3.5.1 Möglichkeiten

Gerade der urbane Ballungsraum mit seiner Dynamik und Vielschichtigkeit profitiert ungemein von den neuen Möglichkeiten der modernen Geodatenverarbeitung. Die Möglichkeiten, sachliche und örtliche Abhängigkeiten sowie dynamische Veränderungen aufzuzeigen, waren in der analogen Kartographie nur limitiert, sind aber im Rahmen digitaler Visualisierung in hohem Maße gegeben. Hiervon profitiert insbesondere die großmaßstäbige thematische Kartographie. Den vollen Umfang der Möglichkeiten erkennt man erst bei Betrachtung der Vielschichtigkeit der Stadtkartographie. GORKI & PAPE (1987) zeigen diese anhand der Klassifikationsmerkmale von OGRISSEK (1980) auf:

- ? **Kartenkategorie** (Zweckbestimmung):  
Orientierung, Verwaltung, Planung, Prognose, Forschung und Lehre
- ? **Kartenthemen** (Kartengegenstände):  
Neben den oben angeführten Kartenkategorien, die allesamt als Kartenthema der Stadtkartographie dienen können, ist auch die Topographie als Thema zu nennen
- ? **Kartentyp** (Strukturniveau):  
Analytisch, synthetisch, komplex und deren mögliche Kombinationen
- ? **Kartengruppe** (Maßstab):  
1:500 (Stadtgrundkarten) bis 1:100.000 (Stadt-Übersichtskarten)

Das breite Spektrum der technisch-kartographischen Produktionsmittel bietet dem Kartographen die Gelegenheit, die Vielschichtigkeit der Stadtkartographie voll auszuschöpfen.

### 3.5.2 Anforderungen

Die digitale, kleinräumige geographische Informationsverarbeitung im Rahmen der Stadtkartographie stellt besondere Anforderungen an die zugrundeliegenden Daten. Die Visualisierung im kleinmaßstäbigen Bereich macht eine absolute Lagegenauigkeit sowie einen hohen Detailreichtum notwendig. Daneben stellt die Aktualität der Daten eine wichtige Voraussetzung für die Verwendung von Produkten der Stadtkartographie dar.

Die heutigen Methoden der topographischen Datenerfassung machen eine Positionsgenauigkeit im Grundriss sowie in der Höhenlage möglich, die selbst im Planmaßstabbereich als ausreichend anzusehen sind. Die Kombination unterschiedlicher Aufnahmeverfahren - wie etwa im Falle der Wiener Mehrzweckstadtkarte (vgl. Kap. 3.4.2.2) - bieten neben der hohen Lagegenauigkeit einen ausreichenden Detailreichtum in allen Maßstabebereichen. Topographische Methoden digitaler Direktmessung machen ein Vercoden der aufgenommenen Geoobjekte bereits im Feld möglich und gewährleisten so eine durchgehend hohe Datenqualität, da keine analogen Zwischenschritte von der Messung bis zum Einspielen der Daten in ein Datenbanksystem notwendig sind. Aus den so erstellten Geomodellen lassen sich kartographische Modelle in einem breiten Maßstabbereich erstellen, ohne - von kartographischer Generalisierung abgesehen - die Lagegenauigkeit zu beeinträchtigen.

Die Notwendigkeit der Aktualität der Daten ist aus mehreren Gründen gegeben. Die Verwendung räumlicher sowie Sachdaten im Bereich der Verwaltung und Planung ist von einer hohen zeitlichen Aktualität abhängig, da nur diese den korrekten Ablauf der vorzunehmenden Arbeiten gewährleistet. Die kartographische Visualisierung kleinräumiger Sachverhalte führt - insbesondere im Internet - zu einem neuen Aspekt der Datenaktualität. Informationen im Internet unterliegen einem "scheinbar" schnelleren Verlust der Aktualität, als dies etwa im Bereich der Printmedien der Fall ist. Der Internet-Nutzer erwartet einen möglichst hohen Aktualitätsgrad der Darstellung, besonders wenn es sich um amtliche Produkte handelt.

Die Möglichkeit, kartographische Darstellungen mittels datenbankgestützter Visualisierungssysteme zu erstellen, macht die Anforderung nach hoher Aktualität der Daten erst möglich. Die zeitlich aktuellsten Daten werden bei Eintreffen einer Anfrage (engl.:

Request) aus dem Datenbanksystem ausgelesen und erst online zu einem Kartenbild umgewandelt [JÖRG 1999].

Die Verwendung von Datenbanksystemen zur Speicherung der Daten wie auch zur Visualisierung dieser führt somit zu einer neuen Charakteristik stadtkartographischer Produkte.

WILMERSDORF (1992) nennt Merkmale dieses neuen Produktprofils:

? **Hohe Datenqualität, Auflösung und Genauigkeit:**

Die Konservierung ursprünglicher Maßgenauigkeit der Einzeldaten und die automatische Prüfung von Gesetzmäßigkeiten führt zu einer verbesserten und anhaltenden Qualität der Positionsgenauigkeit.

? **Aktualität:**

Automatisierte Datenerfassung- und Fortführungsprozesse führen zu einer besseren Abbildung der Dynamik der Änderungen in der Natur im Geomodell.

? **Flexibilität in Inhalt und Darstellung:**

Methoden der graphischen Datenverarbeitung sowie der Internet-basierten Visualisierung bieten flexible Möglichkeiten der Präsentation der in Kap. 3.5.1 genannten Kartenthemen .

## 4 Internet

Wohl keine technische Innovation der letzten Jahrzehnte hat die Welt derart nachhaltig beeinflusst wie das Internet. Umberto ECO (1993) bezeichnet das Internet als jene Erfindung, die das Denken und Handeln der Menschheit am stärksten verändern wird, vergleichbar wohl nur mit der Entwicklung des Telefons, der Eisenbahn und dem Automobil. Obwohl das Internet, so wie wir es heute kennen, noch keine 25 Jahre alt ist, beherrscht es den Alltag wie kein anderes Kommunikationsmedium zuvor.

Eine kurze Chronologie soll die Dynamik zeigen, mit der sich das Internet seit seinen Anfängen in den frühen 60er Jahren entwickelt hat.

### 4.1 Chronologie

1962 begann die Advanced Research Project Agency (ARPA) des amerikanischen Verteidigungsministeriums mit den Planungen zum Aufbau eines kleinen Netzwerkes (ARPANET), um Wissenschaftler verschiedener Dienststellen in den USA einfacher miteinander kommunizieren lassen zu können. Eine der Forderungen an das zu entstehende Netzwerk war, dass bei der Zerstörung eines Teiles des Netzwerkes dieses trotzdem weiter zu funktionieren habe. Nach sieben Jahren Entwicklungszeit ging das ARPANET 1969 in Betrieb. Die ersten vier angeschlossenen Computer (man spricht in diesem Zusammenhang von *Hosts*) befanden sich im Stanford Research Institute, der University of California (Los Angeles), University of California (Santa Barbara), und an der University of Utah.

1973 besitzt das ARPANET etwa 60 Hosts, darunter das University College in London, und das Royal Radar Establishment in Norwegen. Der beliebteste Verwendungszweck dieses frühen Netzes war das Versenden elektronischer Post, heute unter dem Begriff Email allbekannt.

Die wahre Geburtsstunde des Internet, wie man es heute kennt, ist 1982 anzusetzen. Das Transmission Control Protocol/Internet Protocol (TCP/IP) wurde von Bob Kahn und Vint Cerf entwickelt. Diese Sammlung an Protokollen regelt den Transport und die Zustellung der Datenpakete von einem Computer zu einem anderen. Das Problem des

unstandardisierten Zusammenhangs der Rechner im ARPANET war somit gelöst, der einheitliche Standard TCP/IP macht es jedermann möglich, an diesem Netzwerk zu partizipieren. Der Terminus Internet setzt sich in den folgenden Jahren allgemein durch, die Anzahl der Hosts steigt bis 1989 auf 100.000.

1990 wird der Betrieb des ARPANET offiziell beendet. Befürchtungen, das damit das gesamte Internet zum Erliegen kommt, bestätigen sich nicht. Das Verbot der kommerziellen Nutzung des Internet wird 1991 aufgehoben. Im selben Jahr entwickelt Tim Berners-Lee am Europäischen Forschungszentrum für Nuklearphysik (CERN) in Genf die Idee, Texte, Bilder und andere Medien mittels einer Computersprache über das Internet zu versenden. Das grundlegende Prinzip nannte er Hypertext, die Skriptsprache HTML (Hypertext Markup Language). HTML ist bis heute die bedeutendste Sprache des graphischen Teils des Internet, dem World Wide Web (WWW).

Die erste graphische Software-Applikation, mit der man das WWW einfach und Bequem nutzen konnte - der Browser - wurde 1993 von Marc Andreessen entwickelt. Das Internet war im Begriff, das weltweit wichtigste Kommunikationsmedium zu werden. Die jährliche Wachstumsrate der im Internet übermittelten Daten betrug 1993 341,6%.

Die Firma SUN Microsystems entwickelt 1995 die Programmiersprache *Java*, die es dem Benutzer erstmals erlaubt, Inhalte von Internet-Seiten nach seinen Vorstellungen interaktiv zu beeinflussen.

Über 100.000.000 Hosts in mehr als 170 Nationen bilden das Internet im Jahre 2001.

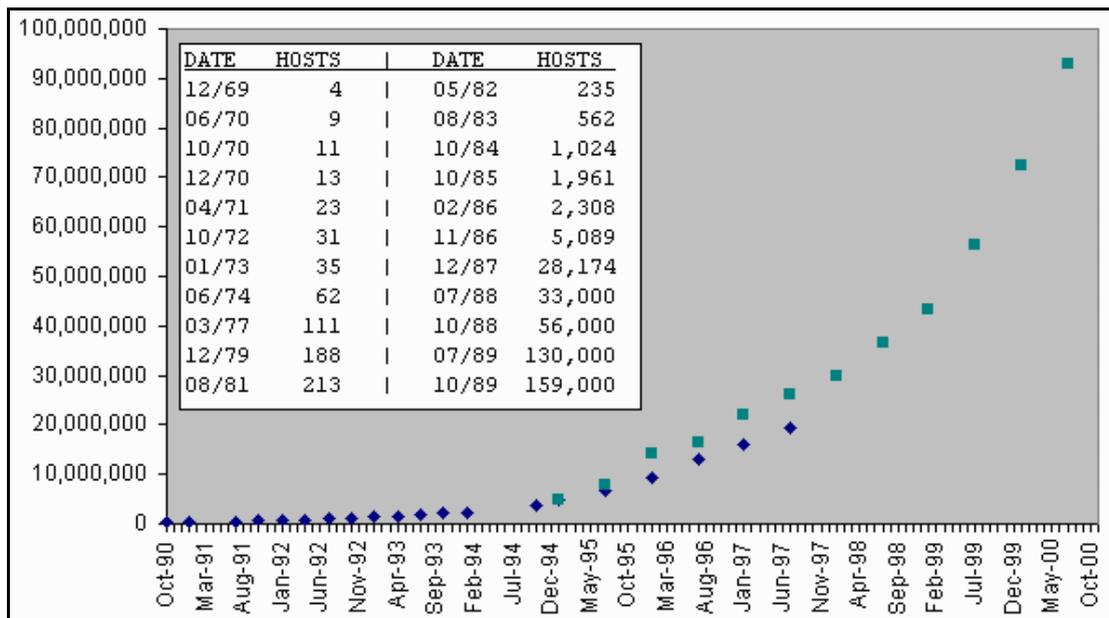


Abb.17: Entwicklung der Anzahl der Internet-Hosts [Hobbes 2001]

## 4.2 Bedeutung des Internet

Worauf ist die heutige Bedeutung des Internet zurückzuführen?

### ? **Zuverlässigkeit:**

Das Internet stellt ein dezentrales Netzwerk dar. In diesem sind alle beteiligten Computer prinzipiell gleichberechtigt. Dies hat zur Folge, dass bei Defekt eines Rechners ein anderer dessen Aufgaben übernehmen kann. Ein totaler Ausfall des Internet ist somit praktisch auszuschließen.

### ? **Verbreitung:**

Das Internet kann heute als ubiquitär angesehen werden. Statistiken sprechen von 250 bis 300 Millionen Internet-Benutzern weltweit. Besonders die Entwicklungen rund um das World Wide Web, der graphischen Benutzerschnittstelle des Internet, trugen dazu bei. Daneben erfreuen sich weitere Internet-Dienste wie Email oder FTP (File Transfer Protocol) großer Beliebtheit.

### ? **Kosten:**

Die grundlegenden Normen des Internet, wie TCP/IP oder HTML, sind freie, offene Standards und stellen keine proprietären Produkte dar. Der Zugang zum Internet ist heute noch häufig mit der Entrichtung von Gebühren verbunden. Die

Tendenz der letzten Jahre weist aber deutlich hin zu einem kostenfreien Nutzen des Internet in der Zukunft.

Bei aller Euphorie soll nicht verschwiegen werden, dass weite Teile des Internet von einer starken Kommerzialisierung betroffen sind. Bestrebungen, ein parallel laufendes Netzwerk aufzubauen, das die ursprünglichen Gedanken des ARPANET - den Austausch wissenschaftlichen Gedankenguts - wieder stärker forcieren soll, sind im Stadium der Realisierung. Die oben genannte Zahl der Internet-Benutzer ist weit davon entfernt, eine Gleichverteilung über alle Kontinente darzustellen. Es zeigt sich vielmehr eine deutliche Konzentration in den USA, Westeuropa und in hoch technisierten Staaten wie Japan und Australien [KRIZ 1999].

### **4.3 Generierungstechniken**

Im folgenden Abschnitt soll erläutert werden, welche Möglichkeiten bestehen, Internetinhalte zu erstellen. Dabei soll besonders auf die Möglichkeiten eingegangen werden, Inhalte zu erzeugen, die vom Benutzer interaktiv beeinflusst werden können.

#### **4.3.1 Statische Internetinhalte**

Die Darstellung kartographischer Inhalte im Internet weist besondere Eigenschaften auf. So soll einerseits eine gewisse Zeitersparnis gegenüber der Verwendung analoger Produkte gegeben sein, andererseits dem Nutzer die Möglichkeit geboten werden, die Darstellung interaktiv zu beeinflussen. Geodatenbestände weisen häufig beträchtliche Speichermengen auf. Die Bearbeitung der Daten ist zeit- und rechenintensiv, die Ergebnisse stellen oftmals große Datenmengen dar. Die derzeit noch geringen Übertragungskapazitäten im Internet zwingen einen aber, möglichst geringe Datenmengen zu übersenden, um die Übertragungszeiten in erträglichem Rahmen zu halten. Es muss somit ein Kompromiss zwischen bearbeitbarer Datenmenge und Übertragungskapazität gefunden werden. Unmittelbar mit dieser Tatsache verknüpft ist die Entscheidung, ob die Darstellung als Vektor- oder Rastergraphik übermittelt werden soll (vgl. Kap. 2.4.1).

HTML stellt heute die Standardsprache des WWW dar und liegt derzeit in der Version 4.0 vor. HTML gehört zur Familie der Seitenbeschreibungssprachen, sie besteht aus einer Folge von ASCII-Zeichen, die vom Browser als Formatierungsbefehle für die Seitengestaltung übersetzt werden. Eine mittels HTML erstellte Internetseite wird auf einem speziellen Rechner - dem Server - abgelegt. Dieser stellt den übrigen Rechnern des Netzwerkes - den Clients- bestimmte Dienste zur Verfügung. Dieses Prinzip liegt dem gesamten Internet zugrunde und wird als Client/Server-Modell bezeichnet. Soll die Hauptfunktion, die der Server erfüllen soll, das Versenden von HTML-Seiten sein, so bedient man sich einer Software, die als Webserver bezeichnet wird. Im vorliegenden Fall würde der Server beim Anfordern einer Internetseite - etwa in der allgemeinen Form *http://servername/seite.html* - diese mittels des Webserver an den Client übermitteln. Reine HTML-Seiten sind statisch. Der Benutzer kann sich einzelne Seiten ansehen, diese aber nicht interaktiv ändern. Reines HTML kann also auch nicht zur Manipulation von Datenbanken, und damit verbundener Visualisierung der Daten, benutzt werden, da das Aussehen einer Seite starr festgelegt ist [MOSSGRABER, 1997].

#### **4.3.2 Interaktive Internetinhalte**

Um diesen Umstand zu beheben, liegen mittlerweile mehrere technische Lösungen vor, in die starre Struktur des HTML-Codes Platzhalter zu integrieren, die je nach vorliegender Anfrage des Clients mit variablen Inhalten aus einer Datenquelle gefüllt werden können. Diese Dateien sind nunmehr in ihrer ursprünglichen Form nicht mehr von HTML-fähigen Browsern lesbar, sie stellen vielmehr ein Art Vorlage dar, die je nach Art der Anfrage in die gewünschte HTML-Datei konvertiert wird.

Diese Technik ermöglicht den aktiven Zugriff auf Datenbanksysteme und deren Bearbeitung mittels SQL. Abhängig davon, ob die Bearbeitung der Daten am Server oder am Client vor sich geht, spricht man von server- oder clientseitigen Lösungen.

##### **4.3.2.1 Serverseitige Technologien**

Bei serverseitigen Implementierungen wird die darzustellende Datei am Server erstellt. Als Ergebnis wird somit reiner HTML-Code zum Client gesendet. Dies bedingt eine geringe zu übertragende Speichermenge und damit verbunden einen schnellen Bildaufbau. Dieses Tatsache ist - wie oben erwähnt - besonders bei der Verarbeitung großer Geometriedatensätze zu beachten. Es wird nur die Anweisung zum Abarbeiten bestimmter Funktionen übersendet, nicht jedoch die Daten selber. Serverseitige Technologien erlauben es somit, Geometrie- sowie Sachdaten auf im Netzwerk zentral gelegenen Servern zu lagern. Beim Eintreffen einer Anfrage werden die notwendigen Daten lokal verarbeitet, das Resultat als verhältnismäßig kleine Datei zurückgesendet.

Aus der Vielzahl der - teils proprietären - Lösungen heben sich besonders folgende hervor [SAMLAND, HINTERMEIER & DÜRR 2000]:

? **CGI (Common Gateway Interface)**

CGI stellt keine Programmiersprache, sondern lediglich eine Schnittstelle am Server dar. Clients wird es damit ermöglicht, ausführbare Programme am Server zu starten. Die Ausgabe des Programmes wird in den HTML-Code integriert und an den Client zurückgesendet. Der Einsatz von CGI-Programmen liegt häufig in der Auswertung von Formularen oder Datenbankabfragen. CGI-Skripts sind in der Regel in der Skriptsprache "Perl" geschrieben, können aber prinzipiell in jeder beliebigen Programmier- oder Skriptsprache verfasst sein. Exemplarisch seien hier FORTRAN, BASIC und "Shellscripts" zu nennen.

? **Java-Servlets**

In der objektorientierten Programmiersprache "Java" geschrieben, stellen Java-Servlets kleine, ausführbare Programme am Server dar. Der Vorteil gegenüber CGI besteht darin, dass, einmal geladen, Servlets im Speicher des Rechners verbleiben und somit bei neuerlicher Verwendung nicht mehr neu geladen werden müssen. Java-Servlets stellen im Vergleich zu anderen serverseitigen Lösungen eine sehr stabile und schnelle Technologie dar.

? **PHP (PHP Hypertext Preprocessor)**

PHP stellt eine relativ neue Möglichkeit der Erzeugung interaktiver Internetseiten dar. PHP ist dabei weniger als Programmiersprache zu sehen als vielmehr als Webservererweiterung, die HTML-Code mit Hilfe einer Skriptsprache generiert. Ein wesentlicher Unterschied zu CGI und Java-Servlets besteht darin, das PHP-Code direkt in HTML integriert wird.

PHP, oft mit Perl verglichen, bietet diesem gegenüber einige Vorteile, die gerade in Hinblick auf die datenbankgestützte kartographische Visualisierung im Internet zu nennen sind. Die Entwicklung von Perl datiert aus dem Jahr 1987, der Fortschritt des Internet bis zum heutigen Stand war damals noch nicht absehbar. Perl lässt deswegen besondere, Internet relevante Funktionen vermissen, die in PHP implementiert sind. Während der Entwicklung von PHP wurde die Bedeutung der Nutzung von Datenbanksystemen im Internet erkannt. Funktionen, die die Bearbeitung von Datenbanken ermöglichen, sind aus diesem Grund ein Hauptbestandteil von PHP.

Internetfähigkeit und Datenbankoptimierung lassen somit darauf schließen, dass sich PHP besonders für den kartographieorientierten Einsatz im Internet eignet. Kap 7 zeigt, wie PHP im praktischen Einsatz zur datenbankgestützten kartographischen Visualisierung im Internet eingesetzt werden kann.

#### **4.2.2.2 Clientseitige Technologien**

Werden Daten nicht auf dem Server, sondern auf dem Client bearbeitet, spricht man von clientseitigen Technologien. Zu beachten ist hierbei, dass die Datenquellen somit mit der aufgerufenen Seite an den Client gesendet werden müssen. Um den dabei entstehenden großen Datenmengen entgegenzuwirken, arbeiten clientseitige Visualisierungssysteme häufig mit Vektordaten, die gegenüber Rasterdaten einen wesentlich geringeren Speicherbedarf haben (vgl. Kap. 2.4.1).

##### **? JavaScript:**

Ähnlich wie PHP wird auch JavaScript direkt in den HTML-Code eingebunden, im Gegensatz zu diesem aber auf dem Client interpretiert. Somit muss die Seite bei einer neuerlichen Darstellung nicht wieder mit dem Server kommunizieren, sondern die interaktive Generierung läuft auf der Benutzerseite ab.

##### **? Plug-Ins:**

Es handelt sich hierbei um Programme, die Browser um zusätzliche Funktionen erweitern. Plug-Ins müssen vom Benutzer einmalig installiert werden, ermöglichen es dann aber, die bereitgestellten Funktionalitäten dauerhaft zu nutzen. Im Bereich der kartographischen Visualisierung im Internet wird vor allem mit Plug-

Ins für Vektorgraphiken gearbeitet. Zu nennen wäre in diesem Zusammenhang SVG (Streaming Vector Graphics) und Flash [WINTER 2000].

Clientseitige Lösungen finden verhältnismäßig wenig Einsatz, da sich bei großen Datenmengen die Tatsache, dass die gesamten Daten und nicht nur das Ergebnis der Anfrage über das Netzwerk verschickt werden müssen als unvorteilhaft herausstellt.

## **5 Datenbankgestützte kartographische Visualisierung im Internet**

### **5.1 Kartographische Ausdrucksformen im Internet**

Die Darstellung graphischer Inhalte im Internet ist so alt wie das World Wide Web. Tim Berners-Lee integrierte in seiner Konzeption des Hypertext die Möglichkeit der Einbindung graphischer Objekte. Seit der HTML Version 1.0 ist es möglich, Rastergraphiken in Internetseiten zu platzieren. Diese Tatsache wurde von Kartographen früh erkannt, kartographische Darstellungen fanden rasch den Weg in das WWW. Sowohl die Zahl, als auch die Vielfalt der kartographischen Produkte ist bis heute stetig angestiegen, was nicht zuletzt auch auf die wachsende Kommerzialisierung des Internet zurückzuführen ist.

KRAAK (2000) teilt sämtliche kartographischen Ausdrucksformen im Internet in vier Gruppen ein:

- |                          |                           |
|--------------------------|---------------------------|
| ? Statisch - "view only" | ? Dynamisch - "view only" |
| ? Statisch - interaktiv  | ? Dynamisch - interaktiv  |

Die Gruppe der statischen-"view only" Abbildungen stellt den klassischen Typ der Karte im Internet dar. In der Regel handelt es sich um gescannte Originalkarten, die direkt als GIF oder JPEG-Datei in HTML eingebettet sind. Diese Form der kartographischen Präsentation erlaubt - ähnlich einer analogen Karte - keine Interaktion und folgt damit dem in Kap. 3.2.1 besprochenen linearen Modell der kartographischen Kommunikation.

Statisch-interaktive Produkte erweitern die Möglichkeiten von "view-only" Karten, indem sie sich als "clickable-maps" (auch "image-maps") präsentieren. Im Bereich der dargestellten Fläche werden sensitive "hot-spots" definiert. Dies sind Regionen, die beim Anklicken der Maus eine interaktive Aktion auslösen. In der Regel handelt es sich dabei um einen Link, einen Verweis auf ein neues HTML-Dokument. "Clickable-maps" können über das <MAP>-Tag direkt in HTML integriert werden und laufen clientseitig ab. Daneben ist es aber im Rahmen von Mapserver-Applikationen möglich, diese serverseitig zu bearbeiten, was einen enormen Zuwachs an Interaktionsmöglichkeiten zulässt (vgl. Kap. 6.4).

Der Bereich der dynamischen kartographischen Darstellungen ist eng mit dem Einsatz von Animationen verbunden. Dynamische "view only" Produkte sind als eine Abfolge von statischen "view only" Darstellungen zu sehen. Man spricht in diesem Zusammenhang häufig von "Animated-GIF". Bezüglich der Möglichkeiten, aktiv in den Ablauf der Animation einzugreifen, gelten sinngemäß die Ausführungen zu statischen "view only" Darstellungen.

Neueste Entwicklungen, vor allem im Bereich von "Java" und "Virtual Reality", liefern kartographische Ausdrucksformen, die als dynamisch-interaktiv zu sehen sind. Der Nutzer ist in der Lage, die Richtung - im Sinne des Ablaufs - der Animation vorzugeben. Im einfachsten Fall könnte dies bedeuten, die Geschwindigkeit oder Ablaufrichtung eines "Animated-GIF" interaktiv zu bestimmen. Hochkomplexe "Virtual Reality"-Systeme versetzen den Nutzer mittels Datendisplays und -handschuh mitten in dreidimensionale Welten [RIEDL 2000].

Es stellt sich die Frage, inwieweit der Einsatz von Datenbanksystemen Vorteile für die vier oben genannten Gruppen mit sich bringt. Hierzu muss unterschieden werden, ob das Datenbanksystem lediglich zur Verwaltung statischer-"view only" Produkte verwendet wird, oder ob die kartographische Ausdrucksform selber durch ein datenbankgestütztes Visualisierungssystem erstellt werden soll.

Im Internet finden sich zahlreiche "Map Libraries", die eine Vielzahl an gescannten Karten und Plänen anbieten. Diese Angebote arbeiten in der Regel datenbankgestützt. Eine eintreffende Anfrage nach einer bestimmten Karte wird von einem Datenbanksystem bearbeitet, und das Suchergebnis an den Webserver übermittelt. Dieser stellt die Karte im Browser des Nutzers als Rastergraphik dar. Festzuhalten ist hierbei, dass das Datenbanksystem lediglich zum Zweck der Verwaltung der statischen Karten dient. Dieser Systemablauf ist kaum als kartographisch zu bezeichnen, da weder Geo- noch Sachdaten zur Erstellung der Karte bearbeitet werden. Im Gegensatz dazu existieren Systeme, die nicht mit fertig erstellten Karten arbeiten, sondern diese erst aus den Inhalten von Datenbanken generieren. Sowohl Geo- als auch Sachdaten werden dabei von datenbankgestützten kartographischen Visualisierungssystemen direkt bearbeitet und liefern als Ergebnis eine interaktiv erstellte Karte.

## 5.2 Datenbankgestützte kartographische Visualisierungssysteme

Karten, die mittels datenbankgestützter kartographischer Visualisierungssysteme erstellt werden, sind als statisch-interaktiv zu sehen, da die Möglichkeit besteht, interaktiv in den Gestaltungsprozess einzugreifen. Die Möglichkeit der Interaktion ist aber nur ein Kriterium, das von modernen kartographischen Visualisierungssystemen verlangt wird. Daneben gibt es noch zahlreiche Anforderungen, die in der Folge angeführt werden sollen.

### 5.2.1 Anforderungen

Die Anforderungen an datenbankgestützte kartographische Visualisierungssysteme sind vielfältig und hängen im Einzelfall von der zu realisierenden Applikation und den gewünschten Funktionen in dieser ab.

Folgende Ansprüche sollte man allgemein an ein System stellen, das als Internet-basiertes datenbankgestütztes kartographisches Visualisierungssystem dienen soll:

? **Kartographische Darstellung:**

Das System sollte in der Lage sein, Geodaten mit zugehöriger Attributierung zu visualisieren. Dies bedingt, dass Zugriffsmöglichkeiten auf das Geo- sowie thematische Modell bestehen, um das kartographische Modell zu erstellen. Reine Visualisierungssysteme dienen ausschließlich der Visualisierung von Geodaten und nicht deren Bearbeitung, da hierfür eine Verbindung zu einem Geoinformationssystem bestehen müsste.

? **Bearbeitung von Vektor- und Rasterdaten:**

Obwohl Geomodelle in der Regel in Vektorformat vorliegen, sollte das System zusätzlich mit Rastergraphiken arbeiten können. Eine mögliche Anwendung wäre in diesem Zusammenhang die Darstellung eines Orthofotos (Raster) mit zusätzlicher Information aus dem Geomodell (Vektor).

? **Bearbeitung von Geometrie- und Sachdaten:**

Die Bearbeitung von Sachdaten ist mit jedem gängigen Datenbankmanagementsystem möglich. Im Idealfall ist ein datenbankgestütztes kartographisches Visualisierungssystem in der Lage, sowohl Sach- als auch

Geometriedaten in einer Datenbank zu bearbeiten. Die Bearbeitung von Geometriedaten ist allerdings nur mit speziellen Datenbankmanagementsystemen möglich. Diese sind in der Lage, Funktionen und Methoden zur Manipulation räumlicher Objekte durchzuführen.

? **Interaktionsmöglichkeiten:**

Interaktion erlaubt es dem Nutzer, die Darstellung nach seinen eigenen Wünschen zu gestalten. Folgende Interaktionsmöglichkeiten stellen eine Auswahl denkbarer Funktionen dar:

**Orientierung in der Karte:**

- ? Pan: Änderung des Bildausschnittes in X-Y-Richtung
- ? Zoom: Änderung des Bildausschnittes in Z-Richtung (entspricht einer Änderung des Maßstabs)

**Kartengestaltung:**

- ? Ebenenwahl: Auswahl der darzustellenden Ebenen des Geomodells
- ? Signaturenwahl: Zuweisung frei wählbarer Signaturen zu Geoobjekt

**Datenmanipulation:**

- ? Klassifizierung: Erstellung eigener Klassen nach frei wählbaren Klassengrenzen

Zu bedenken sei aber, dass der Umfang der Interaktionsmöglichkeiten sehr stark von den zur Verfügung stehenden Daten abhängt. So ist etwa die Entscheidung, den Maßstab frei wählbar zu machen, kaum sinnvoll, da eine adäquate kartographische Darstellung in allen Maßstabsbereichen kaum gewährleistet werden kann. Es empfiehlt sich eher, den Maßstab innerhalb eines begrenzten Bereiches durch den Nutzer interaktiv festlegen zu lassen.

? **Manipulation der Datenbank:**

Neben der Möglichkeit, auf den bestehenden Datenbestand zugreifen und diesen visualisieren zu können, sollte das System die Möglichkeit bieten, neue Daten in die Datenbank einzuspielen sowie bestehende in kontrolliertem Ausmaß bearbeiten zu können. Da der Nutzer nur selten Kenntnis von der Daten- und Tabellenstruktur der gewünschten Datenbank hat, muss der Import neuer Daten über eine graphische Eingabemaske erfolgen, die im Hintergrund die notwendigen SQL-Befehle selbstständig ausführt.

? **Plattformunabhängigkeit:**

Der Anspruch nach Plattformunabhängigkeit ist aus zwei Sichtweisen zu sehen. Zum Einen sollte die Architektur des Visualisierungssystems Heterogenität in Bezug auf das zu verwendende Betriebssystem aufweisen. Damit ist ein variabler Einsatz auf jeglicher technischen Systemarchitektur denkbar. Daneben sollte das fertige Produkt plattformunabhängig zu betrachten sein. Dies bedeutet, dass jeder handelsübliche Browser in der Lage sein sollte, das Produkt fehlerfrei darzustellen. Die gesamte Bildschirmdarstellung - kartographische Ausdrucksform und zusätzliche Information - sollte diesen Anspruch erfüllen. Proprietäre Plug-Ins, die nicht von allen Browsern unterstützt werden, sollten aus diesem Grund nicht zum Einsatz kommen (vgl. Kap. 4.2.2.2).

? **Mehrnutzerbetrieb:**

Das System sollte in der Lage sein, Anfragen mehrerer Nutzer gleichzeitig bearbeiten zu können. Ein entsprechend hohes Niveau an Hardware muss dafür vorausgesetzt werden. Sieht das System vor, dass Datenbanken direkt bearbeitet werden können, muss gewährleistet sein, dass die Konsistenz der Daten bei zeitgleichem Zugriff mehrerer Nutzer erhalten bleibt (vgl. Kap. 2.5.3).

Das Ausmaß, mit dem oben genannte Anforderungen erfüllt werden, ist von der Komplexität der Applikation abhängig. Diese Komplexität spiegelt sich in der Architektur wieder, die das datenbankgestützte Visualisierungssystem aufweist.

## 5.2.2 Architektur

In Kap. 4.3 wurde eine Unterteilung nach dem Kriterium durchgeführt, wo die Generierung der Darstellung stattfindet. Es wurde zwischen client- und serverseitigen Technologien unterschieden. Diese Komponente, die die eigentliche Produktion der Abbildung durchführt, ist aber nur ein Bestandteil eines datenbankgestützten Visualisierungssystems. Daneben stellen das Datenbanksystem und die Benutzeroberfläche weitere Teile eines solchen Systems dar. Das in Kap. 4.3.1 erwähnte Client/Server-Modell des Internet stellt formal ein logisches Modell dar, welches dafür sorgt, dass die Teile des Systems auf Client und Server verteilt werden.

Die Anordnung der Systemkomponenten zueinander und deren Verteilung auf Client und/oder Server wird in Form von Schicht-Modellen dargestellt [HAACK 2000].

- ? 1-Schicht-Modell (Monolithische Applikation)
- ? 2-Schicht-Modell
- ? 3-Schicht-Modell

#### **5.2.2.1 1-Schicht Modell (Monolithische Applikationen)**

Die simpelste Form einer Client/Server-Architektur ist das 1-Schicht-Modell. In dieser Konfiguration stellt ein monolithisches (einheitliches) Serversystem Dienstleistungen für unintelligente "Terminal-Clients" zur Verfügung, die direkt mit dem Server verbunden sind. Alle Arbeitsschritte bis zur Präsentation der Abbildung werden am Server durchgeführt, der Client dient lediglich als graphische Ausgabeplattform der Information.

#### **5.2.2.2 2-Schicht-Modell**

Das monolithische System wird in diesem Modell in zwei Teile getrennt, wobei das Datenbanksystem eine Schicht, die Generierungskomponente samt Benutzeroberfläche eine weitere darstellen. Dieses aus den späten achtziger Jahren stammende Modell findet heute bei simplen CGI Scripten Anwendung und eignet sich höchstens für kleinere Applikationen.

#### **5.2.2.3 3-Schicht-Modell**

Das 3-Schicht-Modell führt eine dreifache Trennung des Systems durch:

- ? Datenbankschicht
- ? Applikationsschicht
- ? Präsentationsschicht

In einem typischen 3-Schicht-Modell befindet sich auf dem Client ein Programm, das sich um die Darstellung der Benutzeroberfläche kümmert. Dieses enthält von den Interaktionsmöglichkeiten abhängige Eingabeformulare und interaktive Elemente. Eingaben werden an die Applikationsschicht, die sich am Server befindet, gesendet, von dieser bearbeitet und an die dritte Schicht des Modells, das Datenbanksystem, übermittelt. Dieses muss sich nicht zwingend auf dem selben Server befinden wie die Applikationsschicht. Nach Bearbeitung der Anfrage durch das Datenbanksystem werden die gewünschten Daten an die Applikationsschicht zurückgesendet, diese erstellt die Darstellung und übermittelt diese schließlich an die Benutzeroberfläche am Client.

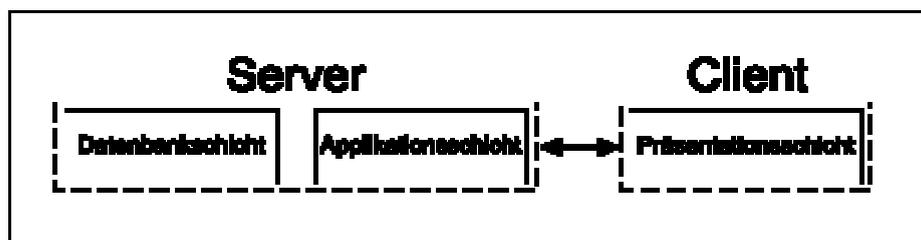


Abb. 18: Aufbau des 3-Schicht-Client/Server-Modells

Das 3-Schicht-Modell stellt den heute gängigen Stand der Technik dar und eignet sich damit auch in Hinblick auf die zu erfüllenden Anforderungen an datenbankgestützte kartographische Visualisierungssysteme (vgl. Kap. 5.2.1).

Als besondere Merkmale des 3-Schicht-Modells seien erwähnt:

? **Flexibilität:**

Die Komponenten des Systems sind klar voneinander getrennt. Änderungen an einem der Bestandteile können somit isoliert vom Rest des Systems vorgenommen werden. Im Idealfall kann die Entwicklung der einzelnen Systemkomponenten somit jeweils von Experten dieses Fachgebietes durchgeführt werden.

? **Geschwindigkeit:**

Da sich die einzelnen Schichten auf verschiedenen Rechnern befinden können, ist eine Skalierbarkeit von Servern gewährleistet. Dies sieht eine Struktur vor, in der ein Server lediglich die Aufgaben der Datenbankschicht übernimmt, ein anderer die der Applikationsschicht.

### ? **Thin-Clients:**

Da am Client lediglich die Präsentationsschicht - die Benutzeroberfläche - vorhanden ist, bestehen keine hohen Hardwareanforderungen an diese Rechner. Diese Tatsache wird zukünftig bei der graphischen Datenausgabe auf Kleingeräten (Mobil-Telefon, Personal Digital Assistant) an Bedeutung gewinnen.

Das 3-Schicht-Modell dient als Basis der meisten gängigen datenbankgestützten kartographischen Visualisierungssysteme im Internet. Die Konsequenz, mit der das Modell umgesetzt wird, ist abhängig von der Komplexität der Applikation. Diese kann von einfachen Systemen, die lediglich eine interaktionslose Darstellung ermöglichen, bis zu voll funktionsfähigen Online-Geoinformationssystemen reichen.

## **5.2.3 Gliederung**

In Abhängigkeit davon, in welchem Umfang die genannten Anforderungen technisch umgesetzt werden, unterscheidet man verschiedene Systeme, die sich sowohl nach technischen Gesichtspunkten, als auch nach dem bestehenden Anforderungsprofil unterscheiden lassen.

### **5.2.3.1 Einteilung nach technischen Gesichtspunkten**

Ein System der Internet-basierten Geodatenvisualisierung kann unterschiedliche Komplexitätsstufen aufweisen. Die Vorstellungen, wie ein kartographisches Visualisierungssystem im Internet genutzt werden kann, sind vielfältig. Die Verwendung von Datenbanksystemen zur Verwaltung der Geo- und/oder Sachdaten ist ein gemeinsames Merkmal aller Systeme.

FITZKE (1997) nennt folgende Möglichkeiten :

? **Mapserver:**

Mapserver übermitteln Karten zur Online-Visualisierung. Während statische Mapserver (auch Map-Libraries) lediglich Zugriff auf vorgefertigte Karten zulassen, können interaktive Mapserver Karten dynamisch zur Laufzeit erstellen und versenden.

? **Online-Auskunftssysteme:**

Neben der Visualisierung von Karten besteht bei diesen Systemen die Möglichkeit, thematische und einfache raumbezogene Abfragen zu bearbeiten.

? **Online-GIS:**

Ein voll funktionsfähiges Geoinformationssystem wird auf einem Server betrieben. Der Client hat vollen Zugang zu allen - auf den entsprechenden Server - bereitgestellten Daten.

? **GIS-Funktions-Server:**

Hier werden nur die Funktionalitäten eines GIS-Servers bereitgestellt. Ein Client kann damit zielgerichtet eigene Geodaten analysieren. Dabei muss er entweder eigene Daten auf den Server transferieren oder die entsprechenden Funktionen als Software-Komponente auf den Client laden.

Neben diesen Systemen findet man in der Fachliteratur häufig den Typus des Geodaten-Servers. Diese bieten ausschließlich Möglichkeiten zur Suche nach bestimmten Datensätzen und deren Übermittlung. Der Client muss diese Datensätze lokal und offline mit einem passenden Geoinformationssystem weiterverarbeiten. Ein reiner Geodaten-Server ist somit kein datenbankgestütztes kartographisches Visualisierungssystem im strengen Sinne, da keine Möglichkeit der kartographischen Visualisierung besteht.

### 5.2.3.2 Einteilung nach Anforderungen

Eine weitere Form der Einteilung wurde durch das OpenGIS®Consortium (1999) durchgeführt, das eine Differenzierung entsprechend der unterschiedlichen Anforderungen durchführt.

? **Picture Case:**

Rasterbilder der Karten werden durch eine serverseitige Applikation erstellt und in Form standardisierter Bildformate (vor allem GIF und JPEG) an den Client gesendet.

? **Graphic Element Case:**

Der Client übernimmt zusätzlich zur Darstellung der Geodaten Aufgaben des "Renderings" und bietet damit einfache Funktionen - Vergrößern, Verkleinern, Navigation - zur Bearbeitung der Karte an.

? **Data Case:**

Alle Funktionen bis auf die Datenabfrage werden lokal auf dem Client bearbeitet. Es ist somit möglich, sich lediglich die Datensätze vom Server zu kopieren und dann offline zu arbeiten. Diesem Vorteil steht allerdings der Nachteil gegenüber, dass bei dieser Form der Geodatenverarbeitung die größte Arbeitslast auf Seiten des Clients liegt.

Die Aufzählung der unterschiedlichen Möglichkeiten der Nutzung datenbankgestützter kartographischer Visualisierungssysteme lässt den Umfang der Einsatzmöglichkeiten derartiger Systeme erahnen. Es handelt sich dabei keineswegs um eine qualitative Abstufung, da das einzusetzende System in hohem Maße von den gestellten Anforderungen an die Applikation abhängig ist. Es ergibt sich somit die Frage der Intention, des Einsatzzweckes des Systems. Das Benutzen eines Online-GIS oder GIS-Funktions-Servers ist nur mit vorhandenen GIS-Kenntnissen möglich und damit einer limitierten Nutzergruppe vorbehalten. Der Einsatz eines solchen Systems wird daher eher im Bereich von firmeninternen Intranet-Lösungen zu finden sein. So ist es denkbar, fachkundigen Mitarbeitern eines Betriebes den Zugang zu einem Online-GIS zu ermöglichen, um anfallende Arbeiten im Aussendienst zu erledigen.

Eine andere Intention haben Mapserver und Online-Auskunftssysteme. Sie sind, auf einer niedrigeren Komplexitätsebene als Online-GIS, auch von Geoinformations- und Kartographie-Ungeschulten zu bedienen. Mapserver sind heute in vielen, vor allem öffentlichen Bereichen, zu finden und stellen im Internet keine Seltenheit mehr dar. Der gegenwärtige Trend sieht einen vermehrten Einsatz von (3-Schicht-Modell) Mapserver-Systemen im Internet, die eine interaktive, dynamisch zur Laufzeit erfolgende Erstellung von Karten ermöglichen. Im folgenden soll dieses System der datenbankgestützten kartographischen Visualisierung im Internet eingehend erläutert werden.

## **6 Datenbankgestützte Mapserver-Systeme**

Internet-basierte datenbankgestützte Mapserver-Systeme dienen primär dem Zweck der Darstellung von Karten. Diese Karten liegen nicht als digitale Versionen analoger Produkte vor, sondern werden dynamisch zur Laufzeit generiert. In weiterer Folge bieten diese Systeme einfache Interaktions- und Manipulationsmöglichkeiten. In der Folge sollen die Komponenten eines datenbankgestützten Mapserver-Systems erläutert werden, wie auch die Funktionsweise des Systems als Ganzes.

### **6.3 Komponenten**

Mapserver-Systeme bestehen in der Regel aus drei Komponenten, dem Datenbanksystem, dem Mapserver und der graphischen Benutzeroberfläche. Um ein voll funktionsfähiges Internet-basiertes System zu erhalten, sind noch zwei weitere Komponenten notwendig, die in aller Strenge nicht zum eigentlichen Mapserver-System zu zählen sind. Zur Verbindung zwischen Datenbanksystem und Mapserver ist eine Datenbankschnittstelle notwendig. Zudem wird ein Webserver benötigt, eine Software, die die Darstellung von HTML-Seiten im Internet erst ermöglicht.

#### **6.3.1 Datenbanksystem**

Das Datenbanksystem dient der Speicherung und Verwaltung der Daten. Kap. 2 behandelt ausführlich die notwendigen Überlegungen zum Aufbau und zur Manipulation einer Datenbasis. Die Wahl des zu verwendenden Datenbankmanagementsystems ist nicht zuletzt davon abhängig, ob lediglich Sachdaten oder Sach- und Geodaten zentral verwaltet werden sollen. Das Bearbeiten räumlicher Daten verlangt spezielle SQL-Funktionen, die nur in ausgewählten Datenbankmanagementsystemen zu finden sind. Eines dieser objektrelationalen Systeme wird vom Marktführer ORACLE angeboten.

### 6.3.1.1 Relationale Systeme

Die Palette relationaler Datenbankmanagementsysteme reicht von den proprietären Produkten Microsoft SQL Server und Access sowie IBM DB2 bis zu frei erhältlichen wie PostgreSQL, mSQL und MySQL. Sämtliche Systeme enthalten alle notwendigen Funktionen zur Bearbeitung von Sachdaten im Rahmen eines kartographischen Visualisierungssystems. Beim Einsatz von Datenbanksystemen in diesem Bereich steht neben der Internet-Anbindung die Geschwindigkeit als wesentliches Kriterium im Vordergrund. Insbesondere MySQL zeigt sich hierbei als gute Lösung, da dieses Produkt insbesondere auf Internetfähigkeit und Geschwindigkeitsoptimierung hin entwickelt wurde. Neben diesen Kriterien stellt MySQL ein sicheres und einfach zu bedienendes Datenbankmanagementsystem dar [MYSQL 2000]. Nicht zuletzt die Tatsache, dass es frei erhältlich ist, macht MySQL insbesondere für den kartographischen Einsatz im Internet zu einer echten Alternative zu proprietären Produkten.

### 6.3.1.2 Objektrelationale, -orientierte Systeme

Das Produkt *Spatial* ist ein auf dem objektrelationalen Datenmodell basierendes Datenbankmanagementsystem, das in der Lage ist, sowohl Geoobjekte in Vektorform als auch deren Attribute zu bearbeiten. Die technische Umsetzung erfolgt hierbei gemäß den Spezifikationen des OpenGIS® Consortium (vgl. Kap 2.6.3).

Als wesentliche Merkmale von ORACLE Spatial wären zu nennen [ORACLE 2000]:

? **Speicherung von Vektorgeometrie und Sachdaten:**

Das Speichern von Geo- und Sachdaten erfolgt in einem Datenbanksystem. Das Bearbeiten der Daten ist mittels SQL möglich.

? **Indizierung von Vektorgeometrie:**

Große Geodatensätze werden in Segmente (engl.: tiles) geteilt, die mittels eines Index verwaltet werden. Dadurch ergibt sich eine deutliche Steigerung der Geschwindigkeit und Effizienz des Systems.

? **Abfrage räumlicher Beziehungen von Geoobjekten:**

Neben dem Einsatz von vergleichenden Operatoren sind vor allem Funktionen wie "Within Distance", "Intersect" und "Containment" zu nennen.

? **Einsatz räumlicher Analysen:**

Funktionen wie "Union" und "Difference" können direkt mittels SQL vom Datenbankmanagementsystem bearbeitet werden.

ORACLE Spatial bietet die Option, Geodaten nicht in einem objektrelationalen, sondern lediglich in einem relationalen Datenmodell abzuspeichern. Bei dieser Lösung sind die oben genannten Funktionen dann allerdings nicht verfügbar.

### 6.3.2 Datenbankschnittstelle

Eine Schnittstelle (auch Interface) bezeichnet allgemein die Verbindungsstelle zwischen zwei Hardware- oder Softwarekomponenten. Im vorliegenden Fall soll die Schnittstelle die Aufgabe erfüllen, über das Internet auf ein Datenbanksystem zuzugreifen, um mit den dort gespeicherten Daten arbeiten zu können. Die Schnittstelle dient dabei als Übersetzer zwischen den Schichten des Systems.

Es existieren diverse Datenbankschnittstellen am Markt, die in Funktionsumfang und Einsatzbereich variieren [MOSSGRABER 1997]:

? **Proprietäre Lösungen:**

Viele Datenbanksystemhersteller bieten mittels geeigneter Werkzeuge den Zugriff auf Datenbanksysteme an. Diese Lösungen sind zumeist leicht implementierbar und bieten hohe Performance. Nachteilig erweist sich allerdings die Tatsache, dass hierfür Schnittstellen verwendet werden, die jeweils nur das eigene Produkt unterstützen. Man ist somit auf den Einsatz dieses Produktes limitiert.

? **OBGC/JDBC:**

Die beiden Schnittstellen ODBC (Open Database Connectivity; Microsoft) und JDBC (Java Database Connectivity; Sun) stellen Definitionen für die Kommunikation von Anwendungen mit Datenbanken dar. Praktisch alle namhaften Hersteller von Datenbankmanagementsystemen unterstützen heute

diese Methoden. Da diese Schnittstellen lediglich Standards darstellen, ist es möglich, herstellerunabhängige Applikationen zu entwickeln.

### ? **Skriptsprachen**

Neben den oben erwähnten Lösungen, die in der Regel mehr oder weniger vorkonfigurierte Lösungen darstellen, bietet sich die Möglichkeit an, Datenbankschnittstellen eigens mittels Skriptsprachen zu programmieren. Dies birgt den großen Vorteil, dass die Umsetzung exakt auf den gewünschten Einsatzzweck optimiert werden kann. Falls es sich um ein CGI-Skript handelt, ist die Sprache prinzipiell frei wählbar. In der Vergangenheit kamen aber verstärkt "Perl", "C" und "Phyton" zum Einsatz. Neben diesen zeigt sich in jüngster Zeit ein verstärkter Einsatz von PHP. PHP stellt eine Skriptsprache dar, die die Fähigkeit hat, direkt auf Datenbanksysteme zuzugreifen. Es ist mittels dieser Sprache möglich, SQL-Anweisungen direkt in PHP-Code - und damit in weiterer Folge in HTML-Code - zu integrieren. Bei Bearbeitung des HTML-Codes durch den Webserver werden die eingelagerten PHP-Abschnitte - und damit auch die SQL-Anweisungen - bearbeitet und in HTML umgewandelt. Die fertige Datei wird als Resultat der Bearbeitung an den Client geschickt.

### **6.3.3 Mapserver**

Im Rahmen eines 3-Schicht-Modells stellt der Mapserver die Applikationsschicht dar und ist jener Teil des datenbankgestützten Mapserver-Systems, der die kartographische Ausdrucksform erstellt. In der Regel handelt es sich dabei um ein serverseitiges Programm, das in der Lage ist, aus einem Geomodell ein kartographisches Modell zu erstellen. Hierzu werden die vektorialen Objekte des Geomodells aufgrund ihrer Attributierung symbolisiert und visualisiert. Das so erstellte Kartenbild wird in ein Rasterbild umgewandelt und in Form einer HTML-Seite an den Client verschickt. Sendet dieser eine Anfrage zur Änderung des Kartenbildes, so wird aus dem Geomodell eine völlig neue Darstellung generiert und übersendet. Clientseitige Lösungen - etwa mittels Java-Applets - bieten eine vergleichsweise bessere Performance als serverseitige Mapserver, da nicht bei jeder Interaktion Verbindung mit dem Server aufgenommen werden muss. Demgegenüber steht der Nachteil, dass zur Darstellung der Abbildung häufig Plug-Ins installiert werden müssen, oder, im Falle von Java, der Browser javafähig sein muss.

Die gängigen Hersteller von proprietären Geoinformationssystemen - exemplarisch seien hier ESRI, MapInfo und SICAD genannt - bieten Mapserver-Erweiterungen ihrer GIS-Produkte an, die nach oben beschriebenen, serverseitigen Verfahren arbeiten. Neben den Systemen, die im kommerziellen GIS-Umfeld anzusiedeln sind, existieren diverse frei erhältliche Programme, die Mapserver-Funktionalitäten aufweisen. Besonders erwähnenswert ist der von der University of Minnesota entwickelte Mapserver, der im Funktionsumfang an proprietäre Produkte heranreicht [UMN 2001].

#### 6.3.4 Webserver

Der Webserver (auch Internet- oder HTTP-Server) ist jene Software, mit der der Browser auf Seite des Clients mit dem Server Kontakt aufnehmen kann. Der Webserver bearbeitet die Anfragen und erstellt neue HTML-Seiten, die dieser wieder an den Client zurückgesendet. Anfragen an den Webserver werden in standardisierter Form übersendet. Hierzu bedient man sich einer URI (Uniform Resource Identifier), die den exakten Ort einer gesuchten Ressource angibt. Im Falle des WWW hat eine URI in der allgemeinen Form das folgende Aussehen:

*http://servername/seite*

Zusätzlich können dem Webserver weitere Parameter übergeben werden, die an die URI angehängt werden:

*http://servername/seite?parameter\_1=wert\_1&parameter\_n=wert\_n*

Die Angabe dieser Parameter macht es in kartographischen Applikationen möglich, Interaktionen des Nutzers wie etwa Änderung des Bildausschnittes oder Maßstabsänderungen an den Server zu übermitteln. Darüber hinaus können die Parameter Angaben zur Datenbankabfrage benötigter Daten beinhalten .

### 6.3.5 Benutzeroberfläche

Interaktion und Kommunikation zwischen dem Nutzer und dem System finden über die Benutzeroberfläche statt. FRANK (1993) bezeichnet die Benutzeroberfläche als "the part of the system with which the user interacts. It is the only part directly seen and thus 'is' the system for the user". Die Gestaltung der Benutzeroberfläche sowie die darin angebotenen Interaktions- und Eingabemöglichkeiten wirken somit direkt auf die Konsistenz und Verständlichkeit des gesamten Mapserver-Systems.

Das Konzept der Benutzeroberfläche hat sich im Laufe der Computerentwicklung von rein textuellen Kommando- und Abfrageeingaben über die Verwendung von Formularen bis hin zu den heute üblichen, vollgraphischen Oberflächen gewandelt. Graphische Benutzeroberflächen - häufig wird der englische Terminus "Graphical User Interface" (GUI) verwendet - ermöglichen eine einfache und intuitive Nutzung der Applikation, wobei die direkte Manipulation der Darstellung mittels Maus oder Tastatur im Vordergrund steht.

Heute verwendete Benutzeroberflächen orientieren sich zumeist nach dem "Macintosh human interface design" und werden als WIMP-Modelle (windows, icons, menus, pointer) bezeichnet. Diese Oberflächen sind in praktisch allen gängigen Computerprogrammen zu finden und bedienen sich zumeist der Büro- und Schreibtischmetapher, die im Zuge der "office automation technology" bei Xerox PARC entwickelt wurde. Dieses Prinzip, das den Bildschirm als Schreibtisch mit verschiedenen Objekten wie Ordner oder Papierkorb darstellt, ist allerdings für kartographische Applikationen nicht zweckmäßig [MEISSNER 1997]. Es gibt daher umfangreiche Bemühungen, adäquate Prinzipien für "Geo-Graphische Benutzeroberflächen" zu erarbeiten, die besonders auf die Bedürfnisse und Funktionen der Darstellung räumlicher Sachverhalte ausgelegt sind. In diesem Zusammenhang sind besonders das OpenGIS® Consortium sowie die Commission on Visualization and Virtual Environments der International Cartographic Association zu nennen [OGC 2001 & ICAVIS 2001].

## 6.4 Funktionsweise

Im Folgenden soll die Funktionsweise eines datenbankgestützten Mapserver-Systems in allgemeiner Form beschrieben werden. Ein Beispiel der praktischen Umsetzung einer derartigen Lösung wird in Kap. 7 gezeigt.

Die Erläuterungen zu den möglichen Formen der Systemarchitektur haben gezeigt, dass sich das 3-Schicht-Modell als ideale Architektur der Systemkomponenten anbietet. Datenbankschicht (Datenbanksystem), Applikationsschicht (Mapserver) und Präsentationsschicht (Benutzeroberfläche) stellen somit getrennt Teile der Applikation dar, die neben Datenbankschnittstelle und Webserver in ein Client/Server-Modell integriert werden müssen. Der Verteilung der Komponenten und die Arbeitsweise des Systems werden in folgender Abbildung gezeigt.:

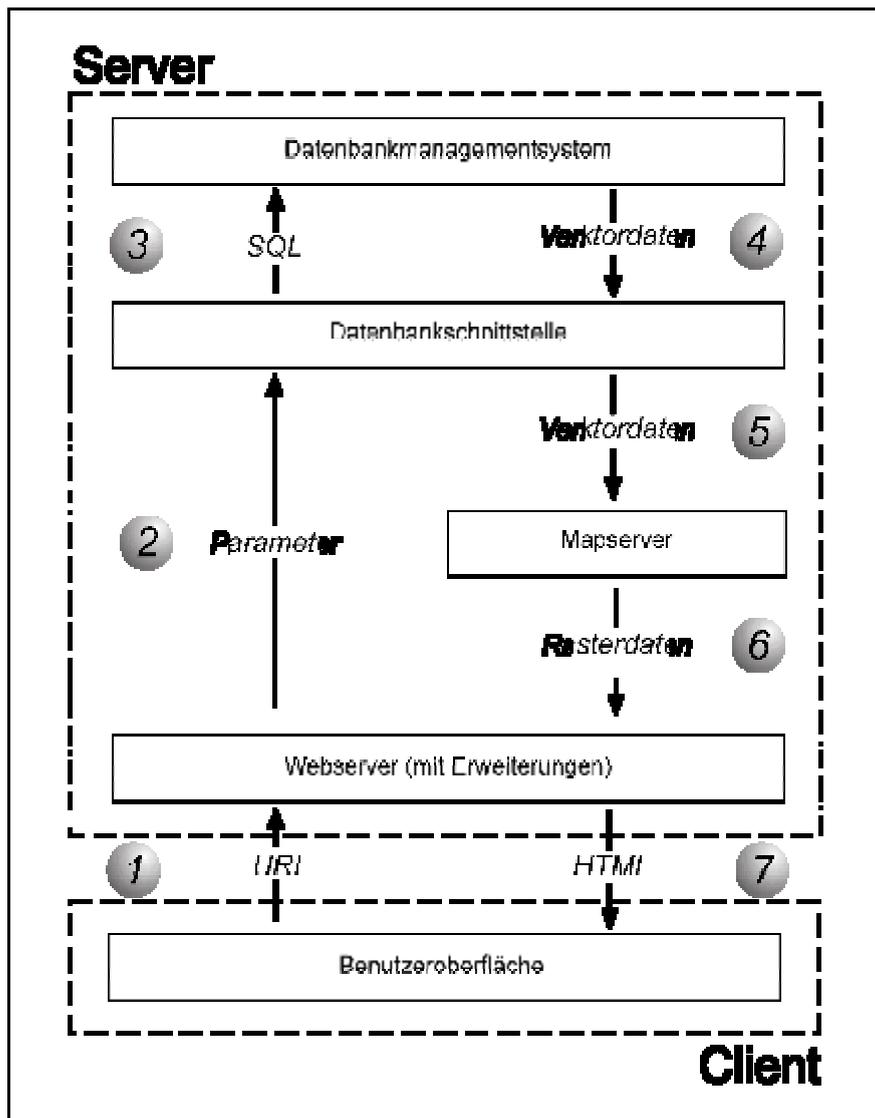


Abb. 19: Allgemeine Funktionsweise eines Internet-basierten datenbankgestützten Mapserver-Systems

Sämtliche Komponenten, bis auf die Benutzeroberfläche, befinden sich serverseitig. Dies führt zu einer Geschwindigkeitsoptimierung der Bearbeitung der Client-Anfragen, da sämtliche Arbeitsschritte lokal ausgeführt werden. Lediglich zur Übersendung der fertigen HTML-Seite mit der kartographischen Darstellung an den Client sowie deren Anfragen zurück an den Server muss eine Verbindung mittels eines Netzwerkes aufgebaut werden.

Der Arbeitsablauf, beginnend mit der Anfrage des Clients bis zum Übersenden der Ergebnisseite, durchläuft folgende Schritte:

1. Übersenden der Anfrage von Client an Server mittels standardisierter URI.
2. Der Webserver wertet die Anfrage aus, beginnt mit dem Erstellen einer neuen HTML-Seite. Dazu ist es notwendig, benötigte Daten aus dem Datenbanksystem abzufragen. URI-Parameter, die hierfür bestimmt sind, werden an die Datenbankschnittstelle übergeben.
3. Die Datenbankschnittstelle übersetzt die Anfrage in die SQL-Syntax und übergibt diese dem Datenbankmanagementsystem.
4. Das Datenbankmanagementsystem wertet die Anfrage aus und übergibt die angeforderte Daten an die Datenbankschnittstelle.
5. Die Datenbankschnittstelle übergibt die Daten im Vektorformat an den Mapserver
6. Der Mapserver erstellt die kartographische Darstellung mittels der übergebenen Daten aus dem Datenbanksystem in Form einer Rastergraphik und übergibt diese an den Webserver.
7. Der Webserver komplettiert die Erstellung der HTML-Seite und übersendet diese an den Client.

Das wiederholte Absenden einer Anfrage vom Client bedingt den neuerlichen Durchlauf der angeführten Arbeitsschritte. In der Praxis zeigt sich, dass die Erstellung einer kompletten HTML-Seite samt kartographischem Inhalt etwa 2 bis 3 Sekunden benötigt. Als möglicher Flaschenhals des Systems erweisen sich somit nicht die diversen Arbeitsschritte am Server, sondern die Übermittlung der Daten an und vom Client.

## **7 Praktische Umsetzung eines datenbankgestützten Mapserver-Systems im Internet**

Die praktische Umsetzung der Mapserver-Technologie erfordert vom Kartographen die Fähigkeit, die einzelnen Systemkomponenten an die Anforderungen und Bedürfnisse der zu erstellenden Applikation anzupassen. Es ist somit im Vorfeld der praktischen Arbeiten festzulegen, welche Absichten man verfolgt. Diese Intentionen geben in weiterer Folge die benötigten Daten vor, ebenso wie die Funktionen, die es zu implementieren gilt. Neben all diesen projektbezogenen Aspekten muss ebenso geklärt werden, welche Systemarchitektur man einsetzen möchte sowie welche Produkte man für die benötigten Systemkomponenten wählt.

Diese Fragen sowie die technische Realisierung eines datenbankgestützten Mapserver-Systems im Internet sollen in diesem Kapitel behandelt werden.

### **7.1 Intention**

Die Wiener Volkshochschule Simmering (VHS 11) veranstaltete im Herbst 1999 die Ausstellung "Kündigungsgrund: Nichtarier". Diese behandelte die Thematik der Vertreibung jüdischer Mitbürger aus Wiener Gemeindebauten in den Jahren 1938 und 1939. Mehr als 2.000 Kündigungsverfahren wurden damals von der Wiener nationalsozialistischen Stadtverwaltung angestrebt und durchgeführt. Viele der delogierten Juden wurden kurze Zeit später in Konzentrationslager deportiert. Im Zuge der Recherchen für die Ausstellung wurden die Daten vieler betroffenen Personen gesammelt und in einer Datenbank gespeichert [EXENBERGER, KOSS & UNGAR-KLEIN 1996].

Neben der Auflistung der Datensätze in diversen Publikationen entstand die Idee, die Thematik in Form einer Internet-Seite zu behandeln. Dieser Wunsch wurde an das Institut für Geographie und Regionalforschung der Universität Wien herangetragen. Es ergab sich die Gelegenheit, die praktische Umsetzung des kartographischen Teils dieser Seite im Rahmen meiner Diplomarbeit exemplarisch zu realisieren.

Hauptfunktion der zu erstellenden Seite sollte die Möglichkeit sein, Datensätze aus einer Datenbank zu selektieren und anschließend in einer kartographischen Darstellung von Wien zu visualisieren. Die Anforderungen an die Applikation bestanden im Wesentlichen aus zwei Teilen:

? **Suche in Datenbank -> Visualisierung in Karte:**

Als Ergebnis der Suche nach Name oder Wohnadresse einer Person wird der Inhalt der Datenbank zu diesem Eintrag tabellarisch dargestellt. Über einen Link besteht die Möglichkeit, den Standort der Wohnadresse in einer Karte von Wien zu visualisieren.

? **Suche in Karte -> Informationen aus Datenbank:**

Mit Hilfe entsprechender Werkzeuge (Zoom, Pan, Selektion) soll es möglich sein, eine Wohnadresse zu lokalisieren und zu selektieren. Auf Wunsch werden die zu der Adresse in der Datenbank gespeicherten Informationen angezeigt.

## **7.2 Daten**

Aufgrund der oben genannten Anforderungen können die benötigten Daten abgeleitet werden. Neben den notwendigen Sachdaten zu den betroffenen Personen und Wohnadressen muss eine Geodatenbasis von Wien in ausreichender Qualität vorhanden sein, um Darstellungen im Bereich von unter 200 Metern kartographisch zufriedenstellend darstellen zu können.

### **7.2.1 Datengrundlagen**

#### **7.2.1.1 Sachdaten**

Die Zahl der gekündigten Wohnungen beträgt über 2.000, die Zahl der betroffenen jüdischen Bewohner 4.500 bis 5.000. Die Daten zu diesen Personen wurden im Rahmen der Projekt-Datenbank "Österreichische Holocaustopfer" vom "Dokumentationsarchiv des

Österreichischen Widerstandes" (DÖW) gesammelt und werden laufend aktualisiert und erweitert.

Ein Auszug dieser Datenbank - nur bereits verstorbene Personen - wurde vom DÖW zur Verfügung gestellt und stand somit als Sachdatenbasis zur Verfügung. Die einzelnen Datensätzen der Datenbank enthalten folgende Informationen:

? Titel	? Interventionen
? Vorname	? Kündigungsadresse
? Zuname	? Weitere Adressen nach Räumung
? Mädchenname (bei Frauen)	? Letzte Adresse vor Deportation
? Geburtsdatum	? Deportationsdatum
? Geburtsort	? Deportationsort
? Beruf	? Todesdatum
? Kündigungsdatum	? Todesort
? Räumungsdatum	? Bemerkungen

### **7.2.1.2 Geometriedaten**

Die Geometriedaten müssen eine ausreichende Qualität aufweisen, um den städtischen Bereich im Maßstabsbereich von etwa 1:1.000 bis 1:10.000 darstellen zu können. Um jede Wohnadresse eindeutig verorten zu können, ist es notwendig, die koordinative Position jeder Hausnummer zu kennen. Diese Informationen müssen daher zwingend in den Geometriedaten enthalten sein.

Alle zu verortenden Adressen liegen innerhalb des Wiener Stadtgebietes. Es war daher naheliegend, auf den Datenbestand der Stadt Wien zurückzugreifen. Freundlicherweise wurden von der Gruppe "Grafische Datenverarbeitung" (GDV) der Magistratsabteilung 14 (MA 14-ADV) Probedatensätze zur Verfügung gestellt [MA 14-ADV 2001]. Diese Daten entstammen dem Räumlichen Bezugssystem Wien (vgl. Kap. 3.4.2.1). Die vorliegenden Ausführungen zur Realisierung der Mapserver-Applikation basieren auf diesen Daten.

Folgende Geometriedatensätze standen zur Verfügung:

? **Realnutzung:**

Polygontopologie mit Attributinformation. Als wesentliche Attribute sind zu nennen:

- ? Flächen-ID
- ? Realnutzungscode
- ? Bezirk
- ? Fläche
- ? Umfang

Die Realnutzungscodierung konnte mittels einer mitgelieferten Referenztabelle aufgeschlüsselt werden.

? **Straßennetz:**

Linientopologie mit Attributinformation. Als wesentliche Attribute sind zu nennen:

- ? Linien-ID
- ? Kennzahl des Straßenabschnitts
- ? Länge

Das Straßennetz ist in Straßenabschnitte geteilt, die jeweils von einem Kreuzungspunkt zum nächsten führen.

? **Adressen:**

Die Informationen zu sämtlichen Adressen im zu bearbeitenden Gebiet lag in CSV-Datei (Comma Separated Values)- durch Beistrich getrennte Werte - vor. Jeder Datensatz enthielt folgende wesentliche Attribute:

- ? Adressen-ID
- ? Koordinatenwerte X und Y
- ? Kennzahl des Straßenabschnitts

- ? Blocknummerierung (aus der Bezirk abgeleitet werden kann)
- ? Straßename
- ? Hausnummer

## **7.2.2 Datenaufbereitung**

Sowohl Sachdaten als auch Geometriedaten bedurften einer eingehenden Bearbeitung, um dem geplanten Verwendungszweck dienen zu können.

### **7.2.2.1 Sachdaten**

Der Sachdatensatz wurde vom DÖW als "Microsoft Excel"-Datei bereitgestellt und umfasste etwa 1.300 Einträge. Die Informationen lagen - soweit möglich - vollständig vor, waren aber nicht in dem Ausmaß strukturiert, um sie direkt in ein Datenbanksystem zu importieren. Es musste daher noch eine umfangreiche Überarbeitung der Daten erfolgen. Dies war notwendig, um die Konsistenz der Daten im Datenbanksystem zu gewährleisten. Daneben sollte die Datenbank so gestaltet werden, dass ein Maximum an Abfragemöglichkeiten gewährleistet ist.

Exemplarisch soll die Überarbeitung am Attribut "Räumungsadresse" gezeigt werden. Im Originaldatensatz lag die Information der Kündigungsadresse einer Person in Form eines Datenfeldes vor:

Räumungsadresse: 1080, Tulpengasse 4

Um eine Optimierung dieser Daten zu erreichen, wurde die vorliegende Information in drei Datenfelder aufgeteilt:

Räumungsbezirk: 1080

Räumungsstraße: Tulpengasse

Räumungsnummer: 4

Diese Struktur macht es nun etwa möglich, gezielt alle auftretenden Fälle in einem Bezirk zu selektieren. Diese Vorgangsweise wurde analog bei allen Attributen angewandt, die Ortsangaben beinhalten. Neben diesen Änderungen mussten alle Datumsangaben in eine adäquate Form gebracht werden - YYYY-MM-DD - um zeitbezogene Abfragen - etwa "...geboren vor 01.01.1901" - durchführen zu können. Zudem musste gewährleistet sein, dass Attribute wie "Titel" und "Beruf" nur standardisierte Werte enthalten, um Inkonsistenzen wie etwa "Arzt/Ärztin" oder "Zahnarzt/Dentist" zu vermeiden. Zuletzt erhielten die Datensätze eine eindeutige ID, die der späteren Verknüpfung mit den Geometriedaten dient.

Die so aufbereiteten Daten wurden als CSV-Datei aus "Excel" exportiert und im letzten Arbeitsschritt als Datenbanktabelle "PERSONEN" in das Datenbankmanagementsystem eingespielt.

#### **7.2.2.2 Geometriedaten**

Die von der MA 14 -ADV zur Verfügung gestellten Daten lagen im "ESRI Arc/Info“-Exportformat "e00" vor. Dieses Datenformat erlaubt es, Geometriedaten mit ihren zugehörigen Attributen aus einem Geoinformationssystem zu exportieren und anderorts wieder zu importieren.

Die flächenhaften Objekte des Datensatzes zur Realnutzung waren nach 42 Nutzungskategorien gegliedert. Da die Visualisierung aller Kategorien eine viel zu komplexe Darstellung für den geplanten Verwendungszweck bewirkt hätte, wurden sämtliche Flächen in eine von sechs definierten Nutzungsklassen eingeteilt:

- |                        |                           |
|------------------------|---------------------------|
| ? Öffentlicher Verkehr | ? Kirche                  |
| ? Grünfläche           | ? Schule                  |
| ? Wohngebiet           | ? Öffentliche Einrichtung |

Die betreffenden Flächen jeder Nutzungsklasse wurden aus dem Gesamtbestand selektiert und gesondert abgespeichert. Erfahrungswerte zeigen, dass sich eine derartige Struktur auch deshalb anbietet, da jede Nutzungsklasse im Rahmen der Visualisierung als einzelne Ebene angesprochen und bearbeitet werden kann.

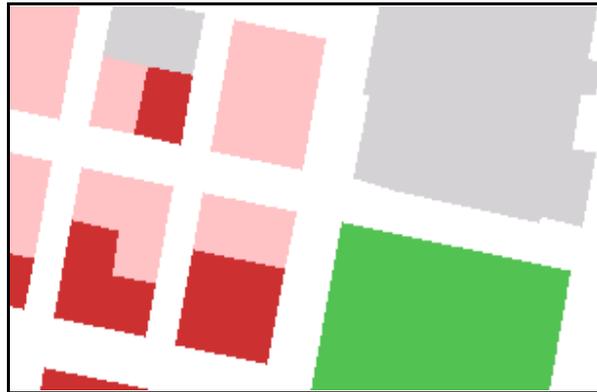


Abb. 20: Darstellung der Realnutzung nach definierten Nutzungsklassen

Die Darstellung der Verkehrsflächen erfolgt nicht durch die Liniengeometrie des Straßennetzes, sondern ergibt sich automatisch aus der Darstellung der flächenhaften Nutzung als jenen Flächen, die keiner Nutzungsklasse zugeordnet sind. Das linienhafte Straßennetz dient allein der Platzierung der Straßennamen und wird in der Visualisierung transparent dargestellt. Der Originaldatensatz "Straßennetz" umfasste allerdings vorerst kein Attribut "Straßenname". Dieses musste durch eine Verknüpfung mit dem Datensatz "ADRESSEN" über den Schlüssel "Kennung des Straßenabschnitts", der in beiden enthalten ist, zugänglich gemacht werden.

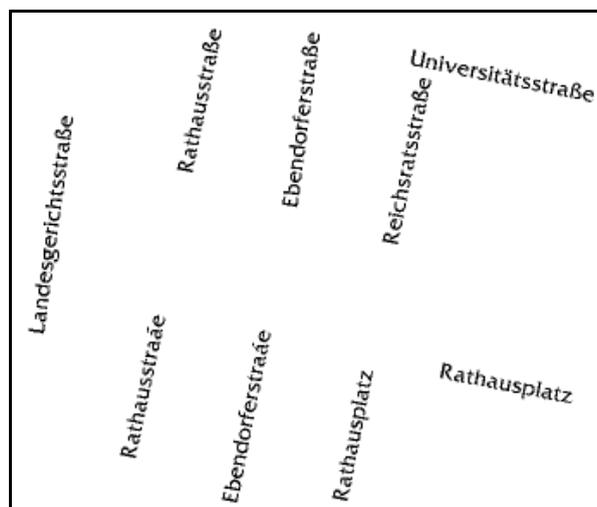


Abb. 21: Platzierung der Straßennamen mittels Liniengeometrie des Straßennetzes

Zur Verortung der Adressen wurde der Datensatz "ADRESSEN" herangezogen. Mittels des Arc/Info-Befehls "generate" wurde die ASCII-Datei in ein Punktcoverage umgewandelt. Anhand der Sachdaten wurden jene Adressen, die als Räumungsadresse einer Person aufscheinen, im Gesamtbestand der Adressen selektiert und in einem gesonderten Coverage abgespeichert. Dadurch ist es möglich, den Standorten der Räumungen Signaturen zuzuweisen.

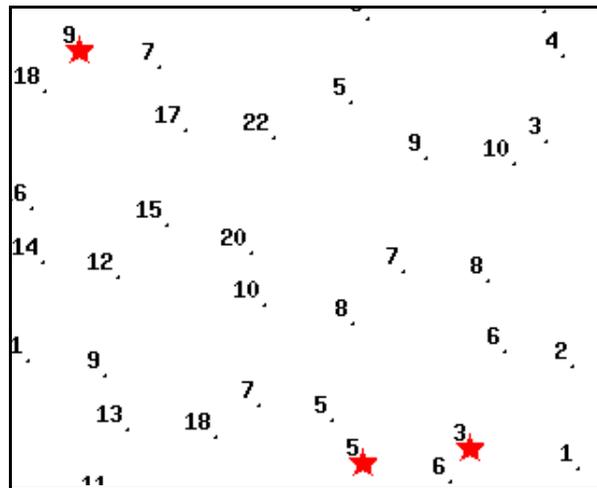


Abb. 22: Visualisierung von Räumungsadressen  
im Datenbestand 'Adressen'

Alle Geometriedaten mussten in ein Format gebracht werden, welche die Bearbeitung zur Laufzeit ohne ein unterstützendes Geoinformationssystem ermöglicht. Aus Kompatibilitätsgründen zur verwendeten Mapserver-Software wurde das Vektorformat "ESRI Shapefile" gewählt. Dieses speichert Geoobjekte mit deren Attributierung als geometrische Formen (engl.: shape) ab, die durch ihre Vektorkoordinaten definiert werden [ESRI 1998].

Die Information des Datensatzes "ADRESSEN" wurde in einem weiteren Arbeitsschritt in das Datenbankmanagementsystem eingespielt, um über den Schlüssel "Straßenname + Hausnummer" eine Verknüpfung mit den Sachdaten zu ermöglichen. Eine Abfrage einer Person im Sachdatensatz - etwa über deren Vor- und Zunamen - liefert somit automatisch die Koordinaten der Räumungsadresse dieser Person. Dies wird benötigt, um eine der Anforderungen der Applikation - Suche in Datenbank -> Visualisierung in Karte - zu ermöglichen.

## **7.3 Technische Umsetzung eines Prototypen**

Der technischen Umsetzung vorausgehend ist die Entscheidung, welche Komponenten eingesetzt werden sollen, und wie diese in einer passenden Systemarchitektur in Beziehung zueinander gesetzt werden.

Die vorliegende Internet-basierte Applikation soll dem 3-Schicht-Modell folgen. Neben Datenbanksystem, Mapserver und Benutzerschnittstelle werden hierfür eine Datenbankschnittstelle und ein Webserver benötigt. Im Datenbanksystem sollen lediglich die Sachdaten gespeichert und bearbeitet werden. Das fertige Produkt soll mittels einer serverseitigen Technologie arbeiten und dem Nutzer ein Ergebnis in Form einer reinen HTML-Seite liefern. Somit ist gewährleistet, dass die Applikation auf jedem handelsüblichen Browser zu betrachten ist. Das gesamte System soll zur Laufzeit ohne den Einsatz eines unterstützenden Geoinformationssystems betrieben werden.

### **7.3.1 Verwendete Komponenten**

Um den exakten Aufbau und die Funktionsweise des Internet-basierten datenbankgestützten Mapserver-Systems so weit als möglich kontrollieren und beeinflussen zu können, wurde nicht auf ein proprietäres System zurückgegriffen. Diese Systeme entsprechen vielfach vorgefertigten, standardisierten Produkten, die wenig Entwicklungsspielraum bieten. Aus diesem Grund sollten frei erhältliche Softwarekomponenten die Basis des Systems darstellen. Die Entwicklungen im Bereich freier GIS- und Kartographiesoftware haben in den letzten Jahren vermehrt "Open-Source-Produkte" hervorgebracht, die im Vergleich zu kommerziellen Anwendungen einen ähnlichen Funktionsumfang aufweisen. "Open Source" stand lange als Synonym für kostenlose, dafür aber schwer implementierbare und schlecht dokumentierte Software. Die Entwicklung derartiger Produkte erfolgt nicht in kommerziell orientierten Softwareschmieden, sondern wird von unzähligen Programmierern freiwillig und dezentral durchgeführt. Der Kreis der Anwender beschränkte sich zumeist auf Spezialisten, die aus unterschiedlichen Gründen nicht auf die im Handel verfügbaren Produkte zurückgreifen. In der Kartographie und Geoinformation führte in den letzten Jahren nicht zuletzt die steigende Popularität des unter der "General Public License" (GPL) stehenden Betriebssystems LINUX zu einer wachsenden Diskussion über die Themen Open Source

und freie Software. Die GPL ist als Lizenzurkunde der "Free Software Foundation" zu sehen. Sie legt die Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung von freier Software fest [FSF 2001]. "Frei" steht in diesem Zusammenhang allerdings weniger für kostenfrei, als vielmehr für Freiheit im Sinne von Recht auf Zugang, Veränderung und Benutzung des Programmiercodes.

Die Verwendung freier Software ist eng mit dem Einsatz des Betriebssystems LINUX verbunden. Da die Vielzahl der angebotenen Produkte für LINUX entwickelt wurden, wurde es auch als Betriebssystem für die vorliegende Applikation gewählt.

### **7.3.1.1 Datenbankmanagementsystem - MySQL**

MySQL ist ein relationales Datenbanksystem, das von der schwedischen Firma MySQL AB entwickelt und vertrieben wird. Seit geraumer Zeit steht MySQL dem interessierten Nutzer als freie Software unter der GPL zur Verfügung. MySQL bietet einige wesentliche Vorteile, die es für die Nutzung im Rahmen eines Internet-basierten Systems besonders empfehlenswert macht. Hier sind besonders die enge Verknüpfung mit dem Internet wie auch der Geschwindigkeitsvorteil gegenüber anderen Produkten zu nennen. Die Entwicklung von MySQL orientiert sich besonders an der Verwendung von Datenbanken im Internet und deren Bearbeitung on-line. Hierfür stehen umfangreiche Programmierschnittstellen zur Verfügung. Die Popularität von MySQL ist auch auf die Tatsache zurückzuführen, dass die Geschwindigkeit der Bearbeitung kleiner bis mittelgroßer Datenbanken von keinem anderen Datenbankmanagementsystem erreicht wird. Gerade beim Einsatz von interaktiv zu gestaltenden Karteninhalten erweist sich diese Tatsache als sehr positiv, da die Bearbeitungszeit einer Anfrage so deutlich reduziert werden kann.

### **7.3.1.2 Mapserver - UMN MapServer**

Die Auswahl an frei erhältlichen kartographischen Visualisierungswerkzeugen wächst stetig. In den meisten Fällen handelt es sich dabei aber um einfache und funktionsarme Java-Programme zur Darstellung kartographischer Inhalte oder um freie Web-Erweiterungen an sich proprietärer Systeme. Der im Rahmen dieses Projektes verwendete Mapserver - der

UMN MapServer - stellt dagegen ein Produkt dar, das sich bei seiner Intention am Funktionsumfang kommerzieller Produkte orientiert. Der UMN MapServer wurde ursprünglich an der University of Minnesota (UMN) in Kooperation mit der NASA und dem Minnesota Department of Natural Resources (MNDNR) entwickelt und wird heute im Rahmen des NASA-Projekts TerraSIP weiter betreut.

Der UMN Mapserver weist folgende wesentliche Merkmale auf:

- ? Verarbeitung von Vektor- und Rasterdaten
- ? Möglichkeit der Kachelung von Vektor- und Rasterdaten
- ? Indexierung räumlicher Daten
- ? Maßstabsabhängige Visualisierungen
- ? Automatisierte Beschriftung räumlicher Objekte
- ? Automatische Erstellung von Zeichenklärung und Maßstabsleiste
- ? Selektion von Geoobjekten mittels Punkt- oder Flächenselektion
- ? Unterstützung von True-Type Vektorschriften
- ? Freie Gestaltungsmöglichkeiten der Ausgabeseiten

Dieser Funktionsumfang kann mit Hilfe von Eingriffen in den Programmiercode oder durch Hinzufügen von selbst programmierten Funktionen erweitert werden. Über die Verwendung von geeigneten Datenbankschnittstellen ist es möglich, Sachdaten direkt aus einem Datenbanksystem zu extrahieren und zu visualisieren. Im vorliegenden Fall kam eine weiterentwickelte Version des UMN MapServers der kanadischen Firma DMSolutions zum Einsatz [DMSOLUTIONS 2001].

### **7.3.1.3 Benutzerschnittstelle - HTML**

Als Benutzerschnittstelle dient eine vollgraphische HTML-Seite. Diese kann ein beliebiges Layout aufweisen, das allerdings auf die implementierten Funktionen abgestimmt sein muss. Soll die interaktive Eingabe von Suchbegriffen - etwa Namen oder Adressen - ermöglicht werden, muss die Benutzeroberfläche entsprechende Eingabemöglichkeiten aufweisen. Die Benutzerschnittstelle ist als HTML-Vorlage am Server gespeichert und wird bei Eintreffen einer Kartenanfrage mit den erstellten Karten vervollständigt und an den Client gesendet.

#### **7.3.1.4 Datenbankschnittstelle - PHP 3**

Als Datenbankschnittstelle dient die Skriptsprache "PHP Hypertext Preprocessor" (PHP) in der Version 3. PHP ist eine in HTML eingebettete Sprache, die eine serverseitige Erstellung interaktiver, dynamischer Internetinhalte ermöglicht. Die wachsende Bedeutung des Internet sowie von Datenbanksystemen führte zur Entwicklung dieser Sprache, die besonders in Hinblick auf Datenbankanwendungen im Internet entwickelt und optimiert wurde. Die Syntax von PHP ist ähnlich jener der objektorientierten Programmiersprache "C", weist daneben allerdings Funktionen und Methoden von "Java" und "Perl" sowie eigene Merkmale auf.

Neben der Funktion als Datenbankschnittstelle dient PHP auch der erweiterten Erstellung der HTML-Seiten im Rahmen der Webserver-Erweiterung

#### **7.3.1.5 Webserver (mit Erweiterungen) - APACHE (mit PHP 3)**

Als Webserver wird das Open-Source Produkt "Apache" der Apache Software Foundation (ASF) verwendet. Apache ist die führende Webserver-Software und wird weltweit auf etwa 2/3 aller Server eingesetzt. Als wesentliche Vorteile dieses freien Produkts gegenüber proprietären Lösungen sind die einfache Handhabung sowie die Einsetzbarkeit auf allen gängigen Betriebssystemen zu nennen.

Webserver sind im Allgemeinen auf die Bearbeitung von HTML beschränkt. Da daneben aber auch PHP verarbeitet werden soll, muss Apache um diese Fähigkeit erweitert werden. Zu diesem Zweck wird PHP als Webserver-Erweiterung installiert. Somit ist das System in der Lage, neben HTML auch PHP-Code zur Laufzeit zu verarbeiten.

### **7.4 Aufbau der Systemarchitektur**

Da vorerst nur Probedaten vorlagen, musste die Systemarchitektur des Prototypen so gestaltet werden, dass beim Austausch dieser gegen die vollständigen Daten möglichst wenig

Arbeitsaufwand zur Adaption des Systems anfällt. Sämtliche Funktionen und Merkmale der finalen Applikation wurden daher in den Prototyp integriert.

#### **7.4.1 Physischer Aufbau**

Die Beschreibung des physischen Aufbaus des Systems geht von einer korrekten Funktionsweise der Komponenten - Linux, Apache (mit PHP), MySQL und UMN MapServer - aus. Dokumentationen zu Installation dieser Produkte sind auf den Internet-Seiten der Hersteller sowie in der Fachliteratur zu finden. Bevor die Verbindung der einzelnen Systemkomponenten erfolgen konnte, musste allerdings die Mapserver-Software konfiguriert werden.

##### **7.4.1.1 Konfiguration des UMN MapServers**

Der UMN MapServer setzt eine bestimmte Verzeichnisstruktur aller verwendeter Daten voraus. Die gesamten Geodaten müssen in einem Geodatenverzeichnis des Mapservers als "Shapefiles" vorliegen. Weiters ist darauf zu achten, dem System benötigte Schriftarten zur Verfügung zu stellen. Sollen Rastergraphiken als Symbole verwendet werden, müssen diese in das entsprechende Verzeichnis kopiert werden. Im Rahmen des Funktionsumfangs des UMN MapServers ist es möglich, eigene Symbole mittels Definition von Vektorkoordinaten zu gestalten. Im Unterschied zu Rastergraphiken weisen diese Symbole alle Vorteile von Vektorgraphiken auf [vgl. Kap. 2.4.1.1).

Als zentrale Konfigurationsdatei des UMN MapServers fungiert das "map file". In dieser Datei werden sämtliche, für die kartographische Darstellung relevanten Eigenschaften aller Objekte definiert.

Folgende Objekte können - mittels deren Eigenschaften - bearbeitet werden:

? **Map Object:**

Definiert das Objekt Karte. Als wesentliche Eigenschaften stehen "Extend" (räumliche Ausdehnung), "Scale" (Maßstab der Darstellung), "Size" (Größe der Darstellung in Pixel) und "Units" (Einheiten) zur Verfügung.

? **Layer Object:**

Definiert das Erscheinungsbild und das Verhalten einzelner Geodatenätze als Ebenen (engl.: layer). Wesentliche Eigenschaften: "Data" (Name des Geodatenatzes), "Status" (Darstellungszustand: Ein/Aus) und "Type" (Punkt, Linie oder Fläche).

? **Class Object:**

Jedes "Layer Object" muss zwingend mindestens ein "Class Object" aufweisen. Dieses weist die graphische Ebene einer thematischen Klasse zu, um sie entsprechend dieser visualisieren zu können. Liegt mehr als ein "Class Object" für eine Ebene vor, wird die Klassenzugehörigkeit mittels Attributwerten oder logischen Ausdrücken bestimmt.

? **Label Object:**

Definiert, wie ein Objekt oder eine Ebene beschriftet wird. Zu erwähnen wären hierbei die Eigenschaften "Font" (Schriftart) und "Position" (Ausrichtung).

? **Legend Object:**

Definiert den Aufbau der Zeichenerklärung.

? **Reference Map Object:**

Definiert werden "Extend", "Size" und "Status" der Übersichtskarte (engl.: reference map).

? **Scalebar Object:**

Definiert den Aufbau der Maßstabsleiste. Die korrekte Darstellung erfolgt gemäß den "Units" des "Map Object". Daneben lassen sich Werte wie "Position" und "Size" frei festlegen.

? **Query Object:**

Definiert den Ablauf einer Datenabfrage. Zu bestimmen ist vor allem, nach welchem "Query Item" die Abfrage stattfindet.

? **Web Object:**

Definiert wesentliche Eigenschaften der Internetanbindung der MapServers.

Das vollständige "map file", das für den Prototypen verwendet wurde, kann im Anhang eingesehen werden.

Oben angeführter Auflistung ist zu entnehmen, dass der UMN MapServer bei jedem Aufruf drei Objekte - die kartographische Darstellung an sich, die dazu passende Übersichtskarte sowie eine Maßstabsleiste - erstellt. Diese müssen zur Übertragung an den Client in eine HTML-Seite eingebettet werden. Hierzu wurde eine PHP-Datei geschrieben, die als Vorlage (engl.: template) dient (vgl. Anhang – PHP-Template). Die dynamische Erstellung von HTML-Inhalten mittels PHP macht es möglich, aus dieser Vorlage genau jenen HTML-Code zu erstellen, der für die Beantwortung der Anfrage vom Client benötigt wird.

#### **7.4.1.2 Verknüpfung der serverseitigen Systemkomponenten**

Die Verknüpfung zwischen UMN MapServer und MySQL erfolgt über die Software-schnittstelle PHP. Hierzu war es notwendig, ein Skript zu schreiben, das bei erforderlicher Bearbeitung von Daten im Datenbanksystem die Verbindung zu diesem herstellt und die gewünschten SQL-Anweisungen selbstständig ausführt. PHP bietet hierfür vordefinierte Funktionalitäten an, um diese Schritte durchzuführen (vgl. Anhang - PHP-Skript Datenbankschnittstelle). Die Verknüpfung vom Mapserver und Webserver erfolgt automatisch über die Webserver-Erweiterung PHP, da diese technisch gesehen in Apache integriert wurde.

Die gewählte Methode der Verknüpfung an sich eigenständiger Produkte im Rahmen des Prototyps zeichnet sich besonders durch die Tatsache aus, dass sich daraus ein modularer Aufbau des Systems ergibt. Diese Struktur ist äußerst flexibel in bezug auf Änderungen einzelner Systemkomponenten. So wäre es etwa möglich, ohne großen Arbeitsaufwand die Datenbankkomponente von MySQL auf ein anderes Produkt umzustellen. Neben dieser Tatsache ist noch zu bemerken, dass neben dieser Produktflexibilität auch eine räumliche Flexibilität des Systems gegeben ist. Dies bewirkt, dass die einzelnen serverseitigen Systemkomponenten theoretisch auf verschiedene Rechner aufgeteilt werden können. Diese Idee wird besonders durch das Open GIS® Consortium verfolgt, das als Vision ein Visualisierungssystem sieht, das weltweit auf verschiedene Datenserver zugreifen kann, um die Kombination der dort gespeicherten Daten zentral darzustellen [OGC 1999].

Nach erfolgter Konfiguration der Mapserver-Software sowie Verknüpfung dieser mit MySQL befand sich der Prototyp in funktionsfähigem Zustand.

#### **7.4.2 Funktionsweise des Prototyps**

Der Prototyp des Mapserver-Systems für das Internet-Angebot im Rahmen des Projekts "Kündigungsgrund Nichtarier" wurde auf einem Institutsrechner des Instituts für Kartographie und Geoinformation der Universität Wien installiert, um einer ausgiebigen Test- und Evakuierungsphase unterzogen zu werden. Während dieser Phase wurde der Prototyp von dafür zugelassenen Personen intensiv benutzt. Dabei auftretende Fehler und Fehlfunktionen wurden ausgebessert. Die folgenden Erläuterungen beziehen sich auf den Prototypen während dieser Testphase und stellen nicht das fertig layoutierte Internetangebot dar.

Die Funktionsweise dieses speziellen Systems soll anhand der beiden möglichen Ablaufszenarien

- ? Suche in Datenbank -> Visualisierung in Karte
- ? Suche in Karte -> Informationen aus Datenbank

gezeigt werden.

Alle im System denkbaren Abläufe werden in Abb. 23 schematisch dargestellt und in der Folge eingehend diskutiert.

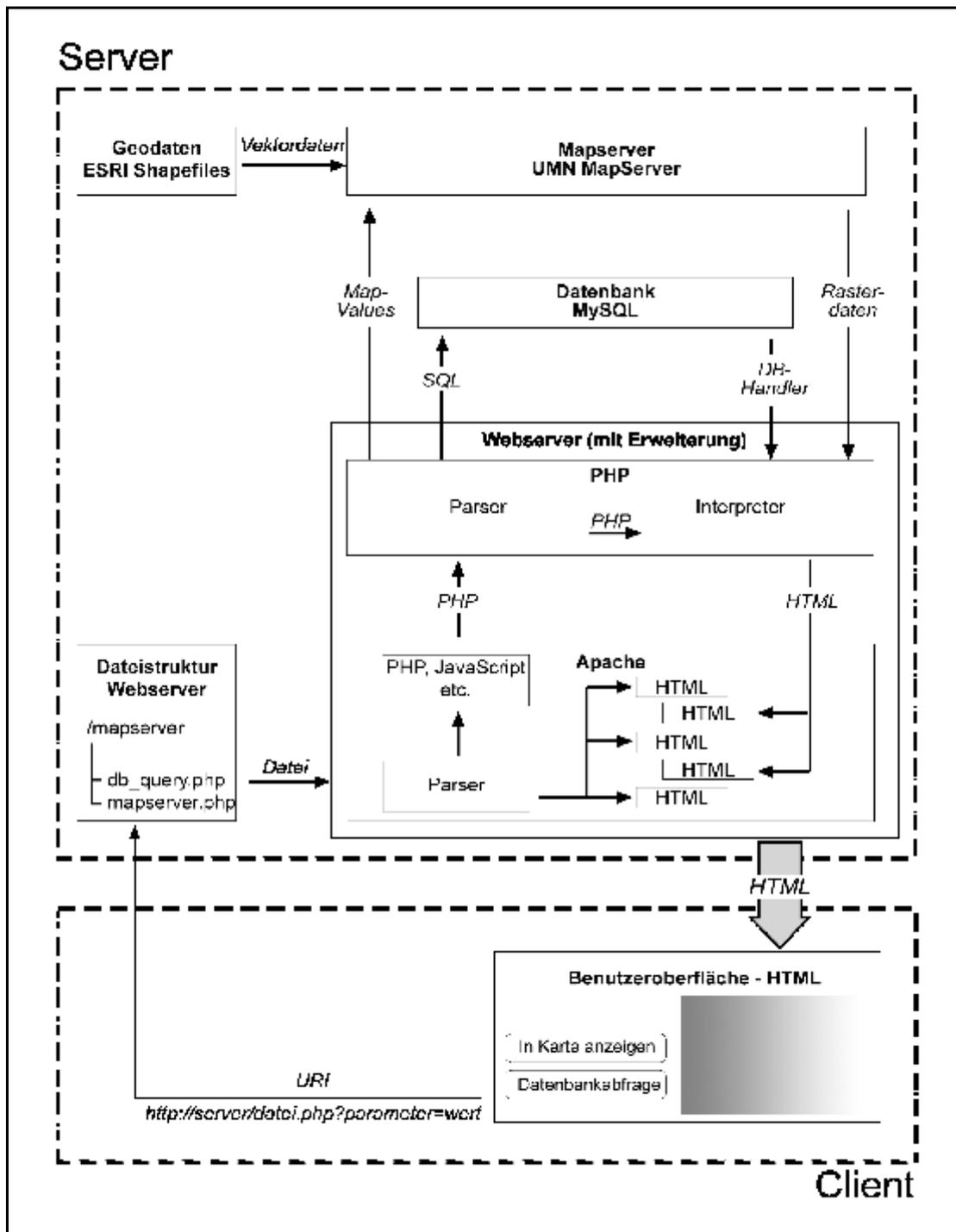


Abb. 23: Schematisches Ablaufdiagramm des Prototyps

### 7.4.2.1 Suche in Datenbank -> Visualisierung in Karte

Dem Nutzer soll die Möglichkeit geboten werden, aus einer Datenbank eine bestimmte Person zu selektieren. Der Ort der geräumten Wohnung dieser Person kann in der Folge in einer Karte visualisiert werden. Zu diesem Zweck steht dem Nutzer im Rahmen der Benutzeroberfläche ein Eingabeformular zur Verfügung, über das wahlweise Vor- und/oder Zuname sowie eine bekannte Adresse angegeben werden können. Exemplarisch soll der gesamte Arbeitsablauf von der Eingabe der Daten bis zur kartographischen Visualisierung anhand der Eingabe des Vornamen "Kurt" und des Zunamen "Wiener" dargestellt werden.

(1) Der Nutzer füllt die jeweiligen Eingabefelder aus und verschickt die Anfrage durch Druck auf den Knopf "Datenbankabfrage". Durch diese Interaktion des Nutzers wird eine URI an den Server geschickt, die neben der Angabe der Zieldatei - das PHP-Datenbank-schnittstellenskript "db\_query.php" - die benötigten Parameter zur Abfrage der gewünschten Person mitliefert. Die URI hat somit folgendes Aussehen:

*[http://server/db\\_query.php?vorname=kurt&nachname=wiener](http://server/db_query.php?vorname=kurt&nachname=wiener)*

(2) Nachdem geprüft wurde, ob sich die gewünschte Datei auf dem in der URI angegebenen Platz in der Dateistruktur des Servers befindet, wird diese an den Webserver zur Bearbeitung weitergeleitet.

(3) Apache fungiert in dieser ersten Phase des Systemablaufs als Sprachanalysator (engl.: Parser), der die Syntax der zu bearbeitenden Datei analysiert. Da PHP direkt in HTML-Code eingebettet wird, ist es dem Webserver möglich, die einzelnen Blöcke unterschiedlichen Codes zu trennen und gesondert zu bearbeiten. Das folgende, äußerst simple Skript zeigt, wie PHP-Code - in diesem Fall eine Funktion zur Anzeige der PHP-Systemparameter - durch die Tags "<?php" und "?>" direkt in HTML platziert werden kann.

```
<HTML>
  <?php
    phpinfo();
  ?>
</HTML>
```

Jene Teile der Datei, die aus HTML-Code bestehen, werden von Apache direkt bearbeitet und verbleiben beim Webserver.

(4) Codeabschnitte anderer Sprachen - im vorliegenden Fall PHP - werden von Apache an die dafür vorgesehenen Webserver-Erweiterungen geschickt und von diesen verarbeitet. Neben dem reinen Code werden auch die mit der URI mitgelieferten Parameter - hier "vorname=kurt&nachname=wiener" - an PHP übergeben.

(5) Der PHP-interne Parser analysiert den Code und extrahiert jene Teile, die für die Abfrage der Datenbank benötigt werden. Alle anderen PHP-Codeteile werden dem Interpreter übergeben.

(6) PHP wandelt die Variablenwerte "Kurt" und "Wiener" in die Variablen \$vorname und \$nachname um, baut diese in den extrahierten SQL-Code ein und verschickt diesen an das Datenbankmanagementsystem MySQL:

```
select
PERSONEN.VORNAME, PERSONEN.ZUNAME,
PERSONEN.GEBURTSORT, PERSONEN.BERUF,
PERSONEN.RÄUMUNGSDATUM, PERSONEN.RÄUMUNGSBEZIRK,
PERSONEN.RÄUMUNGSADRESSE, PERSONEN.RÄUMUNGSNUMMER,
PERSONEN.DEPORTATIONSdatum, PERSONEN.DEPORTATIONSORT,
ROUND(ADRESSEN.X)-150 as XMIN, ROUND(ADRESSEN.X)+150 as XMAX,
ROUND(ADRESSEN.Y)-150 as YMIN, ROUND(ADRESSEN.Y)+150 as YMAX
from PERSONEN, ADRESSEN
where
PERSONEN.RÄUMUNGSSTRASSE = ADRESSEN.STRASSENNAME
and
PERSONEN.RÄUMUNGSNUMMER = ADRESSEN.HAUSNUMMER
and
VORNAME = $vorname and NACHNAME = $nachname
```

Die HTML-Ergebnisseite der Abfrage soll neben den Informationen zur betreffenden Person auch einen Verweis (engl.: link) "In Karte anzeigen" darstellen. Daher ist es

notwendig, aus der Datenbanktabelle "ADRESSEN" die Koordinaten der Räumungsadresse auszulesen, um diese an den Mapserver zu übergeben. Dieser kann anhand der Koordinaten den passenden Kartenausschnitt erstellen. Da der UMN MapServer allerdings vier Koordinaten (Xmin, Ymin, Xmax, Ymax) zur eindeutigen Festlegung eines Bereichs der Karte benötigt, ist es notwendig, die Koordinaten X und Y jeweils mit einem Offset-Wert zu belegen (+/- 150m).

(7) Als Resultat der Abfrage schickt MySQL einen "Handler" an PHP zurück. Dieser fungiert als Zeiger auf alle tatsächlichen Datensätze, die den Bedingungen in der SQL-Abfrage entsprechen.

(8) PHP kann nun über den Handler die Daten aus der Datenbank auslesen und diese zusammen mit den zuvor bereits ausgefilterten PHP-Codeteilen interpretieren. Das Ergebnis wird in HTML umgewandelt und an Apache gesendet.

(9) Apache kombiniert die HTML-Codeteile, die bereits in Schritt (3) abgelegt wurden, mit jenen, die von PHP erzeugt wurden und erstellt die vollständige HTML-Seite.

(10) Die Ergebnisseite wird an den Client verschickt. Diese Seite enthält in tabellarischer Form die Informationen zur gesuchten Person aus der Sachdatentabelle "PERSONEN" sowie den angesprochenen Link zur Darstellung in der Karte.

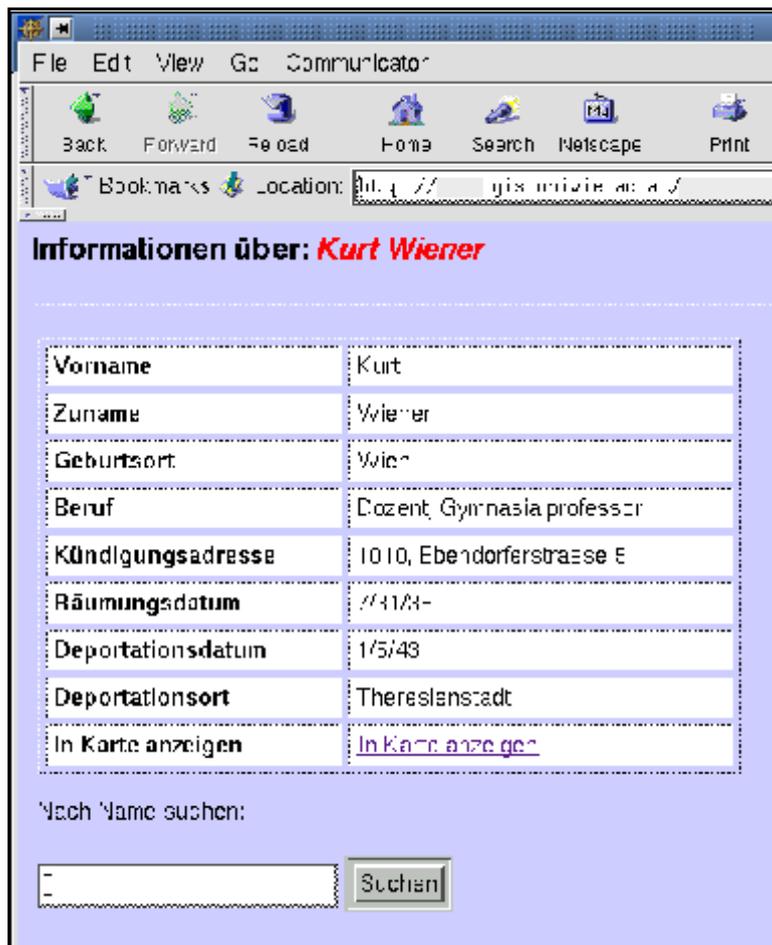


Abb. 24: Anzeige der vollständigen Information aus Datenbanksystem

Die Ausgabe der Ergebnisseite am Client schließt einen kompletten Systemdurchlauf ab. Um die gewünschte Adresse in der Karte zu visualisieren, muss das System erneut durchlaufen werden. Diesmal erfolgt der Funktionsablauf allerdings nicht über die Datenbankschicht, sondern über die Applikationsschicht - den Mapserver.

(11) Das Ausführen des Verweises "In Karte anzeigen" bedingt das neuerliche Absenden einer URI vom Client, diesmal ist allerdings "mapserver.php" - das Initialisierungsskript des UMN MapServers - als Zieldatei angegeben. Als Parameter werden die aus der Tabelle "ADRESSEN" berechneten Koordinatenwerte - hier allgemein als "xxx" bezeichnet - mit angeführt.

*http://servername/mapserver.php?Xmin=xxx&Ymin=xxx&Xmax=xxx&Ymax=xxx*

(12) Ähnlich wie beim Ablauf der zuvor erläuterten Abfrage der Datenbankinhalte wird die gewünschte Datei an den Webserver gesendet (vgl. (2)). Apache trennt die PHP-Datei, bearbeitet den HTML-Teil (vgl. (3)) und übergibt den PHP-Code sowie die Parameter "Xmin, Ymin, Xmax, Ymax" an PHP (vgl. (4)).

(13) PHP wandelt die Parameter in entsprechende Variablen um. In diesem Stadium ist es notwendig, umfangreiche Kontrollfunktionen der übermittelten Koordinatenwerte durchzuführen. So wird etwa bei jeder Generierung eines Kartenausschnitts geprüft, ob das gewünschte Gebiet vollständig innerhalb des verfügbaren Bereichs der Geometriedaten liegt. Ist dies nicht der Fall, wird der Kartenausschnitt entsprechend verschoben. Exemplarisch soll der Programmiercode dieser Funktion angeführt werden, um einen Einblick in die Syntax von PHP zu bekommen:

```
/* ----- */
/* Prüfung, ob Kartenausschnitt innerhalb Bereich der Geodaten */
/* ----- */
if ($dfNewMinX < $dfMaxExtMinX)
{
    $dfNewMinX = $dfMaxExtMinX;
    $dfNewMaxX = $dfNewMinX + ($dfDeltaX);
}
if ($dfNewMaxX > $dfMaxExtMaxX)
{
    $dfNewMaxX = $dfMaxExtMaxX;
    $dfNewMinX = $dfNewMaxX - ($dfDeltaX);
}
if ($dfNewMinY < $dfMaxExtMinY)
{
    $dfNewMinY = $dfMaxExtMinY;
    $dfNewMaxY = $dfNewMinY + ($dfDeltaY);
}
if ($dfNewMaxY > $dfMaxExtMaxY)
{
    $dfNewMaxY = $dfMaxExtMaxY;
    $dfNewMinY = $dfNewMaxY - ($dfDeltaY);
}
SetMapExtents($dfNewMinX, $dfNewMinY, $dfNewMaxX, $dfNewMaxY);
}
```

Ist die Korrektheit der Daten gegeben, werden diese an den MapServer übergeben.

(14) Aufgrund der übermittelten Parameter kann der MapServer die gewünschte Fläche aus den Geodaten - als Vektordaten - extrahieren. Die eigentliche Aufgabe des MapServers

besteht nun darin, aus diesen Daten das Kartenbild, die Übersichtskarte sowie die Maßstabsleiste als Rastergraphiken zu erstellen und an PHP zu übermitteln.

(15) Der übrige Funktionsverlauf bis zur Ausgabe der Ergebnisseite am Client verläuft wieder identisch zum besprochenen ersten Systemdurchlauf - der Datenbankabfrage. PHP kombiniert die Graphiken des MapServers mit jenen PHP-Codeteilen, die vom Parser zuvor bereits gefiltert und an den Interpreter gesendet wurden, und erstellt einen HTML-Code, der an Apache übergeben wird (vgl. (8)). Apache komplettiert die Erstellung der HTML-Seite (vgl. (9)) und übersendet diese an den Client (vgl. (10)).

Der Nutzer erhält als Ergebnis des zweiten Systemdurchlaufs eine Karte, deren Mittelpunkt die überlieferten Koordinaten des Standortes der geräumten Wohnung bilden.

Dieses Beispiel des Szenarios "Suche in Datenbank -> Visualisierung in Karte" hat gezeigt, welche möglichen Funktionsabläufe der Prototyp bietet. Um die zweite Anforderung an das System "Suche in Karte -> Informationen aus Datenbank" erfüllen zu können, ist es notwendig, die beschriebenen Systemdurchläufe in etwas modifizierter Form zu wiederholen.

#### **7.4.2.2 Suche in Karte -> Informationen aus Datenbank**

Mittels Navigieren in der Karte kann der Nutzer einen gewünschten Kartenausschnitt wählen, um in diesem durch Signaturen gekennzeichnete Standorte von geräumten Wohnungen zu selektieren und erste Informationen zu den ehemals dort lebenden Personen aus der Datenbank abzufragen. Ist nähere Information zu einer der Personen erwünscht, ermöglicht ein Verweis den Einblick in sämtliche, in der Datenbank gespeicherten Informationen zu dieser Person.

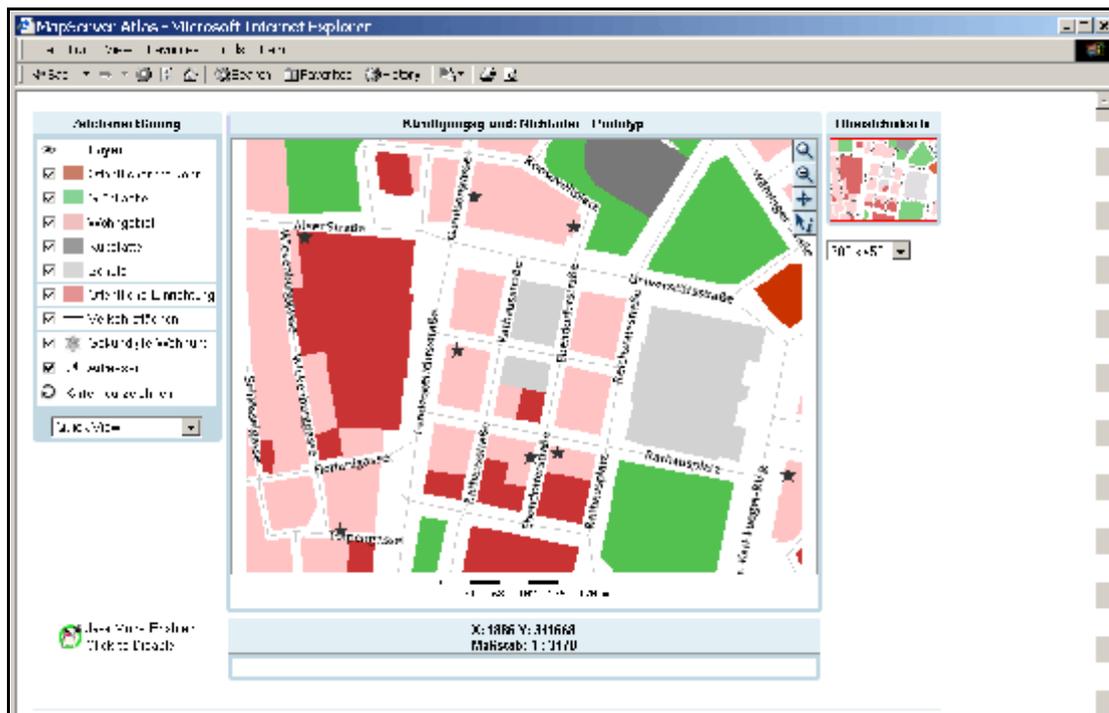


Abb. 25: Hauptbildschirm des Mapserver-Prototyps

Dieser Vorgang der Visualisierung vor der Datenabfrage bedingt einen Systemdurchlauf, der im Hauptbildschirm des Mapserver-Prototyps beginnt.

Der Hauptbildschirm des Mapservers ist in drei Abschnitte gegliedert, die von links nach rechts als Zeichenerklärung, Hauptkartfenster und Übersichtskarte bezeichnet werden können.

Die Zeichenerklärung wird automatisch erstellt, kann aber in Aussehen, Größe und Platzierung variabel gestaltet werden. Zu erkennen ist, dass die Elemente der Zeichenerklärung mit den Einträgen des "map file" ident sind und somit automatisch gemäß den Geodatensätzen erstellt werden. Im Rahmen der Zeichenerklärung hat der Nutzer die Möglichkeit, einzelne Ebenen ein- oder auszuschalten. Das "Quick View"-Auswahlmenü unterhalb der Zeichenerklärung bietet dem Nutzer die Möglichkeit, vordefinierte Bereiche der Karte anzusteuern. Im Rahmen der Visualisierung von Wien wäre es denkbar, hier die Wiener Gemeindebezirke als Auswahl anzubieten.

Rechts neben dem Hauptkartenfenster befindet sich die Übersichtskarte, die neben einer Orientierungshilfe auch eine Möglichkeit der Interaktion bietet. So wird durch Auswählen eines Punktes in der Übersichtskarte eine Karte im Hauptkartenfenster erstellt, deren Koordinaten des Mittelpunktes dem ausgewählten Punkt entsprechen.



Unterhalb der Übersichtskarte befindet sich ein Auswahlnenü, mit dem die Größe des Hauptkartenfensters bestimmt werden kann. So ist es dem Nutzer möglich, gegebenenfalls auf eine kleinere verfügbare Bildschirmfläche oder eine schlechte Netzwerkverbindung durch Verkleinerung der Darstellung zu reagieren.

Bei direkten Anfragen an den UMN MapServer muss grundsätzlich unterschieden werden, ob nur eine Änderung des Kartenausschnitts - mittels Pan- oder Zoomfunktion - gewünscht ist, oder eine Informationsabfrage der Geoobjekte stattfinden soll.

Eine Änderung des Kartenausschnitts wird prinzipiell interaktiv auf dem Hauptkartenfenster durchgeführt. Hierzu stehen folgende Werkzeuge aus der Werkzeugpalette - in der rechten oberen Ecke des Hauptkartenfensters - zur Verfügung:

-  ? Zoom in: Bildausschnitt verkleinern, Maßstab vergrößern
-  ? Zoom out: Bildausschnitt vergrößern, Maßstab verkleinern
-  ? Pan: Bildausschnitt in X-Y- Richtung verschieben
-  ? Information: Abfrage eines Geoobjekts

Eine Änderung des Kartenausschnitts bewirkt einen Arbeitsablauf, der im Wesentlichen dem Systemdurchlauf beim Erstellen der Karte im Szenario "Suche in Datenbank -> Visualisierung in Karte" entspricht. Wird etwa mit dem "Zoom-in"- Werkzeug ein verkleinerter Bildausschnitt gewählt, so werden dessen Koordinaten als Parameter für die Generierung der Karte herangezogen. Entspricht diese Darstellung nicht den Vorstellungen des Nutzers, so kann dieser mit Hilfe der Navigationswerkzeuge eine neue Karte vom MapServer anfordern, was mit dem neuerlichen Durchlauf des Systems verbunden ist.

Ist die gewünschte Kartenansicht präsent, können mittels der integrierten Informationsabfragemöglichkeit die Räumungsadressen abgefragt werden. Hierfür wurde jede Räumungsadresse mit einer eindeutigen ID versehen, die eine Verknüpfung mit den Sachdaten erlaubt. Die Informationsabfrage eines Geoobjekts in der Karte bedingt somit,

dass einerseits vom MapServer eine Karte erzeugt werden muss, andererseits aber auch in der Datenbank gespeicherte Informationen zu diesem Objekt abgefragt werden müssen.

Hierzu wird vom Client neben den Koordinaten der präsenten Karte die ID des abzufragenden Objekts als Parameter der URI an den Server gesendet.

```
http://servername/mapserv.php?Xmin=xxx&Ymin=xxx&Xmax=xxx&Ymax=xxx  
&id=z
```

Der Systemablauf erfolgt bis zur Analyse des PHP-Codes in der Webserver-Erweiterung gleich wie oben angeführte Beispiele (vgl. (1-5)). Da die Parameter von PHP in Variablen umgewandelt werden, ist es möglich, jene Werte, die für die Erstellung der Karte notwendig sind, dem MapServer zuzuordnen, und jene für die Datenbankabfrage in die SQL-Syntax einzubauen. Es erfolgt somit ein paralleler Zugriff auf den MapServer und das Datenbankmanagementsystem. Die SQL-Syntax der Informationsabfrage eines Standorts einer geräumten Wohnung sei angeführt:

```
select  
PERSONEN.VORNAME, PERSONEN.ZUNAME,  
PERSONEN.RÄUMUNGSBEZIRK, PERSONEN.RÄUMUNGSADRESSE,  
PERSONEN.RÄUMUNGSNUMMER  
from PERSONEN, ADRESSE  
where  
PERSONEN.RÄUMUNGSSTRASSE = (select ADRESSE.STRASSENNAME  
where ID = $id)  
and  
PERSONEN.RÄUMUNGSNUMMER = (select ADRESSEN.HAUSNUMMER  
where ID = $id)
```

Der weitere Ablauf sieht die bereits bekannten Schritte bis zur Übertragung der Ergebnis-seite an den Client vor (vgl. 8-10)).

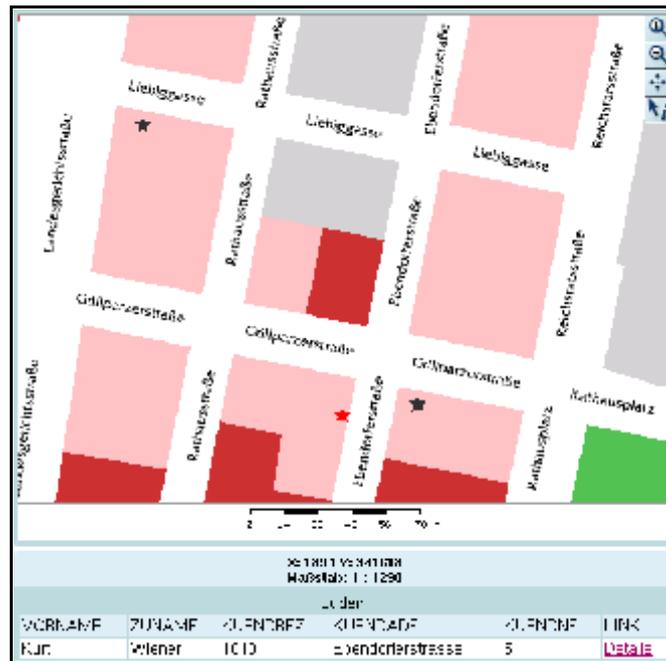


Abb. 26: Abfrage eines Geobjekts

In Abb. 26 wurde mittels "Zoom-in" ein verkleinerter Bildausschnitt gewählt sowie einer der Standorte der geräumten Wohnungen abgefragt. Als Ergebnis der Anfrage liefert das System erste Informationen über Adresse und zugehörige Person. Zusätzlich erfolgt eine farbliche Markierung (engl.: hilited) des abgefragten Objekts.

Zusätzlich zur dargestellten Information wird automatisch ein Link erzeugt, der die Abfrage sämtlicher Informationen zu dieser Person aus der Datenbank ermöglicht. Die Arbeitsschritte dieser Datenbankabfrage sind analog zur Datenbankabfrage in Kap. 7.4.2.1 zu sehen, und liefern ein Ergebnis gemäß Abb. 24 (vgl. (1-10)).

## 8 Zusammenfassung

Kartographische Darstellungen im Internet sind heute weit verbreitet. Vielfach stellen diese aber statische Abbildungen dar, die, einmal erzeugt, nicht mehr verändert werden können oder sollen. Ist die Aktualität des dargestellten Sachverhaltes nicht mehr gegeben, wird die Darstellung, wenn überhaupt, entfernt und nur selten durch eine aktualisierte Version ersetzt. Dies ist damit zu begründen, dass die Herstellung einer qualitativ hochwertigen Karte ein zeit- und kostenintensiver Prozess ist, und die Fortführung der Aktualität diese Intensität noch steigert.

Als mögliche Alternative könnten Applikationen zur automatischen Erstellung kartographischer Produkte angesehen werden. Diese erzeugen, im Rahmen der durch die Bildschirmdarstellung gegebenen Limitierungen, selbstständig kartographische Darstellung. Genau jene Beschränkungen im graphischen Bereich führen aber dazu, dass ein Qualitätsniveau, wie man es von gedruckten Produkten her kennt, nicht erreicht wird. Diesem Nachteil steht allerdings ein Vorteile gegenüber, den das Internet bietet – die Interaktion. TAYLOR (1996) bezeichnet die Möglichkeiten, die sich durch die Interaktion zwischen Nutzer und Karte ergeben, als eine der größten Herausforderungen der modernen Kartographie. Interaktion erlaubt es dem Nutzer, aus einem vorhandenen Datenmodell ein persönliches, auf seine Bedürfnisse abgestimmtes Graphikmodell erzeugen zu lassen.

Die Möglichkeit, aus einem Datenmodell beliebig viele Graphikmodelle zu erstellen, bedarf einer Vorgehensweise, die häufig als "Content/Presentation paradigm" bezeichnet wird. Hierunter versteht man die Trennung von Datenspeicherung und Datenvisualisierung. Die Visualisierung der Daten übernimmt – im vorliegenden Fall – der Mapserver, die Speicherung und Verwaltung das Datenbankmanagementsystem. Die automatisierte, dynamische, interaktive Erstellung kartographischer Darstellungen im Internet ist prinzipiell auch ohne den Einsatz von Datenbanksystemen möglich. Der zur Verfügung stehende Datenbestand wäre in diesem Fall allerdings statisch, die inhaltlichen und graphischen Interaktionsmöglichkeiten dadurch stark eingeschränkt.

Kartographische Visualisierungen im Internet mit Hilfe datenbankgestützter Mapserver-Systeme bieten die Option, auf sich verändernde, tagesaktuelle Datenbestände zurückgreifen zu können. Die Trennung von Speicherung (Datenbank), Visualisierung (Mapserver) und Präsentation (Benutzerschnittstelle) führt zu einem flexiblen, modularen

Aufbau der Systemarchitektur. Die einzelnen Teile des Systems können isoliert voneinander bearbeitet werden. In der Praxis ergibt sich dadurch die Situation, dass etwa bei Eingabe neuer Sachdatensätze in eine Datenbank, diese sofort und vollautomatisch visualisiert werden können, ohne dass der Bearbeiter der Datenbank Kenntnisse des Visualisierungs- oder Präsentationsprozesses haben muss.

Die Zukunftsaussichten für den Einsatz datenbankgestützter Visualisierungssysteme im Bereich der modernen Kartographie und Geoinformation sind sehr gut. Das wachsende Interesse an der Präsentation räumlicher Daten seitens Institutionen, die auf große Datenbestände zurückgreifen können sowie die enorme Nachfrage von Nutzerseite wird die technische Entwicklung verbesserter Software-Komponenten beschleunigen. Mit diesen Werkzeugen ist es dem Kartographen möglich, die Qualität der kartographischen Visualisierungen im Internet weiter zu steigern.

## Literaturverzeichnis

- ABREU, J., VAN DEN EIJNDEN, B., SCHOLTEN, H., GEHREIS, B.: Combining Spatial Metadata Search-Engine with Webmapping. In: GIS-Zeitschrift für raumbezogene Information und Entscheidungen, 13.Jg., 2000, Heft 5.
- AdV (Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland): ATKIS (Amtliches Topographisch-Kartographisches Informationssystem). Landesvermessungsamt Nordrhein-Westfalen, 1998.
- ARNBERGER, E.: Thematische Kartographie. Mit einer Kurzeinführung über EDV-unterstützte Kartographie und Quellen der Fernerkundung. Braunschweig, Westermann Verlag, 1993.
- BAHRENBERG, G., GIESE, E., NIPPER, J.: Statistische Methoden in der Geographie Band 1. Stuttgart, Teubner Studienbücher, 1990.
- BARTELME, N.: Die ÖNORMEN-Reihe A2261 vor dem Hintergrund der internationalen Normung. Vortrag TU-Graz ON-Seminar, 1999.
- BELADA, P.: Die Mehrzweckstadtkarte, geometrische Grundlage für das Wiener GIS - Stand des Projektes. In: STROBL, J., DOLLINGER, F.: Angewandte geographische Informationstechnologie VI. Beiträge zum GIS-Symposium 1994. Salzburg, Salzburger Geographische Materialien, 1994.
- BERLIN, K.: WebGIS-Einsatz im europäischen Forschungsverbundprojekt BASIS. In: STROBL, J., BLASCHKE, T.: Angewandte Geographische Informationsverarbeitung XI. Beiträge zum AGIT-Symposium 1999. Heidelberg, Wichmann Verlag, 1999.
- BILL, R., FRITSCH, D.: Grundlagen der Geoinformationssysteme. Karlsruhe, Wichmann Verlag, 1991.
- BRANDI-DOHRN, F.: GIS im WWW: Beispiele und Nutzen. In: STROBL, J., BLASCHKE, T.: Angewandte Geographische Informationsverarbeitung XI. Beiträge zum AGIT-Symposium 1999. Heidelberg, Wichmann Verlag, 1999.
- CARTWRIGHT, W., PETERSON, M., GARTNER, G.: Multimedia Cartography. Berlin, Springer-Verlag, 1999.
- CODD, E.: Relational database: A practical foundation for productivity. In: Communications of the ACM, 25. Jg., 1982, Heft 2.
- CUTHBERT, A.: User Interface with Geospatial data (=OpenGIS®Project Document 98-060). OpenGIS®Consortium, Inc., 1998.
- DMSolutions Group. Im WWW unter: <http://www.dmsolutions.ca/home.html>, 2001.

- DOYLE, A.: WWW Mapping Framework (=OpenGIS®Project Document 97-009). OpenGIS®Consortium, Inc., 1997.
- DVW (Deutscher Verein für Vermessungswesen e.V.): Gewinnung von Basisdaten für Geoinformationssysteme (= Schriftenreihe des DVW, Band 4). Stuttgart, Wittwer Verlag, 1992.
- ECO, U.: Wie man eine wissenschaftliche Arbeit schreibt. Stuttgart, UTB-Taschenbücher, 1993.
- ESRI (Environmental Systems Research Institute): ESRI Shapefile Technical Description. Im WWW unter: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>, 2001.
- EXENBERGER, H., KOSS, J., UNGAR-KLEIN, B.: Kündigungsgrund Nichtarier: Die Vertreibung jüdischer Mieter aus den Wiener Gemeindebauten in den Jahren 1938 – 1939. Wien, Picus Verlag, 1996.
- FGDC (Federal Geographic Data Committee): Content Standard for Digital Spatial Metadata. Im WWW unter: <http://www.fgdc.gov/metadata/constan.html>, 2001.
- FITZKE, J., RINNER, C., SCHMIDT, D.: GIS-Anwendungen im Internet. In: GIS-Zeitschrift für raumbezogene Information und Entscheidungen, 10.Jg., 1997.
- FRANK, A.U.: The use of Geographical Information Systems: The user interface is the system. London, Belhaven Press, 1993.
- FRANK, A.U., EGENHOFER, M.J., JACKSON, J.P.: A topological data model for spatial databases. In: Symposium on the Design and Implementation of Large Spatial Databases. New York, Springer-Verlag, 1990.
- FRANK, A.U., EGENHOFER, M.J.: Object-Oriented Modeling in GIS: Inheritance and Propagation. In: AUTO-CARTO 9, Ninth International Symposium on Computer-Assisted Cartography. Baltimore, American Society for Photogrammetry and Remote Sensing and American Congress on Surveying and Mapping, 1989.
- FREITAG, U.: Die Entwicklung der Theorie der Kartographie. Symposium Theorie 2000. Dresden, 2000.
- FRITZKE, J., GREVE, K.: GIS und WWW-Vom Prototyp zur Anwendung. In: STROBL, J., DOLLINGER, F.: Angewandte Geographische Informationsverarbeitung X. Beiträge zum AGIT-Symposium 1998. Heidelberg, Wichmann Verlag, 1998.

- FRITZSCHE, A.: Effiziente Datenrecherche und Pflege durch eine Datenbank-Inter-  
netkopplung am Beispiel des Projekts OWINET in Oberfranken (= Beitrag zum  
Symposium: Computerunterstützte Raumplanung CORP '98). Im WWW  
unter: <http://www.corp.at/html/fritzsche.html>, 1998.
- FSF (Free Software Foundation): GNU-GPL-General Public License. IM WWW  
unter.: <http://www.gnu.org/copyleft/gpl.html>, 2001.
- FUTTER, N.: Flexible Web-Technologien als Basis für breite Anwendbarkeit von  
GIS. In: STROBL, J., BLASCHKE, T.: Angewandte Geographische Infor-  
mationsverarbeitung XI. Beiträge zum AGIT-Symposium 1999. Heidel-  
berg, Wichmann Verlag, 1999.
- GARDELS, K.: A Web Mapping Scenario (=OpenGIS®Project Document 98-068).  
OpenGIS®Consortium, Inc., 1998.
- GORKI, H.F., PAPE, H.: Stadtkartographie (= Die Kartographie und ihre Rand-  
gebiete Band III/1). Wien, Franz Deuticke, 1987.
- HAACK, S.: Erstellung einer Web-Applikation mit Open Source Mitteln. Tech-  
nische Fachhochschule Berlin, Dipl.-Arb., 2000.
- HAKE, G., GRÜNREICH, D.: Kartographie. Berlin - New York, Walter de Gruyter,  
1994.
- HEUER, A., SAAKE, G.: Datenbanken - Implentierungstechniken. Landsberg,  
mitp-Verlag, 1999.
- HEUER, A., SAAKE, G.: Datenbanken - Konzepte und Sprachen. Landsberg, mitp-  
Verlag, 2000.
- ICAVIS (Commission on Visualization and Virtual Environments): Im WWW unter:  
<http://www.geovista.psu.edu/icavis>, 2001.
- JÖRG, W.: GIS-Applikationen der Stadt Wien im Intranet und Internet. In: STROBL,  
J., BLASCHKE, T.: Angewandte Geographische Informationsverarbeitung.  
Beiträge zum AGIT-Symposium 1999. Heidelberg, Wichmann Verlag, 1999.
- KAINZ, W.: Datenmodelle und Datenbanken für raumbezogene Informationssysteme. In:  
MAYER, F. (Hrsg.): Digitale Technologien in der Kartographie, Wiener  
Symposium 1988, Tagungsband. Wien, Institut für Geographie der Universität  
Wien, 1989 (=Wiener Schriften zur Geographie und Kartographie, Band 2).
- KAINZ, W.: Skriptum der Vorlesung "Mathematik für Kartographen". Wien, Institut für  
eographie der Universität Wien, 1995.
- KAINZ, W.: Qualitätsaspekte bei der Bearbeitung und Verwendung von Geodaten. In:  
RETSCHMER, I., KRIZ, K. (Hrsg.): 25 Jahre Studienweig Kartographie.

- Wien, Institut für Geographie der Universität Wien, 1999 (=Wiener Schriften zur Geographie und Kartographie, Band 12).
- KOLB, W., STURM, V.: Die digitale Katastralmappe in einem Geographischen Informationssystem. In: STROBL, J., DOLLINGER, F.: Angewandte Geographische Informationsverarbeitung X. Beiträge zum AGIT-Symposium 1998. Heidelberg, Wichmann Verlag, 1998.
- KRAAK, M. J.: Web Cartography - Developments and Prospects. London, Taylor & Francis, 2000.
- KRIZ, K.: Perspektiven in der Kartographie. In: KRETSCHMER, I., KRIZ, K. (Hrsg.): 25 Jahre Studienzweig Kartographie. Wien, Institut für Geographie der Universität Wien, 1999 (=Wiener Schriften zur Geographie und Kartographie, Band 12).
- MA-41 (Magistratsabteilung 41): Stadtvermessung Wien, Aufgaben und Produkte. Im WWW unter: <http://www.wien.gv.at/ma41/index.htm>, 2001.
- MEISSNER, A.: Datenvisualisierung in offenen GIS Umgebungen. Geographisches Institut Freie Universität Berlin, Dipl.-Arb., 1997.
- MONMONIER, M.: How to lie with maps. Chicago, The University of Chicago Press, 1991.
- MOSSGRABER, J.: Konzeption, Entwurf und Umsetzung eines Metadatenmodells zur Interpretation und Verwaltung von Informationen mit geographischem Bezug. Fakultät für Informatik Universität Karlsruhe, Dipl.-Arb., 1997.
- MÜLLER, J.C., LAGRANGE, J.P., WEIBEL, R.: GIS and Generalization, Methodology and Practice. London, Taylor & Francis, 1995.
- MySQL Technical Information: Im WWW unter: <http://www.mysql.com/information>, 2001.
- OGC (OpenGIS®Consortium, Inc.): OpenGIS Simple Feature Specification For SQL (=Open GIS®Project Document 99-049). OpenGIS®Consortium Inc., 1999.
- OGC (OpenGIS®Consortium, Inc.): Request for Quotation and Call for Participation in the OGC Web Mapping Szenario. OpenGIS®Consortium Inc., 2000.
- OGC (OpenGIS®Consortium, Inc.): Im WWW unter: <http://www.opengis.org>, 2001.
- ORACLE: Spatial. Im WWW unter: <http://otn.oracle.com/products/spatial>, 2001.
- OSWALDER, G.: Datenqualität. Institut für Geographie der Universität Wien, Dipl.-Arb., 1996.

- ÖSTERREICHISCHES VOLKSHOCHSCHULARCHIV (Hrsg.): Kündigungsgrund: Nichtarier (=Spurensuche - Zeitschrift für Geschichte der Erwachsenenbildung und Wissenschaftspopularisierung. 11. Jg., 2000, Heft 1-2). Wien, Österreichischer Volkshochschulverlag, 2000.
- PAUL, G.: Aufbau eines Digitalen Landschaftsmodells von Österreich. In: AKTIS – Stand und Fortführung (= Schriftenreihe des DVW, Band 39). Stuttgart, Wittwer Verlag, 2000.
- PETCHENIK, B., ROBINSON, R.: The Map as a Communications System. In: The Cartographic Journal Vol. 12, 1975.
- PETERSON, M. P.: Cartography and the Internet: Introduction and Research Agenda. In: Cartographic Perspectives, 26, 1997. Im WWW unter: <http://maps.unomaha.edu/NACIS/CP26/article1.html>, 1997.
- RIEDL, A.: Virtuelle Globen in der Geovisualisierung. Untersuchungen zum Einsatz von Multimediatechniken in der Geokommunikation. (= KRETSCHMER, I., KRIZ, K. (Hrsg.): Wiener Schriften zur Geographie und Kartographie, Band 13). Wien, Institut für Geographie der Universität Wien, 2000.
- SAMLAND, W., HINTERMEIER, A., DÜRR, J.: Extensible Markup Language (XML) und Scalable Vector Graphics (SVG). Karlsruhe, infowerk Nürnberg, 2000.
- SCHENK, H.J.: Relational Database Concept and Research. Aspects to Cover Spatial Data Needs. In: Photogrammetrie et Systemes d'Information du Territoire. Lausanne, Presses Polytechniques Romandes, 1990.
- SCHICK, B.: Geoinformationssysteme im Internet: Die SICAD Internet Suite. In: STROBL, J., DOLLINGER, F.: Angewandte Geographische Informationsverarbeitung X. Beiträge zum AGIT-Symposium 1998. Heidelberg, Wichmann Verlag, 1998.
- SCHILCHER, M., ROSCHLAUB, R., KALTENBACH, H.: Geoinformationssysteme - Zwischenbilanz einer stürmischen Entwicklung. Im WWW unter: <http://www.gis.bv.tum.de/public/schilcher/zfv-artikel/>, 1996.
- SCHUBERT, H.: Datengrundlagen und Datenverfügbarkeit für Raumplanung in Österreich. (= Beitrag zum Symposium: Computerunterstützte Raumplanung CORP '97). Im WWW unter: <http://www.corp.at/corp97/html/schubert.htm>, 1997.
- SCHWARTZ, S., TOCHTERMANN, K., RIEKERT, W.F., HERMSDÖRFER, D., HANSEN, J.: Open Geospatial Warehouse - Raumbezogene Dienstleistungen im

- Intra-/Internet der Stadt Köln. In: STROBL, J., BLASCHKE, T.: Angewandte Geographische Informationsverarbeitung XI. Beiträge zum AGIT-Symposium 1999. Heidelberg, Wichmann Verlag, 1999.
- STORCH, H.: WebGIS oder WWW-Mapping? Die Grenzen von WWW-Strategien für den öffentlichen Zugang zu raumbezogenen Umweltinformationen. In: STROBL, J., BLASCHKE, T.: Angewandte Geographische Informationsverarbeitung XI. Beiträge zum AGIT-Symposium 1999. Heidelberg, Wichmann Verlag, 1999.
- SYKORA, P.: Orientierung im Hochgebirge mit Karte und GPS. Institut für Geographie und Regionalforschung Universität Wien, Dipl.-Arb., 2000.
- TAYLOR, F.: Challenge and Response in Cartographic Design. In: KELLER, P., WOOD, C. (Hrsg.): Cartographic Design, Theoretical and Practical Perspectives. New York, John Wiley & Sons, 1996.
- UMN MapServer. University of Minnesota Free Mapserver . Im WWW unter: <http://mapserver.gis.umn.edu>, 2001.
- VOSS, A.: Das große PC & Internet Lexikon 2000. Düsseldorf, Data Becker GmbH & Co. KG, 2000.
- VOSSEN, G., WITT, K.U.: Entwicklungstendenzen bei Datenbank-Systemen. München, R. Oldenbourg Verlag, 1991.
- WEBSTERS's New World College Dictionary. London, MacMillan Reference Books, 1999.
- WEIDENMANN, B.: Psychische Prozesse beim Verstehen von Bildern. Bern, Hans Huber, 1988.
- WIEN GRAFIK - Das Grafische Informationssystem (GIS) der Stadt Wien. Im WWW unter: <http://www.wien.gv.at/gdvmo2/wiengrafik/>, 2001.
- WIESER, T., THÜRKOW, D.: Projektorientiertes Datenmanagement unter Verwendung einer Relationalen Datenbank. In: STROBL, J., BLASCHKE, T.: Angewandte Geographische Informationsverarbeitung XI. Beiträge zum AGIT-Symposium 1999. Heidelberg, Wichmann Verlag, 1999.
- WILMERSDORF, E.: Graphische Datenverarbeitung in der großmaßstäbigen Stadtkartographie. In: MAYER, F. (Hrsg.): Kartographiekongreß Wien 1989, zugleich 38. Deutscher Kartographentag, Tagungsband. Wien, Institut für Geographie der Universität Wien, 1990 (=Wiener Schriften zur Geographie und Kartographie, Band 4).

- WILMERSDORF, E.: Jüngste Entwicklungen der computerunterstützten Stadtkartographie in der Stadt Wien. In: KRETSCHMER, I., KRIZ, K. (Hrsg.): Kartographie in Österreich '96. Wien, Institut für Geographie der Universität Wien, 1996 (=Wiener Schriften zur Geographie und Kartographie, Band 9).
- WINTER, A. M.: Internetkartographie mit SVG - Prototyp für einen thematischen Atlas. Institut für Geographie und Regionalforschung Universität Wien, Dipl.-Arb., 2000.
- ZAKON, R.: Hobbes' Internet Timeline Version 5.2. Im WWW unter: <http://www.zakon.org/robert/internet/timeline>, 2001.

## Verzeichnis der Abkürzungen

<b>ANSI</b>	American National Standard Institute
<b>ARPA</b>	Advanced Research Project Agency
<b>ASF</b>	Apache Software Foundation
<b>BEV</b>	Bundesamt für Eich- und Vermessungswesen
<b>CERN</b>	Europäisches Forschungszentrum für Nuklearphysik
<b>CGI</b>	Common Gateway Interface
<b>DB</b>	Datenbank
<b>DBMS</b>	Datenbankmanagementsystem
<b>DBS</b>	Datenbanksystem
<b>DDL</b>	Data Definition Language
<b>DKM</b>	Digitale Katastralmappe
<b>DLM</b>	Digitales Landschaftsmodell
<b>DML</b>	Data Manipulation Language
<b>FGDC</b>	Federal Geographic Data Committee
<b>FTP</b>	File Transfer Protocol
<b>GIF</b>	Graphics Interchange Format
<b>GNU</b>	GNU is not Unix
<b>GPL</b>	General Public License
<b>GUI</b>	Graphical User Interface
<b>HTML</b>	Hypertext Markup Language
<b>ICAIVS</b>	International Cartographic Association - Commission on Visualization & Virtual Environments
<b>IQL</b>	Interactive Query Language
<b>ISO</b>	International Organization for Standardization
<b>JDBC</b>	Java Database Connectivity
<b>JPEG</b>	Joint Photographic Interchange Group
<b>MNDNR</b>	Minnesota Department of Natural Resources
<b>MZK</b>	Mehrzweckstadtkarte
<b>ODBC</b>	Open Database Connectivity
<b>OGC</b>	OpenGIS® Consortium
<b>PARC</b>	Palo Alto Research Center
<b>PHP</b>	PHP Hypertext Preprocessor
<b>QBE</b>	Query by Example
<b>QUEL</b>	Query Language
<b>RBW</b>	Räumliches Bezugssystem Wien
<b>SEQUEL</b>	Structured English Query Language
<b>SQL</b>	Standard Query Language
<b>SSL</b>	Storage Structure Language
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol
<b>UMN</b>	University of Minnesota
<b>WIMP</b>	Windows, Icons, Menues, Pointer
<b>WWW</b>	World Wide Web

## Abbildungsverzeichnis

Abbildung 1: Struktur eines Datei-Systems.....	13
Abbildung 2: Aufbau eines Daten-Verwaltungs-Systems.....	14
Abbildung 3: Struktur moderner Datenbankmanagementsysteme.....	16
Abbildung 4: Der objektorientierte Ansatz.....	20
Abbildung 5: Vergleich Rasterdaten - Vektordaten.....	21
Abbildung 6: Objektdarstellung mittels Vektordaten.....	22
Abbildung 7: Position der Datenqualität im kartographischen Qualitätsmodell.....	26
Abbildung 8: Darstellung der Tabelle 'Polygone'.....	35
Abbildung 9: Rechteckselektion (1).....	36
Abbildung 10: Rechteckselektion (2).....	37
Abbildung 11: Modell der linearen kartographischen Kommunikation.....	43
Abbildung 12: Modell der dialogischen kartographischen Kommunikation.....	45
Abbildung 13: Modell der monologischen kartographischen Kommunikation.....	46
Abbildung 14: Ausschnitte aus der Digitalen Katastralmappe.....	54
Abbildung 15: Ausschnitte aus dem Räumlichen Bezugssystem Wien.....	55
Abbildung 16: Ausschnitte aus der Mehrzweckstadtkarte Wien.....	57
Abbildung 17: Entwicklung der Anzahl der Internet-Hosts.....	63
Abbildung 18: Aufbau des 3-Schicht-Client/server-Modells.....	75
Abbildung 19: Allgemeine Funktionsweise eines Internet-basierten datenbankgestützten Mapserver-Systems.....	86
Abbildung 20: Darstellung der Realnutzung nach definierten Nutzungsklassen.....	94
Abbildung 21: Platzierung der Straßennamen mittels Liniengeometrie des Straßennetzes.....	94
Abbildung 22: Visualisierung von Räumungsadressen im Datenbestand 'Adressen'.....	95
Abbildung 23: Schematisches Ablaufdiagramm des Prototyps.....	104
Abbildung 24: Anzeige der vollständigen Information aus Datenbanksystem.....	108
Abbildung 25: Hauptbildschirm des Mapserver-Prototyps.....	111
Abbildung 26: Abfrage eines Geoobjekts.....	114

## **Tabellenverzeichnis**

Tabelle 1: Beziehungstypen in einem relationalen Datenmodell.....	18
Tabelle 2: Gegenüberstellung Vektordaten-Rasterdaten.....	23
Tabelle 3: Inhalt der Tabelle 'Capital' .....	34
Tabelle 4: Inhalt der Tabelle 'Polygone'.....	35
Tabelle 5: Methoden zum Testen räumlicher Beziehungen gemäß OpenGIS®Consortium-SQL92Erweiterung.....	39
Tabelle 6: Methoden zur räumlichen Analyse gemäß OpenGIS®Consortium-SQL92Erweiterung.....	39

# Programmcode

## UMN MapServer Konfigurationsdatei ("map file")

```
#
# Start of map file
#
# $Id: proto.map,v 1.11 2001/04/28 19:21:16 Alexander Pucher $
#
NAME PROTO
STATUS ON
SIZE 600 450
SHADESET symbols/shade.sym
MARKERSET symbols/marker.sym
LINESET symbols/line.sym
EXTENT 1567.65 341416.206 2205.776 341919.681
UNITS METERS
SHAPEPATH "../data"
IMAGECOLOR 255 255 255
FONTSET ../data/fonts.txt

#LABELOVERLAP FALSE

#
# Start of web interface definition
#
WEB

    MINSCALE 1000
    MAXSCALE 1550000
#
# On Windows systems, /tmp and /tmp/ms_tmp/ should be created at the root
# of the drive where the .MAP file resides.
#
    IMAGEPATH "/tmp/vienna_tmp/"
    IMAGEURL "/vienna_tmp/"
    LOG "/tmp/gmap.log"
END

#
# Start of reference map
#
REFERENCE
    IMAGE images/adv.gif
    EXTENT 1567.65 341416.206 2205.776 341919.681
    SIZE 100 89
    STATUS ON
    COLOR -1 -1 -1
    OUTLINECOLOR 255 0 0
END
```

```
QUERYMAP
  STYLE HILITE
  COLOR 255 0 0
END
```

```
#
# Start of legend
#
LEGEND
  KEYSIZE 18 12
  LABEL
    TYPE BITMAP
    SIZE MEDIUM
    COLOR 0 0 89
  END
  STATUS ON
END
```

```
#
# Start of scalebar
#
SCALEBAR
  IMAGECOLOR 255 255 255
  LABEL
    COLOR 0 0 0
    SIZE SMALL
  END
  SIZE 150 5
  COLOR 255 255 255
  BACKGROUNDCOLOR 0 0 0
  OUTLINECOLOR 0 0 0
  UNITS meters
  INTERVALS 5
  STATUS ON
END
```

```
#
# Start of layer definitions
#
LAYER
  NAME park
  DESCRIPTION "Grünflächen"
  TYPE POLYGON
  STATUS ON
  DATA park
# LABELITEM "NUTCD"
  CLASS
    NAME "Grünflächen"
    COLOR 83 193 80
    SYMBOL 3
    LABEL
      POSITION AUTO
    SIZE MEDIUM
    COLOR 0 0 0
  END
```

```
    END
END

LAYER
  NAME wohn
  DESCRIPTION "Wohngebiete"
  TYPE POLYGON
  STATUS ON
  DATA wohn
# LABELITEM "NUTCD"
  CLASS
    NAME "Wohngebiete"
    COLOR 255 195 195
    SYMBOL 3
    LABEL
      POSITION AUTO
    SIZE MEDIUM
    COLOR 0 0 0
  END
END
END
```

```
LAYER
  NAME verwal
  DESCRIPTION "Öffentliche Gebäude"
  TYPE POLYGON
  STATUS ON
  DATA verwal
# LABELITEM "NUTCD"
  CLASS
    NAME "Öffentliche Gebäude"
    COLOR 202 51 51
    SYMBOL 3
    LABEL
      POSITION AUTO
    SIZE MEDIUM
    COLOR 0 0 0
  END
END
END
```

```
LAYER
  NAME bahn
  DESCRIPTION "Strassenbahn"
  TYPE POLYGON
  STATUS ON
  DATA bahn
# LABELITEM "NUTCD"
  CLASS
    NAME "Strassenbahn"
    COLOR 202 51 0
    SYMBOL 3
    LABEL
      POSITION AUTO
```

```
        SIZE MEDIUM
        COLOR 0 0 0
    END
END
END
```

```
LAYER
NAME schule
DESCRIPTION "Schulen"
TYPE POLYGON
STATUS ON
DATA schule
# LABELITEM "NUTCD"
    CLASS
        NAME "Schulen"
        COLOR 210 210 210
        SYMBOL 3
        LABEL
            POSITION AUTO
            SIZE MEDIUM
            COLOR 0 0 0
    END
END
END
```

```
LAYER
NAME kirche
DESCRIPTION "Kirchen"
TYPE POLYGON
STATUS ON
DATA kirche
# LABELITEM "NUTCD"
    CLASS
        NAME "kirche"
        COLOR 120 120 120
        SYMBOL 3
        LABEL
            POSITION AUTO
            SIZE MEDIUM
            COLOR 0 0 0
    END
END
END
```

```
LAYER
NAME strasse
DESCRIPTION "Strassen"
TYPE LINE
STATUS ON
DATA strasse
# LABELITEM "STRABS"
    CLASS
        NAME "Strassen"
        COLOR 0 0 0
        SYMBOL 1
```

```

    LABEL
#   TYPE TRUETYPE
#   ANTIALIAS
#   FONT arial
#   SIZE 11
#   MINFEATURESIZE 100
    ANGLE AUTO
    POSITION UC
#   OFFSET 0 2
#   buffer 2
#   SIZE MEDIUM
    COLOR 0 0 0
    END
  END
END

```

```

LAYER
  NAME adr
  DESCRIPTION "Adressen"
  TYPE POINT
  STATUS ON
  DATA adr
  LABELITEM "VON"
  CLASS
    NAME "Adressen"
    COLOR 0 0 0
    SYMBOL 1
    LABEL
      POSITION AUTO
      SIZE MEDIUM
      COLOR 0 0 0
    END
  END
END

```

```

HEADER "ADR_ID X Y STRASSENNA VON"
TOLERANCE 10
QUERYITEM "ADR_ID"
FOOTER "gew_footer.html"
QUERY
  EXPRESSION ./
  TEMPLATE "tnt"
END
END

```

```

LAYER
  NAME proto
  DESCRIPTION "Juden"
  TYPE POINT
  STATUS ON
  DATA proto
# LABELITEM "ZUNAME"
  CLASS
    NAME "Juden"
    COLOR 255 0 0
    SYMBOL 3

```

```

        LABEL
        POSITION AUTO
        SIZE MEDIUM
        COLOR 0 0 0
    END
END

HEADER "VORNAME ZUNAME KUENDBEZ KUENDADR KUENDNR INFOS"
TOLERANCE 10
QUERYITEM "ADR_ID"
# FOOTER "gew_footer.html"
QUERY
    EXPRESSION ./
    TEMPLATE "tft"
END
END

END # Map File

```

## PHP - Template Datei

```

<?php_track_vars?>
<HTML>
<?php
/*****
/* $Id: proto.phtml,v 1.24 2001/04/28 15:51:50 Alexander Pucher $ */
/*****
/*      GMap mapping engine (PHP/MapScript version)      */
/*                                          */
/* This application uses the PHP/MapScript MapServer extension */
/* to navigate through a dataset.                      */
/*****

/* ----- */
/* Load required modules                               */
/* ----- */
if (PHP_OS == "WINNT" || PHP_OS == "WIN32")
{
    dl("php3_mapscript.dll");
    // dl("php3_dbase.dll");
}
else
{
    dl("php_mapscript.so");
}

include("vienna.php3");

if ($HTTP_FORM_VARS["MAP_NAME"])
    $gpoMap = ms_newMapObj(strval($HTTP_FORM_VARS["MAP_NAME"]));
else
    $gpoMap = ms_newMapObj("vienna.map");

```

```

//$gpoMap = ms_newMapObj("vienna_key.map");

/* ----- */
/* extents set in the .map file. */
/* */
/* max extents are 10% of the map. */
/* ----- */
$dfTmpDeltaX = ($gpoMap->extent->maxx - $gpoMap->extent->minx)/10;
$dfTmpDeltaY = ($gpoMap->extent->maxy - $gpoMap->extent->miny)/10;
$dfTmpDeltaX = 0;
$dfTmpDeltaY = 0;

$dfMapExtMinX = $gpoMap->extent->minx;
$dfMapExtMinY = $gpoMap->extent->miny;
$dfMapExtMaxX = $gpoMap->extent->maxx;
$dfMapExtMaxY = $gpoMap->extent->maxy;

$dfMaxExtMinX = $gpoMap->extent->minx - $dfTmpDeltaX;
$dfMaxExtMinY = $gpoMap->extent->miny - $dfTmpDeltaY;
$dfMaxExtMaxX = $gpoMap->extent->maxx + $dfTmpDeltaX;
$dfMaxExtMaxY = $gpoMap->extent->maxy + $dfTmpDeltaY;

$gpoQueryResults = "";

GMap75CheckClick();

$gszCommand = $HTTP_FORM_VARS["CMD"];
if ( sizeof($gszCommand) == 0)
{
    $gszCommand = "ZOOM_IN";
}

/* ----- */
/* switch between the rosa java applet and simple html for */
/* zooming and querying. */
/* ----- */

if (strlen($HTTP_FORM_VARS["PREVIOUS_MODE"])==0)
    $gblsHtmlMode = 0; // Start with Java On by default
else
    $gblsHtmlMode = intval($HTTP_FORM_VARS["PREVIOUS_MODE"]);

if ($HTTP_FORM_VARS[JavaOn_x])
    $gblsHtmlMode = 0;

if ($HTTP_FORM_VARS[JavaOff_x])
    $gblsHtmlMode = 1;

function IsHtmlMode()
{
    GLOBAL $gblsHtmlMode;
    return $gblsHtmlMode;
}
?>

```

```

<HEAD>

<SCRIPT LANGUAGE="JavaScript">

function ProvinceSelected()
{
  if (document.forms[0].ViewRegion.selectedIndex != 0 &&
      document.forms[0].ViewRegion.selectedIndex !=1)
    document.forms[0].submit();
}
function MapSizeSelected()
{
  if (document.forms[0].MapSize.selectedIndex != 0 &&
      document.forms[0].MapSize.selectedIndex !=1)
    document.forms[0].submit();
}
</SCRIPT>

<TITLE>MapServer Atlas</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
</HEAD>

<BODY BGCOLOR="#FFFFFF">
<FORM ACTION="vienna.phtml" METHOD=POST NAME="myform">
  <TABLE BORDER="0" CELSPACING="5" CELLPADDING="0">
    <TR>
      <TD VALIGN="TOP" ALIGN="CENTER">

        <TABLE BORDER="0" CELSPACING="0" CELLPADDING="0">
          <TR>
            <TD><IMG SRC="images/corner_TL_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
            <TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
            <TD><IMG SRC="images/corner_TR_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
          </TR>
          <TR ALIGN="CENTER">
            <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
            <TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2"><B>Zeichenerklärung</B></FONT></TD>
            <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
          </TR>
          <TR>
            <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
            <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
            <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
          </TR>
          <TR>
            <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
            <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>

```

```

        <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
    </TR>
    <TR>
        <TD BGCOLOR="#C1D8E3">&nbsp;</TD>
        <TD BGCOLOR="#C1D8E3">
            <TABLE WIDTH="100%" BORDER="0" CELLSPACING="1" CELLPADDING="2">
                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER"><IMG SRC="images/icon_eye.gif" WIDTH="17"
HEIGHT="11"></TD>
                    <TD ALIGN="CENTER">&nbsp;</TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2"><B>Layer</B></FONT></TD>
                </TR>

                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER">
                        <INPUT TYPE="checkbox" NAME="bahn" VALUE="Y"
<?php if (GMapGetStatus("bahn") == 1) echo "CHECKED"; ?>>
                    </TD>
                    <TD ALIGN="CENTER"><IMG SRC="images/str.gif" WIDTH="20"
HEIGHT="15"></TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Öffentlicher Verkehr</FONT></TD>
                </TR>

                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER">
                        <INPUT TYPE="checkbox" NAME="park" VALUE="Y"
<?php if (GMapGetStatus("park") == 1) echo "CHECKED"; ?>>
                    </TD>
                    <TD ALIGN="CENTER"><IMG SRC="images/park.gif" WIDTH="20"
HEIGHT="15"></TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Grünflächen</FONT></TD>
                </TR>

                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER">
                        <INPUT TYPE="checkbox" NAME="wohn" VALUE="Y"
<?php if (GMapGetStatus("wohn") == 1) echo "CHECKED"; ?>>
                    </TD>
                    <TD ALIGN="CENTER"><IMG SRC="images/wohn.gif" WIDTH="20"
HEIGHT="15"></TD>
                    <TD><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2">Wohngebiete</FONT></TD>
                </TR>

                <TR BGCOLOR="#FFFFFF">
                    <TD ALIGN="CENTER">
                        <INPUT TYPE="checkbox" NAME="kirche" VALUE="Y"
<?php if (GMapGetStatus("kirche") == 1) echo "CHECKED"; ?>>
                    </TD>
                    <TD ALIGN="CENTER"><IMG SRC="images/kirche.gif" WIDTH="20"
HEIGHT="15"></TD>

```

```

        <TD><FONT FACE="Arial, Helvetica, sans-serif"
            SIZE="2">Kirche</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
            <INPUT TYPE="checkbox" NAME="schule" VALUE="Y"
                <?php if (GMapGetStatus("schule") == 1) echo "CHECKED"; ?>>
            </TD>
        <TD ALIGN="CENTER"><IMG SRC="images/schule.gif" WIDTH="20"
HEIGHT="15"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif"
            SIZE="2">Schule</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
            <INPUT TYPE="checkbox" NAME="verwal" VALUE="Y"
                <?php if (GMapGetStatus("verwal") == 1) echo "CHECKED"; ?>>
            </TD>
        <TD ALIGN="CENTER"><IMG SRC="images/verwal.gif" WIDTH="20"
HEIGHT="15"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif"
            SIZE="2">Öffentliche Einrichtungen</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
            <INPUT TYPE="checkbox" NAME="strasse" VALUE="Y"
                <?php if (GMapGetStatus("strasse") == 1) echo "CHECKED"; ?>>
            </TD>
        <TD ALIGN="CENTER"><IMG SRC="images/leg_20.gif" WIDTH="20"
HEIGHT="15"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif"
            SIZE="2">Verkehrsflächen</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
            <INPUT TYPE="checkbox" NAME="proto" VALUE="Y"
                <?php if (GMapGetStatus("proto") == 1) echo "CHECKED"; ?>>
            </TD>
        <TD ALIGN="CENTER"><IMG SRC="images/star.gif" WIDTH="17"
HEIGHT="21"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif"
            SIZE="2">Gekündigte Wohnungen</FONT></TD>
    </TR>

    <TR BGCOLOR="#FFFFFF">
        <TD ALIGN="CENTER">
            <INPUT TYPE="checkbox" NAME="adr" VALUE="Y"
                <?php if (GMapGetStatus("adr") == 1) echo "CHECKED"; ?>>
            </TD>
        <TD ALIGN="CENTER"><IMG SRC="images/adr.gif" WIDTH="20"
HEIGHT="15"></TD>

```

```
<TD><FONT FACE="Arial, Helvetica, sans-serif"
      SIZE="2">Adressen</FONT></TD>
</TR>
```

```
<TR BGCOLOR="#FFFFFF">
  <TD ALIGN="CENTER">
    <INPUT TYPE=IMAGE SRC="images/icon_redraw.gif" WIDTH="19"
HEIGHT="19" NAME=redraw BORDER=0>
  </TD>
  <TD COLSPAN="2"><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">Karte
neu zeichnen</FONT></TD>
</TR>
</TABLE>
```

```
</TD>
<TD BGCOLOR="#C1D8E3">&nbsp;</TD>
</TR>
```

```
<TR>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
```

```
</TR>
<TR>
  <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
  <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
  <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
```

```
</TR>
<TR>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
```

```
</TR>
<TR>
  <TD BGCOLOR="#C1D8E3">&nbsp;</TD>
  <TD BGCOLOR="#C1D8E3" ALIGN="CENTER">
    <FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">
    <SELECT NAME="ViewRegion" onChange="ProvinceSelected()>
```

```
      <OPTION>Quick View</OPTION>
      <OPTION>-----</OPTION>
      <OPTION VALUE="1567,341416,2205,341919">GESAMTES GEBIET </OPTION>
      <OPTION VALUE="1789,341558,1989,341758">Liebiggasse 5 </OPTION>
      <OPTION VALUE="1805,341647,2005,341847">Universitätsstrasse 7</OPTION>
```

```

</SELECT >

</FONT>
</TD>
<TD BGCOLOR="#C1D8E3">&nbsp;</TD>
</TR>
<TR>
<TD><IMG SRC="images/corner_BL_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
<TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
<TD><IMG SRC="images/corner_BR_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
</TR>

</TABLE>

</TD>
<TD VALIGN=TOP ALIGN="RIGHT">
<TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
<TR>
<TD><IMG SRC="images/corner_TL_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
<TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
<TD><IMG SRC="images/corner_TR_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
</TR>
<TR ALIGN="CENTER">
<TD BGCOLOR="#ccff">&nbsp;</TD>
<TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica, sans-serif"
SIZE="2"><B>Kündigungsgrund:
Nichtarier - Prototyp</B></FONT></TD>
<TD BGCOLOR="#E2EFF5">&nbsp;</TD>
</TR>
<TR>
<TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
<TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
<TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
</TR>
<TR>
<TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
<TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
<TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
</TR>
<TR>
<TD BGCOLOR="#C1D8E3">&nbsp;</TD>
<TD BGCOLOR="#999999">
<TABLE WIDTH="400" BORDER="0" CELLSPACING="1" CELLPADDING="0"
HEIGHT="300">
<TR BGCOLOR="FFFFFF" ALIGN="CENTER">
<TD><?php GMapDrawMap() ?></TD>
</TR>

```

```

    </TABLE>
  </TD>
  <TD BGCOLOR="#C1D8E3">&nbsp;</TD>
</TR>
<TR>
  <TD BGCOLOR="#C1D8E3">&nbsp;</TD>
  <TD BGCOLOR="#FFFFFF" ALIGN="CENTER"><?php GMapDrawScaleBar()
  ?></TD>
  <TD BGCOLOR="#C1D8E3">&nbsp;</TD>
</TR>
<TR>
  <TD><IMG SRC="images/corner_BL_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
  HEIGHT="4"></TD>
  <TD><IMG SRC="images/corner_BR_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
</TR>
</TABLE>

</TD>
<TD VALIGN="TOP">

<TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
  <TR>
    <TD><IMG SRC="images/corner_TL_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
    <TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif" WIDTH="1"
    HEIGHT="4"></TD>
    <TD><IMG SRC="images/corner_TR_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
  </TR>
  <TR ALIGN="CENTER">
    <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
    <TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica, sans-serif"
    SIZE="2"><B>Übersichtskarte</B></FONT></TD>
    <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
  </TR>
  <TR>
    <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
    HEIGHT="1"></TD>
    <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
    HEIGHT="1"></TD>
    <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
    HEIGHT="1"></TD>
  </TR>
  <TR>
    <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
    HEIGHT="4"></TD>
    <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
    HEIGHT="4"></TD>
    <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
    HEIGHT="4"></TD>
  </TR>
  <TR>
    <TD BGCOLOR="#C1D8E3">&nbsp;</TD>
    <TD><?php GMapDrawKeyMap() ?></TD>
    <TD BGCOLOR="#C1D8E3">&nbsp;</TD>
  </TR>

```

```

<TR>
  <TD><IMG SRC="images/corner_BL_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
  <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
  HEIGHT="4"></TD>
  <TD><IMG SRC="images/corner_BR_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
</TR>
</TABLE>

```

```

<FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">
<BR>
<SELECT NAME="MapSize" onChange="MapSizeSelected()">

```

```

<OPTION>Map Size</OPTION>
<OPTION>-----</OPTION>
<OPTION VALUE="400,300"
  <?php if (($gpoMap->width == 400) && ($gpoMap->height == 300))
    echo "SELECTED";?>> 400 x 300</OPTION>
<OPTION VALUE="600,450"
  <?php if (($gpoMap->width == 600) && ($gpoMap->height == 450))
    echo "SELECTED";?>> 600 x 450</OPTION>
<OPTION VALUE="800,600"
  <?php if (($gpoMap->width == 800) && ($gpoMap->height == 600))
    echo "SELECTED";?>> 800 x 600</OPTION>

```

```

</SELECT>
</FONT><BR>
<BR>

```

```

<?php if (!IsHtmlMode()) echo "<!--" ?>
<TABLE BORDER="0" CELSPACING="0" CELLPADDING="0">
  <TR>
    <TD>
      <INPUT TYPE="radio" NAME="CMD" VALUE="ZOOM_IN"
      <?php if ($gszCommand == "ZOOM_IN") echo "CHECKED"; ?> >
    </TD>
    <TD><IMG SRC="images/icon_zoomin.gif" WIDTH="25" HEIGHT="25"></TD>
  </TR>
  <TR>
    <TD>
      <INPUT TYPE="radio" NAME="CMD" VALUE="ZOOM_OUT"
      <?php if ($gszCommand == "ZOOM_OUT") echo "CHECKED"; ?> >
    </TD>
    <TD><IMG SRC="images/icon_zoomout.gif" WIDTH="25" HEIGHT="25"></TD>
  </TR>
  <TR>
    <TD>
      <INPUT TYPE="radio" NAME="CMD" VALUE="RECENTER"
      <?php if ($gszCommand == "RECENTER") echo "CHECKED"; ?> >
    </TD>
    <TD><IMG SRC="images/icon_recentre.gif" WIDTH="25" HEIGHT="25"></TD>
  </TR>
  <TR>
    <TD>
      <INPUT TYPE="radio" NAME="CMD" VALUE="QUERY_POINT"
      <?php if ($gszCommand == "QUERY_POINT") echo "CHECKED"; ?>>

```

```

</TD>
  <TD><IMG SRC="images/icon_info.gif" WIDTH="25" HEIGHT="25"></TD>
</TR>
</TABLE>
<?php if (!IsHtmlMode()) echo "-->" ?>
<FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">
</FONT></TD>
</TR>
<TR>
  <TD VALIGN="TOP" ALIGN="CENTER">
    <TABLE CELLPADDING="2" CELLSPACING="0">
      <TR>
        <TD><INPUT TYPE="image" BORDER="0" <?php if (IsHtmlMode())
printf("NAME=\"JavaOn\" SRC=\"images/java_on.gif\""); else printf("NAME=\"JavaOff\"
SRC=\"images/java_off.gif\"");?> WIDTH="24" HEIGHT="25"></TD>
        <TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2"><?php if (IsHtmlMode())
printf("Java Mode Disabled<BR>Click to Enable"); else printf("Java Mode Enabled<BR>Click to
Disable");?></FONT>
        <INPUT TYPE="HIDDEN" NAME="PREVIOUS_MODE" VALUE=<?php echo
IsHtmlMode(); ?>></TD>
      </TR>
    </TABLE>
  </TD>
  <TD ALIGN="CENTER">

  <TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
    <TR>
      <TD><IMG SRC="images/corner_TL_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
      <TD BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
      <TD><IMG SRC="images/corner_TR_LB.jpg" WIDTH="4" HEIGHT="4"></TD>
    </TR>
    <TR ALIGN="CENTER">
      <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
      <TD BGCOLOR="#E2EFF5"><FONT FACE="Arial, Helvetica,
sans-serif" SIZE="2"><B>X: <?php echo round
(($gpoMap->extent->minx+$gpoMap->extent->maxx)/2); ?> Y:
      <?php echo round (($gpoMap->extent->miny+$gpoMap->extent->maxy)/2);
?><br>Maßstab: 1 : <?php echo round ($gpoMap->scale); ?></B></FONT></TD>
      <TD BGCOLOR="#E2EFF5">&nbsp;</TD>
    </TR>
    <TR>
      <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
      <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
      <TD BGCOLOR="#999999"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="1"></TD>
    </TR>
    <TR>
      <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
      <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
      <TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"

```

```

        HEIGHT="4"></TD>
</TR>
<TR>
<TD BGCOLOR="#C1D8E3">&nbsp;</TD>
<TD WIDTH=<?php echo $gpoMap->width; if ($gpoQueryResults) echo "
BGCOLOR="#C1D8E3"; ?><?php if ($gpoQueryResults) GMapDumpQueryResults();
else echo "&nbsp;"; ?></TD>
<TD BGCOLOR="#C1D8E3">&nbsp;</TD>
</TR>
<TR>
<TD><IMG SRC="images/corner_BL_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
<TD BGCOLOR="#C1D8E3"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="4"></TD>
<TD><IMG SRC="images/corner_BR_DB.jpg" WIDTH="4" HEIGHT="4"></TD>
</TR>
</TABLE>
</TD>
</TR>
<TR>
<TD COLSPAN="3">&nbsp;</TD>
</TR>
<TR>
<TD COLSPAN="3" BGCOLOR="#E2EFF5"><IMG SRC="images/pixel.gif" WIDTH="1"
HEIGHT="2"></TD>
</TR>
<TR>
<TD COLSPAN="3">&nbsp;</TD>
</TR>
<TR>
<TD COLSPAN="3">
<TABLE WIDTH="100%" BORDER="0" CELLSPACING="0" CELLPADDING="0">
<TR ALIGN="CENTER">
<TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2"><A
HREF="http://www.gis.univie.ac.at/karto"><IMG SRC="images/logo.jpg" WIDTH="45"
HEIGHT="45" BORDER="0"></A></FONT></TD>
<TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2"><A
HREF="http://mapserver.gis.umn.edu/"><IMG
SRC="images/powered_mapserver.jpg" WIDTH="96" HEIGHT="58"
BORDER="0"></A></FONT></TD>
<TD>
<TABLE BORDER="0" CELLSPACING="0" CELLPADDING="0">
<TR>
<TD><A HREF="http://php.net"><IMG SRC="images/logo_php.gif" WIDTH="34"
HEIGHT="19" BORDER="0"></A></TD>
<TD><FONT FACE="Arial, Helvetica, sans-serif" SIZE="2">/
MapScript</FONT></TD>
</TR>
</TABLE>
</TD>
<TD>
<FONT FACE="Arial, Helvetica, sans-serif" SIZE="2"><A
HREF="http://www.dmsolutions.on.ca/rosa/"><IMG SRC="images/powered_rosa.jpg"
WIDTH="100" HEIGHT="36" BORDER="0"></A></FONT></TD>
</TR>

```

```

        </TABLE>
    </TD>
</TR>

</TABLE>
    <?php if ($HTTP_FORM_VARS["MAP_NAME"])
        printf("<INPUT TYPE=HIDDEN NAME=MAP_NAME
VALUE=\\\"%s\\\">",strval($HTTP_FORM_VARS["MAP_NAME"])); ?>
</FORM>
</BODY>
</HTML>

```

## PHP-Skript: Datenbankschnittstelle

```

<?
// Quelle: Alexander Pucher (pucher@atlas.gis.univie.ac.at)
//      www.gis.univie.ac.at/karto
?>
<html>
<head>
<title>Tabellen anzeigen</title>
</head>
<body bgcolor="CCCCFF">
<?

// Verbindungsvariablen definieren

$dbhost="localhost";
$dbuser="root";
$dbpass="mysql";
$dbname="juden";
$table="proto";

// Verbindung herstellen

mysql_connect( "$dbhost", "$dbuser", "$dbpass") or die( "Unable to connect to SQL server");
@mysql_select_db( "$dbname") or die( "Unable to select database");

// Abfrage der Inhalte der Tabelle

if (!$sent)
    $result = mysql_query("select ID, VORNAME, ZUNAME from $table where ID like '%$id%'");
else
    $result = mysql_query("select ID, VORNAME, ZUNAME from $table where ZUNAME
like '%$zuname%'");

// Abfrage und Erstellung der Kopfzeile

if (!$sent)
    $result4 = mysql_query("select VORNAME, ZUNAME, GEBORT, BERUF, RAEUDAT,
KUEBEZ, KUEADR, KUENR, DEPDAT, DEPORT,
ROUND(X)-150 as XMIN, ROUND(X)+150 as XMAX, ROUND(Y)-150 as YMIN,

```

```

ROUND(Y)+150 as YMAX from $table where id like '%$id%');
else
$result4 = mysql_query("select VORNAME, ZUNAME, GEBORT, BERUF, RAEUDAT,
KUEBEZ, KUEADR, KUENR, DEPDAT, DEPORT,
ROUND(X)-150 as XMIN, ROUND(X)+150 as XMAX, ROUND(Y)-150 as YMIN,
ROUND(Y)+150 as YMAX from $table where ZUNAME like '%$zuname%");

while (list ($vorname, $zuname, $gebort, $beruf, $raeutat, $kuebez, $kueadr,
$kuenr,$depdat,$deport,$xmin,$xmax,$ymin,$ymax) = mysql_fetch_row ($result4)) {

print (" <table BORDER CELLSPACING=4 CELLPADDING=4 WIDTH=40%\"
BGCOLOR=#FFFFFF\" bordercolor=#666666\">\n".
" <tr>\n".
" <td><b>Vorname</b></td>\n".
" <td>$vorname</td>\n".
" </tr>\n".
" <tr>\n".
" <td><b>Zuname</b></td>\n".
" <td>$zuname</td>\n".
" </tr>\n".
" <BR>\n".
" <tr>\n".
" <td><b>Geburtsort</b></td>\n".
" <td>$gebort</td>\n".
" </tr>\n".
" <BR>\n".
" <tr>\n".
" <td><b>Beruf</b></td>\n".
" <td>$beruf</td>\n".
" </tr>\n".
" <BR>\n".
" <tr>\n".
" <td><b>Kündigungsadresse</b></td>\n".
" <td>$kuebez, $kueadr $kuenr</td>\n".
" </tr>\n".
" <BR>\n".
" <tr>\n".
" <td><b>Räumungsdatum</b></td>\n".
" <td>$raeutat</td>\n".
" </tr>\n".
" <BR>\n".
" <tr>\n".
" <td><b>Deportationsdatum</b></td>\n".
" <td>$depdat</td>\n".
" </tr>\n".
" <BR>\n".
" <tr>\n".
" <td><b>Deportationsort</b></td>\n".
" <td>$deport</td>\n".
" </tr>\n".
" <BR>\n".
" <tr>\n".
" <td><b>In Karte anzeigen</b></td>\n".
" <td><a href = \"vienna.phtml?ViewRegion=$xmin,$ymin,$xmax,$ymax\">In Karte
anzeigen</a></td>\n".

```

```
        " </tr>\n".
        "</table>\n" );
    }

?>
<P>
Nach Name suchen:
<FORM method=post action=proto_query.php3>
<INPUT type=text name=zuname size=25>
<INPUT type=submit value=Suchen>
<INPUT type=hidden name=sent value=1>
<br><br><br>
<HR>
<PRE>
<?
echo "Last modified: ".date("F d Y H:i:s", getlastmod());
?>
</PRE>
</body>
</html>
```